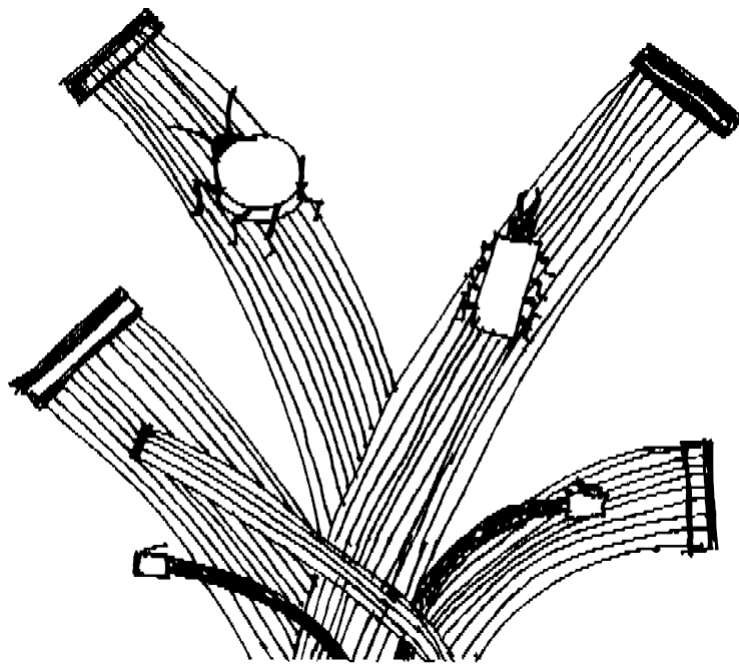


digital self- defense guide

collective work



sixth edition

winter 2023

digital self-defense guide

Collective work
guide@boum.org

OpenPGP fingerprint:
D487 4FA4 F6B6 88DC 0913
C9FD 326F 9F67 250B 0939

winter 2023

Made with free software, in particular *bookdown*, *Pandoc* and *LaTeX* for page layout, *GIMP* and *Inkscape* for images, *Scribus* for covers, *Git* and numerous discussions to work together, all under *Debian GNU/Linux* and *Tails*.



Copyleft: this work is free, you can co- pier it, distribute it and modify it under the terms of the *Free Art License* - <http://www.artlibre.org/>

ISBN : 978-2-912631-05-3

Contents

Contents	iii
Preface to this edition	xi
Why this guide?	1
The downside of digital memory	1
Nothing to hide?	1
Understanding so as to be able to choose	2
Take the time to understand	3
How to read this guide	5
An "offline" tome	5
An "online" tome	5
Casting off	5
Volume 1 - Off-line	9
<hr/>	
I Understanding	11
<hr/>	
Introduction	13
1 Computer basics	15
1.1 Data processing machines	15
1.2 Hardware	15
1.3 Electricity, magnetic fields, noise and radio waves	21
1.4 Software	22
1.5 Data storage	23
2 Traces on every floor	27
2.1 In RAM	27
2.2 In virtual memory	28
2.3 Sleep and hibernation	28
2.4 Newspapers	29
2.5 Automatic backups and other lists	29
2.6 Metadata	30

3	Malware, bugs and spyware	31
3.1	Legal background	31
3.2	Malware	32
3.3	Spy equipment	34
3.4	Keyloggers, or keystroke recorders	35
3.5	Digital investigation platforms	36
3.6	Printing problems?	36
4	A few safety illusions	39
4.1	Proprietary, open source, free software.....	39
4.2	An account's password does not protect its data	41
4.3	About "deleting" files.....	42
4.4	Portable software: a false solution.....	44
5	One way to protect data: cryptography	47
5.1	Protecting data from prying eyes.....	47
5.2	Ensuring data integrity	53
5.3	Symmetrical, asymmetrical?.....	55
II	Choosing appropriate responses	57
<hr/>		
	Introduction	59
6	Trust and risk reduction	61
6.1	Risk reduction.....	61
6.2	A story of trust.....	62
7	Risk assessment	63
7.1	What do we want to protect?	63
7.2	Who are we protecting ourselves against?	63
8	Defining a security policy	65
8.1	A matter of compromise.....	65
8.2	What to do?	65
8.3	A few rules	66
	Use cases	69
9	Use case: a fresh start, to stop paying the piper	71
9.1	Background.....	71
9.2	Assessing risks.....	72
9.3	Defining a security policy.....	72
10	Use case: working on a sensitive document	79
10.1	Context.....	79
10.2	Assessing risks.....	79
10.3	Which operating system is best for working on the document?	80
10.4	Working on a sensitive document... on a <i>live</i> system	82
10.5	Working on a sensitive document... in Windows	82
10.6	Clean up the metadata of the finished document	88
10.7	Limits common to these safety policies	88

11 Use case: archiving a completed project	89
11.1 Context.....	89
11.2 Is this really necessary?	89
11.3 Assessing risks	89
11.4 Method.....	90
11.5 What passphrase?.....	90
11.6 A hard drive? One key? Several keys?	91
III Tools	93
<hr/>	
Introduction	95
12 Using a terminal	97
12.1 What is a terminal?.....	97
12.2 About controls.....	98
12.3 Administrative privileges	99
12.4 Warning.....	100
12.5 One exercise	100
12.6 Watch out for tracks!	101
12.7 Further information	101
13 Choose a passphrase	103
14 Booting from CD, DVD or USB stick	107
14.1 Try naively.....	107
14.2 Attempting a one-time boot device selection	107
14.3 Modifying firmware parameters	108
15 Using a <i>live</i> system	113
15.1 Discrete <i>live</i> systems.....	113
15.2 Download, check and install Tails	114
15.3 Cloning or updating a Tails key.....	115
15.4 Booting on a <i>live</i> system	115
15.5 Using Tails persistence	116
16 Installing an encrypted system	119
16.1 Limits.....	119
16.2 Download installation support.....	120
16.3 Check the footprint of the installation image.....	121
16.4 Preparing the installation support	122
16.5 The installation itself.....	123
16.6 Setting up the Debian main package repository.....	128
16.7 A few tips for continuing	129
16.8 Documentation on Debian and GNU/Linux.....	129
17 Choosing, checking and installing software	131
17.1 Selection criteria.....	131
17.2 Find and install software.....	134
17.3 Finding and installing a Debian package	135
17.4 Adding deposits.....	136

18 Deleting data "for real	139
18.1 A little theory	139
18.2 On other systems.....	140
18.3 Let's go.....	140
18.4 Deleting files... and their contents	141
18.5 Deleting an entire disk "for real	141
18.6 Deleting the entire contents of a disc	142
18.7 Making previously deleted data irrecoverable	143
19 Partitioning and encrypting a hard disk	145
19.1 Overview.....	145
19.2 Preparing a disk for encryption	146
19.3 Creating an unencrypted partition	147
19.4 Creating an encrypted partition.....	148
19.5 Using an encrypted hard disk.....	148
20 Saving data	151
20.1 Special case of persistent Tails storage	151
20.2 With file manager and encrypted storage	151
20.3 Using Déjà Dup.....	153
21 Sharing a secret	157
21.1 Share a passphrase.....	157
21.2 Reconstruct passphrase	158
22 Using checksums	161
22.1 Get the checksum of a file	161
22.2 Check file integrity.....	162
23 Installing and using a virtualized system	163
23.1 Installing the Virtual Machine Manager.....	163
23.2 Enabling hardware virtualization.....	164
23.3 Installing a virtualized Windows.....	165
23.4 Taking a snapshot of a virtual machine	168
23.5 Restore the state of a virtual machine from a snapshot	168
23.6 Installing new software on a virtualized system	169
23.7 Sharing a USB stick with a virtualized system.....	170
23.8 Sharing a CD or DVD with a virtualized system.....	171
23.9 Sharing a folder with a virtualized system	171
24 Keeping your system up to date	175
24.1 Keeping Tails up to date	175
24.2 Keeping an encrypted system up to date	176
24.3 Daily updates for an encrypted system	176
24.4 Upgrading to a new stable version.....	177
25 Clean up document metadata	185
25.1 Installing the necessary software.....	185
25.2 Cleaning one or more files	185

Volume 2 - Online **189**

IV Understanding **191**

Introduction	193
26 Network basics	197
26.1 Interconnected computers	197
26.2 Communication protocols	199
26.3 Local networks	203
26.4 Internet: interconnected networks	205
26.5 Customers and servers	209
27 Tracks all along the line	213
27.1 On client computer	213
27.2 On the box: network card hardware address	215
27.3 On routers: packet headers	217
27.4 On the server	217
27.5 The traces we leave behind	219
28 Communications monitoring and control	221
28.1 Who wants the data back?	221
28.2 Logs and data retention	224
28.3 Mass listening	229
28.4 Targeted attacks	231
28.5 In conclusion	238
29 Web 2.0	239
29.1 Rich Internet applications"	239
29.2 ... and volunteer web surfers	240
29.3 Data centralization	240
29.4 Program control	241
29.5 From centralization to decentralized self-hosting	242
30 Contextual identities	243
30.1 Definitions	243
30.2 From contextual identity to civil identity	244
30.3 Compartmentalization	246
30.4 Social media: centralized functions and a unique identity	247
31 Hiding the content of communications: asymmetric cryptography	249
31.1 Limits of symmetric encryption	249
31.2 A solution: asymmetric cryptography	249
31.3 End-to-end encryption	251
31.4 Digital Signature	252
31.5 Verify public key authenticity	253
31.6 Persistent confidentiality	258
31.7 Summary and limitations	258
32 Tor or onion routing	261
32.1 The problem: hiding origin and destination	261
32.2 One solution: Tor	261
32.3 onion services	266
32.4 Join the Tor network	266
32.5 Some limitations of Tor	267

V	Choosing the right answers	273
<hr/>		
	Introduction	275
	33 Use case: consulting websites	277
	33.1 Background	277
	33.2 Assessing risks	277
	33.3 Defining a security policy.....	278
	33.4 Choose from available tools	279
	33.5 Browsing websites with Tor Browser	281
	33.6 Browsing websites with Tails	282
	34 Use case: publishing a document	285
	34.1 Background.....	285
	34.2 Assessing risks	285
	34.3 Defining a security policy	285
	35 Use case: exchanging messages	289
	35.1 Background.....	289
	35.2 Assessing risks	289
	35.3 Two issues	290
	35.4 Webmail or mail client?.....	290
	35.5 Webmail	291
	35.6 Customer mail	292
	35.7 Exchange emails while hiding your identity.....	293
	35.8 Exchange confidential emails	295
	36 Use case: dialog	299
	36.1 Background.....	299
	36.2 Assessing risks	299
	36.3 Defining a security policy	299
	36.4 The limits.....	301
	37 Use case: sharing sensitive documents	303
	37.1 Background.....	303
	37.2 Assessing risks	303
	37.3 Protect the source.....	304
	37.4 Protecting recipients	305
	37.5 Protecting confidential files.....	305
VI	Tools	309
<hr/>		
	Introduction	311
	38 Installing and configuring the Tor Browser	313
	38.1 Install Tor Browser	314
	38.2 Tor Browser update	314
	39 Browsing the web with Tor	315
	39.1 Go to Tor Browser download folder	315
	39.2 Geolocation limits	316
	40 Choosing web hosting	319
	40.1 A few selection criteria	319
	40.2 Content type.....	320
	40.3 In practice	322

41 Verify an electronic certificate	323
41.1 Verify a certificate or a certification authority.....	323
41.2 Find the fingerprint of a certificate already installed	326
42 Using a visual keyboard in Tails	327
43 Configuring and using the Thunderbird mail client	329
43.1 Launch Thunderbird	329
43.2 Configuring onion routing for Thunderbird	329
43.3 Set a master password in Thunderbird.....	330
43.4 Setting up a mail account.....	330
43.5 Advanced Thunderbird configuration	331
44 Using OpenPGP encryption in Thunderbird	333
44.1 Creating a key pair.....	333
44.2 Exporting and sharing our public key.....	336
44.3 Importing, verifying and exporting public keys	337
44.4 Key pair management: extend, change, revoke your key pair	340
44.5 Encrypt and/or sign emails in Thunderbird.....	342
45 Using OpenPGP encryption in the office	343
45.1 Importing a key into the office keychain	343
45.2 Signing a key.....	344
45.3 Verifying a digital signature.....	345
45.4 Signing data.....	346
45.5 Encrypting data	347
45.6 Decrypting files	348
46 Using instant messaging with OTR	351
46.1 Install the Pidgin instant messaging client.....	351
46.2 Launch Pidgin.....	351
46.3 Setting up an e-mail account.....	352
46.4 Create an XMPP instant messaging account.....	352
46.5 Encrypting the server connection.....	352
46.6 Activate the OTR (<i>Off-the-Record</i>) plugin	352
46.7 Setting up a private conversation.....	353
47 Managing passwords	355
47.1 Choosing a good passphrase.....	355
47.2 Using a password manager	355
48 Using OnionShare	359
48.1 Using OnionShare in Tails	359
48.2 Using OnionShare in Debian.....	359
Who is speaking?	361
Index	363
Credits	367

Preface to this edition

Since the second paper edition of the *guide* was published in 2017, new informations about digital espionage technologies, the tools we recommend or the laws we suffer have emerged.

*
* *

Let's start with a trip to the dark side of digital novelties.

"Data determines everything we do" ¹ Cambridge Analytica's cynical slogan. This company was at the heart of a scandal that exposed the power of the big Internet groups over society: it siphoned off the personal data of tens of millions of Facebook accounts with the platform's consent for a so-called "scientific study". It then sold its targeted psychological manipulation services, which were used to influence the 2015 presidential campaign in Nigeria ²the 2016 US presidential election and the Brexit vote. ³ The scandal was revealed in the media in 2018 thanks to the revelations of a former employee.

With the COVID-19, new milestones have been reached in the abuse of digital technology. "In the age of COVID-19, connectivity is not a commodity, but a necessity. Virtually all human activities - commerce, education, health, politics, socialization - seem to have moved online. [...] States and non-state actors in every country are now exploiting the opportunities created by the pandemic to shape new online narratives, censor critical discourse and build new technological systems of social control." ⁴ These are the findings of the digital freedom organization Freedom House.

The French health pass, for example, is a digitally signed, two-dimensional barcode containing the person's name and health status, as well as information on vaccinations received and the dates of recent injections. Not only is this information readable in plain text, but it also enables "even more private health information to be inferred about certain citizens". ⁵ Beyond these criticisms of

1. "Data drives all we do" in English, quoted by Le Monde (*Le Monde*, 2018, *Ce qu'il faut savoir sur Cambridge Analytica, la société au-c-ur-du-scandale Facebook* [https://www.lemonde.fr/pixels/article/2018/03/22/ce-qu-il-faut-savoir-sur-cambridge-analytica-la-societe-au-c-ur-du-scandale-facebook_5274804_4408996.html]).

2. Wikipedia, 2022, *Christopher Wylie* [https://fr.wikipedia.org/wiki/Christopher_Wylie].

3. Wikipedia, 2021, *Facebook-Cambridge Analytica scandal* [https://fr.wikipedia.org/wiki/Scandale_Facebook-Cambridge_Analytica].

4. Adrian Shahbaz and Allie Funk, 2020, *The Pandemic's Digital Shadow*, in Freedom House, 2020, *Freedom of the Net 2020* [https://freedomhouse.org/sites/default/files/2020-10/10122020_FOTN2020_Complete_Report_FINAL.pdf].

5. Florian Maury, Piotr Chmielnicki, 2021, *Health Pass and privacy: what are the risks?* [<https://www.broken-by-design.fr/posts/pass-sanitaire/>].

the way in which the health pass functions, its mass adoption accustoms the population to submit to widespread control via digital tools ⁶.

Célia Izoard denounces "under the mask of Covid, the complete digitization of society". ⁷ which "continues, daily brutalizing the outdated and refractory". ⁸. For her, the question that health policies too often answer is "how can France use the pandemic to consolidate its technological and economic leadership on the international stage? In a report by the French Senate, we read: "The prospects opened up by the use of digital technologies are immense, and the Covid-19 crisis only gave a foretaste of the many possible uses. [It would be irresponsible not to seize such opportunities. ⁹

The European Union's borders offer a glimpse of what these "opportunities" could be. With the E-Borders program, "digital spaces for collecting data on migrants are at the heart of the European partners' strategy". ¹⁰. Frontex, the European border and coast guard agency, prides itself on making ever greater use of "constantly evolving" technologies. ¹¹ automation, robotization and artificial intelligence. ¹².

The use of these frontiers as laboratories echoes the growing use in judicial investigations of digital surveillance methods, which until recently were reserved for counter-terrorism: the use of keyloggers, decryption of hard drives, and so on. For example, in an investigation targeting the anti-nuclear movement around Bure, dozens of computers and telephones were investigated. ¹³. According to one magistrate, "the exceptional nature of the investigative measures, with highly advanced technologies and wiretaps, stems from all the impasses of criminal conspiracy."¹⁴

Following the same logic, states seem to be mounting attacks using as-yet-unknown security flaws (so-called "*zero-day* vulnerabilities") more and more massively. These were used by China against the Uyghurs in 2018, prompting a US digital liberties organization to say that "it's highly likely that this won't be the last time we see a state actor target an ethnic or activist group en masse thanks to vulnerabilities...".

6. La Quadrature du Net, 2021, *Passe sanitaire : quelle surveillance redouter ?* [<https://www.laquadrature.net/2021/08/19/passe-sanitaire-quelle-surveillance-redouter/>].

7. Célia Izoard, 2021, *Sous le masque du Covid, la numérisation intégrale de la société*, Repor- terre [<https://reporterre.net/Sous-le-masque-du-Covid-la-numerisation-integrale-de-la-societe>].

8. Célia Izoard, 2021, *La numérisation du quotidien, une violence inouïe et ordinaire*, Repor- earth [<https://reporterre.net/La-numerisation-du-quotidien-une-violence-inouie-et-ordinaire>].

9. Véronique Guillotin, Christine Lavarde, René-Paul Savary, 2021, *Rapport d'information fait au nom de la délégation sénatoriale à la prospective sur les crises sanitaires et outils numériques: répondre avec efficacité pour retrouver nos libertés*, French Senate [<https://www.senat.fr/rap/r20-673/r20-6731.pdf>], p. 51.

10. Catherine Puzzo, 2018, *Multiple borders and new agents of migration control in the UK*, Sciences & Actions Sociales n° 9, p 20 [<https://www.cairn.info/revue-sciences-et-act-ions-sociales-2018-1-page-18.htm#pa20>].

11. Frontex, 2017, *Research and Development in border management* [<https://frontex.europa.eu/media-centre/multimedia/videos/research-and-development-in-border-management-Gloaln>](in English).

12. Piotr Szostak, 2021, *With drones and algorithms, Europe builds a virtual wall against migrants*, Gazeta Wyborcza translated by Courier International [<https://www.courrierinternational.com/article/interview-with-drones-and-algorithms-europe-builds-a-virtual-wall-against-them>].

13. Marie Barbier, Jade Lindgaard, 2020, *L'État a dépensé un million d'euros contre les antinu- cléaires de Bure*, Reporterre [<https://reporterre.net/2-3-L-Etat-a-depense-un-million-d-euros-con-tre-les-antinucleaires-de-Bure>].

14. Laurence Blisson, magistrate and former General Secretary of the Syndicat de la magistrature, quoted by Marie Barbien and Jade Lindgaard (*op. cit.*)

zero-day"¹⁵. Companies such as NSO Group¹⁶ or Cellebrite¹⁷ sell spyware that includes such tools.

Following all these revelations in the media, the protection of digital privacy is more topical than ever. So much so, in fact, that companies are seizing on the issue to offer "secure" services, with their well-known limitations: the desire to make a profit leads them to claim to provide guarantees they are unable to keep. In 2021, the encrypted e-mail provider Protonmail provided the French authorities with information on Youth for Climate activists that it claimed not to be registering.¹⁸ Protonmail subsequently affirmed that it had no way of refusing such a legal request and modified its site, which claimed otherwise¹⁹. In 2021, Proton responded to 4,920 legal requests (out of 6,243 requests received)²⁰.

*
* *

On the legal front at European level, several texts on the law that applies to personal data took effect in May 2018: a regulation that sets the general framework for data protection (RGPD)²¹ as well as a directive applicable solely to files in the criminal sphere (Police-Justice Directive)²². They are supposed to protect personal data by regulating how it can be processed²³ by public administrations and organizations. The RGPD also requires personal data to be stored and processed securely. In practice, the regulation is poorly implemented by organizations, as breaches are very little monitored²⁴. The directive that applies to police and judicial processing, on the other hand, facilitates data transfers between law enforcement agencies at European level and beyond.²⁵

The legal battle over data retention at European level clearly illustrates the limits of these regulations. The Court of Justice of the European Union (CJEU) has twice invalidated the European Data Protection Directive.^{26 27} twice, before finally, on de-

15. Cooper Quintin and Mona Wang, 2019, *Watering Holes and Million Dollar Dissidents: the Changing Economics of Digital Surveillance*, Electronic Frontier Foundation [<https://www.eff.org/deeplinks/2019/09/watering-holes-and-million-dollar-dissidents-changing-economics-digital>], .

16. Le Monde, 2021, *Apple fixes computer flaw linked to Pegasus** spying software* [https://www.lemonde.fr/pixels/article/2021/09/14/apple-repare-une-faible-informatique-liee-au-logiciel-d-espionnage-pegasus_6094541_4408996.html].

17. Privacy International, 2012, *Surveillance Company Cellebrite Finds a New Exploit: Spying on Aylum Seekers* [<https://privacyinternational.org/fr/node/2776>].

18. Gaspard d'Allens, 2021, *With its reputation for security, Protonmail gave the police information about climate activists*, Reporterre [<https://reporterre.net/Repute-sur-Protonmail-a-livre-a-la-police-des-information-on-climate-militants>].

19. Emma Confrere, 2021, *Émoi après que la messagerie sécurisée ProtonMail a collaboré à une enquête judiciaire*, Le Figaro [<https://web.archive.org/web/20210916174658/https://www.lefigaro.fr/secteur/high-tech/me-after-secure-messaging-protonmail-a-collabore-a-a-enquet-e-judiciary-20210907>].

20. Proton, 2022, *Transparency Report* [<https://proton.me/legal/transparency>].

21. Official Journal of the European Union, 2016, *Regulation (EU) No. 2016/679 of April 27, 2016, on the protection of individuals with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC* [<https://eur-lex.europa.eu/legal-content/FR/TXT/HTML/?uri=CELEX:32016R0679>].

22. Journal officiel de l'Union Européenne, 2016, *Directive No. 2016/680 of April 27, 2016* [<https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016L0680>], known as the "Police-Justice Directive".

23. Processing" includes anything related to the collection, aggregation, use or sharing of data.

24. La Quadrature du Net, 2021, *Les GAFAM échappent au RGPD, la CNIL complice* [<https://www.laquadrature.net/2021/05/25/les-gafam-echappent-au-rgpd-avec-la-complicite-de-la-cnil/>].

25. La Quadrature du Net, 2016, *Summary of the Personal Data Directive* [https://wiki.laquadrature.net/Synth%C3%A8se_de_la_directive_sur_les_donn%C3%A9es_personnelles#Transfers_and_C3.A9changes_de_donn.C3.A9es_personnelles].

26. Court of Justice of the European Union, 2014, *The Court of Justice declares the Directive on the invalid data retention*, Press release no. 54/14 [<https://curia.europa.eu/jcms/upload/docs/application/pdf/2014-04/cp140054en.pdf>] on the "Digital Rights" ruling.

27. Court of Justice of the European Union, 2016, *Member States may not impose a general data retention obligation on communications service providers*

page

29

France's request²⁸, to partially reverse its position²⁹. France took advantage of the opportunity to maintain the systematic retention of connection logs for national security purposes or to track down the perpetrators of criminal offences.³⁰ Belgium, on the other hand, has confirmed that it is essentially abandoning the generalized and undifferentiated storage of connection data.³¹

The other new regulations applicable in France are too numerous to list here.³² As this is not the main purpose of this book, we will limit ourselves to citing two examples. The authorities' power of censorship has been extended, with the ability to remove web content for "terrorist content" in less than an hour.³³ Or the umpteenth sleight of hand used to announce the end of the state of emergency, while at the same time normalizing some of its exorbitant provisions under ordinary law.³⁴ An indefinite extension of the state of emergency".³⁵

Once again, despite the spread of a feeling of powerlessness, these various revelations on the state of digital surveillance make it all the more necessary to equip ourselves with the means to understand it and adapt our practices accordingly.

*
* *

Since the last edition of the *guide*, technical developments have also led to the updating of several passages: the widespread use of SSD disks means that data deletion has to be rethought; processor technologies have evolved.³⁶ As for web animations, *Flash* technology, which posed numerous privacy problems, has been abandoned in favor of HTML5 and various associated technologies... which pose new problems.

As far as attacks are concerned, the updates concern firmware (e.g. Intel's Management Engine) and data exfiltration using flaws in web services.

électroniques, Press release no. 145/16 [https://curia.europa.eu/jcms/jcms/p1_268807/fr/] on the "Tele2" affair.

28. French Council of State, 2018, *Reading of July 26, 2018* [<https://www.conseil-etat.fr/fr/ari-aneweb/CE/decision/2018-07-26/394922>].

29. Court of Justice of the European Union, 2020, *The Court of Justice confirms that the right to the Union precludes national legislation requiring a provider of electronic communications services, for the purposes of combating crime in general or safeguarding national security, to transmit or store data on a general and undifferentiated basis*

relating to traffic and location, Press release no. 123/20 [<https://curia.europa.eu/jcms/upload/docs/application/pdf/2020-10/cp200123en.pdf>] on the Privacy International, La Quadrature du Net, French Data Network and Ordre des barreaux francophones et germanophone cases.

30. French Council of State, 2021, *Decision of April 21, 2021* [<https://www.conseil-etat.fr/conten-t/download/159464/file/393099.pdf>].

31. Belgian Constitutional Court, 2021, *Ruling no. 57/2021 of April 22, 2021* [<https://www.const-court.be/public/f/2021/2021-057e.pdf>].

32. Non-exhaustive list: Act No. 2019-222 of March 23, 2019 on the 2018-2022 programming and reform for justice, law no. 2019-1479 of December 28, 2019 on finance for 2020, law no. 2020-766 of June 24, 2020 aimed at combating hateful content on the Internet, ordinance no. 2020-1733 of December 16, 2020 on the legislative part of the code on the entry and residence of foreigners and the right of asylum, law no. 2021-646 of May 25, 2021 for global security preserving freedoms, law no. 2021-1109 of August 24, 2021 reinforcing respect for the principles of the Republic, law no. 2021-998 of July 30 2021 on the prevention of acts of terrorism and intelligence...

33. The government fought tooth and nail to get it. *La Quadrature du Net, 7 mai 2021, Règlement de censure terroriste adopté : résumons* [<https://www.laquadrature.net/2021/05/07/reglement-de-censure-terroriste-adopte-resumons/>].

34. *Developpez.com, 2017, France: deputies approve the seizure of computer hardware and the copying of a suspect's data* [<https://www.developpez.com/actu/162736/France-les-deputes-approuvent-la-saisie-de-materiel-informatique-et-la-copie-de-donnees-d-un-suspect-dans-le-cadre-de-la-lutte-contre-le-terrorisme/>].

35. *Commission nationale consultative des droits de l'Homme, July 6, 2017, opinion on the bill strengthening internal security and the fight against terrorism* [<https://www.legifrance.gouv.fr/jorf/id/JORFTEXT000036039262>].

36. Particularly with the disappearance of 32-bit processors.

The explanations of Tor have been extensively revised, as Tor no longer claims to provide anonymity, but rather confidentiality. Recommendations for choosing passphrases and using passwords have also been revised to take account of new research on the subject.

On the tools side, two new versions of the Debian GNU/- Linux operating system have been released, as well as new versions of the Tails *live* system. This update of the *guide* is based on Debian 11 "Bullseye", which has brought many changes to both the graphics and the software on offer. The tools have therefore been revised to ensure that the recipes work on these new systems. This has also led to a number of changes, including an overhaul of the use of the OpenPGP *chif-frement* in Thunderbird and new instructions for using the Tor Browser.

Smartphones are increasingly being targeted in police investigations such as the one at Bure³⁷ and many of the devices exploiting *zero-day* vulnerabilities target *smartphones* in particular. However, this guide does not deal with risk reduction when using smartphones. A complete work would be necessary, for which the people updating this *guide* have neither the time nor the expertise. The very operation of mobile telephony raises difficult privacy issues.³⁸

We need to take these new developments into account in our approach to the digital world and our security policies.

*
* *

Beyond the technical evolutions, a new writing dynamic around the *guide* has led to more general developments.

A complete re-reading has been carried out, resulting in numerous rewordings and updated examples. A new section on risk reduction applied to digital tools has been added to the choice of responses adapted to each situation. The *Working on a sensitive document* use case has been reworked, and the *Publish a document* use case now includes the protection of people who are going to consult it.

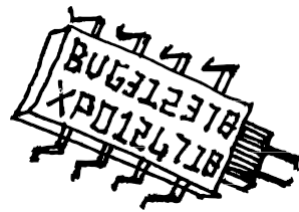
The question of gender in the *guide's* formulations has been around since its inception. For this edition, as much effort as possible has been made to use epicene writing. But the French language is so discriminating that finding non-gendered formulations is not always possible. In such cases, we decided to use the feminine, going against the usual rules specifying the use of the masculine. However, as we write this preface and justify our choices, we realize that it doesn't allow transgender or non-binary people to feel included, even within the *guide* team. With only a few months to go before the book's release, we unfortunately can't consider doing this work again. We'll wait for the next edition, if it comes, to find an inclusive solution.

*
* *

Thanks to this revision, we hope that the following pages will continue to be a wise companion as you traverse the digital jungle... at least, until the next one.

37. Marie Barbier, Jade Lindgaard, 2020, *La justice a massivement surveille les militants antinu-cléaires de Bure*, Reporterre [<https://reporterre.net/1-3-La-justice-a-massivement-surveille-les-mili-tants-antinucleaires-de-Bure>].

38. Surveillance Self-Defense, 2018, *The Problem with Mobile Phones* [<https://ssd.eff.org/en/module/le-probl%C3%A8me-with-the-t%C3%A9l%C3%A9phones-portables>].



Why this guide?

The downside of digital memory

These days, computers, the Internet and cell phones tend to take up more and more space in our lives. Digital technology often seems very practical: it's fast, you can talk to lots of people far away, you can have your whole story in photos, you can easily write well-formatted texts... but it doesn't only have advantages; or at least, it doesn't only have them for us, but also for other people we don't necessarily want to help.

It's much easier to eavesdrop on conversations via cell phones than on a noisy street, or to find the information you need on a hard drive than on a shelf overflowing with papers.

What's more, a great deal of our personal information ends up being published somewhere, whether by ourselves or by others, whether because we're encouraged to do so - that's what *Web 2.0* is all about - or because technologies leave traces, or simply because we're not careful.

Nothing to hide?

"*But don't be paranoid: I've got nothing to hide!*" we might reply to the previous statement...

However, two simple examples tend to show the contrary: nobody wants to see their secret credit card or *eBay* account codes fall into the wrong hands. Nor would anyone want to be burgled because their address has been published on the Internet in spite of themselves, and their absence confirmed on social media.

But beyond these silly questions of defending private property, data confidentiality should be an issue *in itself*.

First of all, because we're not the ones judging what is or isn't allowed to be done with a computer. People are arrested on the basis of the traces left behind by the use of digital tools for activities that didn't please a government, not necessarily their own - and not just in China or Iran.

Many people, be they governments, employers, advertisers or cops³⁹ have a vested interest in gaining access to our data. The growing importance of

39. The term "cops" is used here as defined in the introduction to the *Guide d'autodéfense juridique* : *Face à la police / Face à la justice* [<https://infokiosques.net/spip.php?article538>]: "In this guide, the word 'cop' is used interchangeably with any type of constable or policeman, regardless of rank or status [...]".

information in the global economy and politics can only encourage them. In fact, we already know that they don't mind cross-checking individuals. But what do we know about the legal and illegal practices of those closest to us?

What's more, how can we be sure that what's authorized today will be tomorrow? Governments change, as do laws and situations. And this can happen extremely quickly, as many people saw with the application of the state of ur- gence in France for two years in 2015 ⁴⁰ before certain measures were incorporated into ordinary law. ⁴¹ If we don't have to hide the fact that we regularly visit an activist website, for example, how do we know what will happen if it is linked to a repressive process? Traces *will have been left on the computer...* and could be used as incriminating evidence.

Putting data protection practices in place when we feel we don't need them directly also makes them more "normal", more acceptable and less suspect. People who have no other option for survival than to hide their digital activities will be grateful, no doubt.

Generally speaking, we restrict our actions as soon as we know that others may be listening, watching or reading us. Would we sing in the shower if we knew it was bugged? Would we learn to dance if cameras were pointed at us? Would we write an intimate letter as freely if someone was reading over our shoulder? Having things to hide is not only a question of legality, but also of intimacy.

In this way, control companies see each of us as a potential threat to be monitored. Hiding is therefore a *political* and *collective* issue, if only to put obstacles in the way of those who would like us to be permanently exposed and identifiable.

All this may lead us to think that we don't want to be controllable by any "Big Brother". Whether he already exists or whether we anticipate his emergence, the best thing we can do is to make sure he can't use all the marvellous tools that modern technology offers us - or him - against us.

So let's have something to hide too, if only to cover our tracks!

Understanding so you can choose

This guide is an attempt to describe intimacy (or rather, the lack of it) in understandable terms in the digital world, and to set the record straight on certain preconceived ideas so that we can better understand what we're exposing ourselves to when using a particular tool.

So that you can sort out the "solutions", which can be dangerous if you don't know their limits.

Reading these few pages, you might get the impression that nothing is really safe with a computer; well, it's true. And it's not. There are appropriate tools and uses. And in the end, the question is often not so much "should we or shouldn't we use these technologies?", but rather "when and how should we (or shouldn't we) use them?".

40. Wikipedia, 2017, *State of emergency in France* [https://fr.wikipedia.org/wiki/Etat_d%27urgence_in_France].

41. République Française, 2021, *Loi du 30 juillet 2021 relative à la prévention d'actes de terrorisme et au renseignement* [<https://www.vie-publique.fr/loi/279661-loi-30-juillet-2021-prevention-terror-isme-et-renseignement>].

Taking the time to understand

Software is designed to be as accessible and easy to use as possible. Similarly, the acceleration of computers and Internet connections makes their operation almost instantaneous, almost imperceptible. Thanks to the widespread use of Wi-Fi networks, we no longer even need to connect our devices to cables to exchange data.

This simplification of tools suggests that understanding how they work is superfluous. Unfortunately, this also means we have to trust and delegate many decisions to experts we take at their word. Learning and understanding take time and patience, but they also give us power and autonomy.

How to read this guide

This guide is an attempt to bring together what we've learned from years of practice, mistakes, reflections and discussions, and to share it.

To make everything easier to digest, we've divided everything we wanted to tell into two volumes. As the *offline* part is an essential prerequisite for understanding the issues involved in the *online* part, these two volumes have been brought together in a single book.

An "offline" tome

A first volume, dedicated to *offline* computer use, was released in 2010. Before we even think about connecting our computers, this first installment takes a closer look at how these machines work. We'll see that *the* possibilities for control and surveillance *via* digital tools are innumerable.

An online tome

As its name suggests, this second installment will focus on the use of computers online, i.e. connected to each other. A vast program...

In wealthy countries at least, Internet use has become a way of life. Checking e-mail, downloading files and obtaining information online are now part of everyday life for many of us. Anyone could say that, in a way, they *know what* the Internet *is*. Let's just say that almost everyone is capable of using it for a few common purposes.

Our aim in this second volume, however, will not be to define in minute detail what the Internet is. At the very most, we will provide some sufficient elements of understanding to enable navigation - the ambiguity of the term, which refers as much to "web browsing" as to the possibility of orienting oneself in a complex space with the help of adapted tools.

Casting off

Once again, we're off on a journey into the murky waters of the digital world. Each volume will be divided into three parts. The first will set the scene and explain the basic concepts, providing a general understanding. The second part will deal with typical use cases. Finally, the third and last part will describe in detail the tools needed to implement the security policies discussed in the second part, as well as their uses.

Boxes will provide details that deviate from the text:



PRECISION

This type of box provides examples or additional details that are optional to read.



TO FIND OUT MORE...

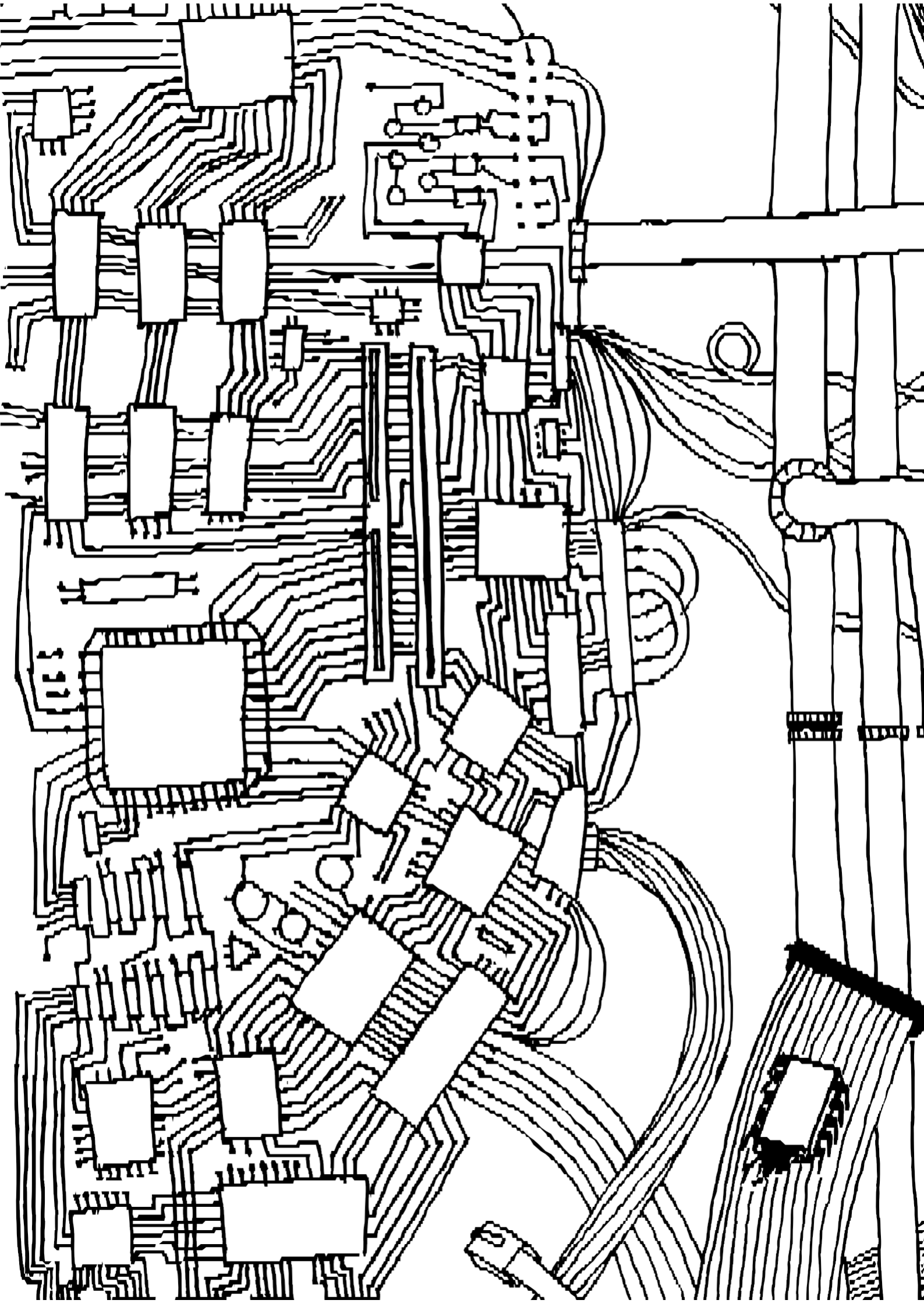
Here, they'll be directions to go further, for those who feel like hacking around off-piste.

*

* *

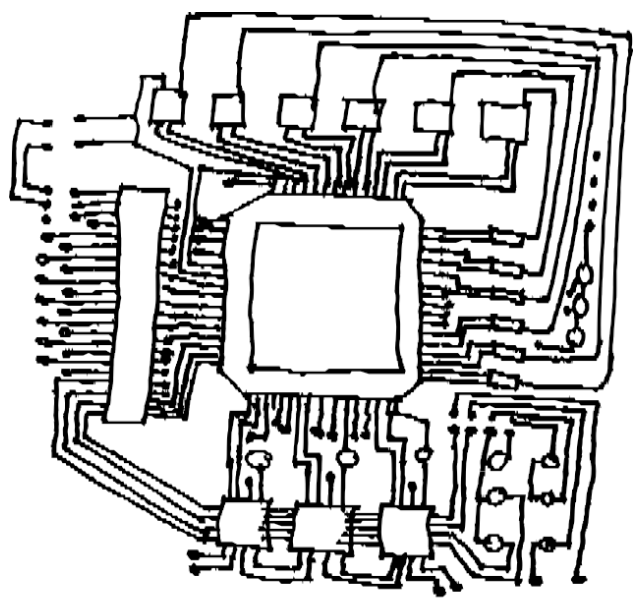
Not only do technologies evolve very quickly, but we may have made mistakes or written untruths in these pages. We will try to keep these notes up to date at <https://guide.boum.org/>.

Adapting our practices to our use of the digital world is therefore necessary if we want, or need, to pay some attention to its impact. But the journey has little meaning alone. So we urge you to build your own digital raft, jump aboard with gusto, and don't forget to take along this guide and a few distress rockets to send your comments and ideas for *use cases* to guide@boum.org.



VOLUME 1

Outside connections



PART ONE

Understanding

Introduction

Given the sheer complexity of computer and digital tools, the amount of information you have to swallow in an attempt to acquire a few self-defense practices can seem enormous. It certainly is for those seeking to understand everything at once...

This first volume will therefore focus on the use of a computer "offline" - we could just as well say *prior to any connection*. But it also covers more general knowledge that applies *whether or not the computer is connected* to a network. So, until the second volume, we'll leave aside the threats specifically linked to the use of the Internet and networks.

For this *off-line* piece, as for the others, we'll take the time to dwell on basic notions, and their implications in terms of security / confidentiality / privacy ¹. After analyzing concrete use cases, we'll look at a few practical recipes.

One final note before we jump in: *the illusion of security is far worse than the clear awareness of weakness*. So let's take the time to read through the first few parts before jumping on our keyboards or throwing our computers out of the window.

1. The idea here is to appeal to a somewhat vague notion: something that would revolve around the possibility of deciding what we reveal, to whom we reveal it, and what we keep secret; something that would also include a certain attention to thwarting attempts to pierce these secrets. The term used here is *privacy*. No French word seems appropriate to convey all the meaning we'd like to put behind this notion. Elsewhere, we often come across the term "security", but its current usage makes us want to avoid it.

Computer basics

First things first.

A *computer* is not a magician's hat where you can put rabbits away and take them out again when you need them, and which can open a window on the other side of the world at the touch of a button.

A computer is a collection of more or less complex components, linked together by electrical connections, cables and sometimes radio waves. All this *hardware* stores, transforms and replicates signals to manipulate the information we see on a beautiful screen with lots of buttons to click.

Understanding how these main components fit together, understanding the basics of what makes them all work, is the first step towards understanding the strengths and weaknesses of these machines, to which we entrust so much of our data.

1.1 Data processing machines

Computers are machines invented to process data. This means they can record, analyze and classify data in very large quantities very quickly.

In the digital world, copying data only costs a few microwatts, in other words, not very much. So we have to consider that *putting information on a computer* (and this is even truer when it's on a network) *means accepting that this information can escape us* without us even realizing it.

This guide can help to reduce the risks, but we have to face up to this reality.

1.2 The equipment

The sum total of interconnected components, our computer is first and foremost an accumulation of objects that we can touch, move, tweak and break.

The *screen/keyboard/tower* (or CPU) combo, or laptop, is handy when you just want to plug the wires in the right places. But to find out what's happening to our data, we need to take a closer look.

We're talking here about the contents of a so-called *personal computer*, sometimes called a PC. But other machines have the same components and are also computers: cell phones, Internet connection "boxes", tablets, MP3 players, cash registers, Linky or Gazpar communicating meters ¹on-board computers, connected objects of all kinds, *etc.*

1. Linky and Gazpar communicating meters are the replacements for historical electricity and gas meters - [Wikipedia, 2021, Compteur communicant \[https://fr.wikipedia.org/wiki/Compteur_communicant\]](https://fr.wikipedia.org/wiki/Compteur_communicant).

1.2.1 The motherboard



A motherboard

A computer is mainly made up of electronic components. The *motherboard* is a large printed circuit board that connects most of these elements through the equivalent of electrical wires. At the very least, a processor, a RAM bar, a storage system (hard disk or other memory), a firmware to start the computer and other cards and peripherals are connected to the motherboard, as required.

Here, we're going to take a quick look at each of these elements to give you an idea of who's doing what, which will come in very handy later on.

1.2.2 The processor



An Intel Pentium 60 MHz microprocessor chip in its housing

The processor (also known as CPU, for *central processing* unit) is the component that handles data processing.

The most concrete example of how a processor works is a calculator. On a calculator, you enter data (numbers) and the operations to be performed on them (addition, multiplication or other) before examining the result. This result can then be used as the basis for further calculations.

A processor works in exactly the same way. Given data (which may be a list of operations to be performed), it executes the requested processes in a chain. That's all it does, but it does it really fast.

But if the processor is just a calculator, how can it process information other than numbers, such as text, images, sound or mouse movement?

Simply by transforming everything that isn't into a number, using a previously defined code. For text, this might be A = 65, B = 66, *and so on*. Once this code has been defined, we can *digitize* our information. With the above code, we can transform "GUIDE" into 71 85 73 68 69.

This series of numbers represents the letters that make up our word. But the digitization process will always result in a loss of information. In this example, we're losing the specificity of handwriting, even though erasures and hesitant letters are just as much "information". Passing things through the sieve of the digital world inevitably means losing bits and pieces.

In addition to data, the operations that the processor must perform (its *instructions*) are also encoded in the form of numbers. A program is therefore a series of instructions, manipulated like any other data.



PRECISION

Inside the computer, all these numbers are themselves represented by electrical states: absence of current or presence of current. So there are two possibilities, those famous 0s and 1s you see everywhere. This is why we speak of binary language (*bi-naire*), whose unit of measurement is the² *bit*. Finally, data processing is carried out with the help of a good bundle of wires and several billion *transistors* (switches, not so different from those used to switch the light on or off in the kitchen).

Not all processors work in the same way. Some have been designed to be more efficient for certain types of calculation, others to consume as little energy as possible, *and so on*. What's more, not all processors have exactly the same instructions. There are large families of them, known as *architectures*. This is important, because a processor with a given architecture will generally only be able to run programs designed for that architecture.

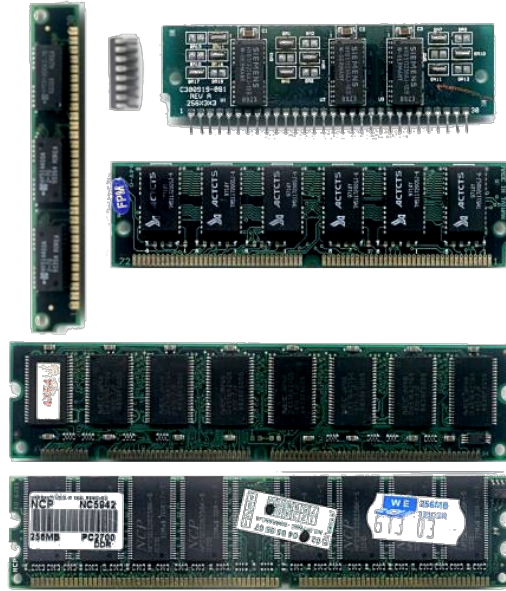
The majority of personal computers run on x86-64 architecture³ architecture (also known as x64, AMD64 or Intel 64), while many phones and other minicomputers run on ARM architecture.

2. For more information, see [Wikipedia, 2014, Bit](https://fr.wikipedia.org/wiki/Bit) [https://fr.wikipedia.org/wiki/Bit].

3. Until the 2010s, some personal computers used an older version of the x86 architecture, where the data manipulated was encoded on 32 bits, compared with 64 bits for the x86-64 version. These are referred to as *32-bit* or *64-bit* processors.

1.2.3 RAM

Random Access Memory (RAM) is often supplied in the form of *memory sticks* and plugged directly into the motherboard.



Different memory modules

RAM stores all software and documents opened when the computer is switched on. This is where the processor fetches the data to be processed and stores the results of operations. Virtually all information processed by the computer therefore passes through RAM in a directly usable - and therefore unencrypted - form.

RAM is connected to the processor, enabling data to be read, written and modified very quickly, according to the processor's needs.

We call it *RAM* as opposed to *read-only memory* (hard disk, USB stick, SSD disk, etc.): unlike these components, the data it contains becomes unreadable after a few minutes or hours (depending on the model) when the RAM is no longer supplied with electricity.

1.2.4 Hard disk or SSD



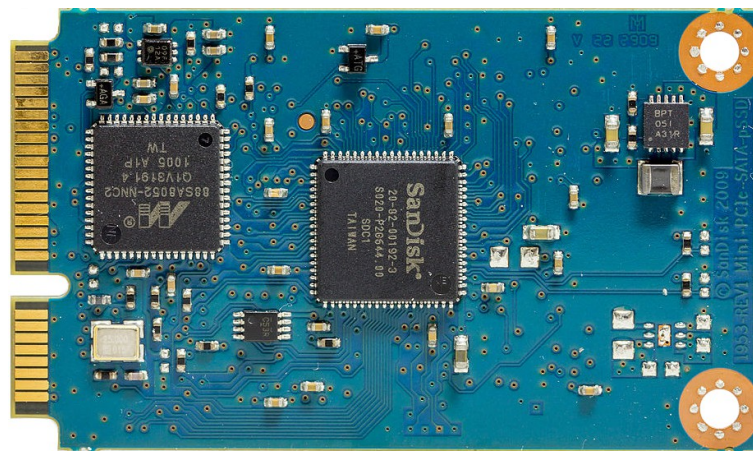
A 3 1/2" hard drive

Since RAM is erased as soon as the power runs out, the computer needs somewhere else to store data and programs between power-ups. This is where *persistent* or *read-only memory* comes into play: a memory where written information remains, even without a power supply.

To do this, we use a storage medium such as a *hard disk* or diskette.

SSD.

The term hard disk generally refers to rotational hard disks, also known as *magnetic hard disks* or *mechanical hard disks*. These rotational hard disks often take the form of a metal shell containing several disks that rotate without stopping, like a miniature turntable. On these discs are tiny pieces of iron, and on top of each disc are *playback heads*. Using magnetic fields, these detect and modify the position of the iron pieces. It's the position of the pieces of iron that codes the information to be stored.



An internal SSD

Because of their mechanical movements, rotational hard drives are slow. That's why, in recent years, more than half the storage media sold have been SSDs or *Solid State Drives* (or electronic disks or solid-state disks), rather than rotational hard disks.⁴ SSDs work with a different type of memory: *flash* memory, the same type used in *USB sticks* and *SD cards*. In an SSD, data is stored using several hundred miniature switches. This all-electronic memory is around 25 times faster than a spinning hard disk.

Rotational hard disks and SSDs can store *much more information* than RAM, but are much slower.

Information is stored in the form of *bits*, of which there are several multiples⁵ — For example, the capacity of a hard disk, often expressed in *gigabytes* (GB), terabytes (TB), etc., can be quantified simply by turning to page 16.

The information stored on a disk (hard disk or SSD) is often documents, but also programs with all the data they need.

such as temporary files, *system logs*, backup files, configuration files, etc.
29 backup files, configuration files, etc.

The disk used therefore retains a quasi-permanent and almost exhaustive memory of all kinds of traces that speak about us, what we do, with whom and how, as soon as we use a computer.

4. Q4, 2021, *SSD Market Share* [<https://www.t4.ai/industry/ssd-market-share/>].

5. Wikipedia, 2017, *Byte* [<https://fr.wikipedia.org/wiki/Octet>].

1.2.5 Other peripherals

With just a processor, some RAM and a storage medium, you've already got a computer. Not very talkative, though. So we usually add other *peripherals*, such as a screen, keyboard, mouse, network card (wired or wireless), micro SD card reader, *etc.*, to the mix.

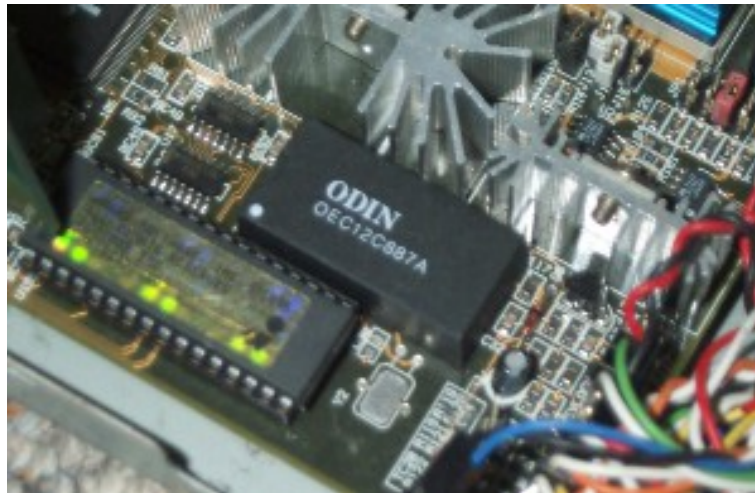
Many of these peripherals are connected via USB (for *Universal Serial Bus*), a standard that enables the connection of printers, keyboards, mice, additional hard drives, network adapters or what are commonly known as "USB sticks".

The link between the processor and the various USB peripherals is provided by a specific set of chips, called a *chipset*. The chipset is soldered to the motherboard, or even integrated into the same housing as the processor.

Most of today's chipsets integrate additional peripherals designed to provide secure environments for the computer's operating system and program execution. These include Intel's Management Engine (ME) and AMD's Platform Security Processor (PSP). These peripherals are often a source of concern, as their operation is opaque and they can sometimes be used as backdoors⁶ on computers equipped with them.

Other peripherals may require the addition of an extra card, known as a *daughterboard*, as is the case with most Wi-Fi adapters.

1.2.6 Motherboard firmware



A firmware chip on a motherboard

To start the computer, you need to give the processor an initial program, so that it can load the programs to be executed next.

This small piece of software, called *firmware*⁷ is contained in a memory chip attached to the motherboard. This is *flash* memory, as in USB sticks or SSD disks.

The historic firmware of most personal computers is called BIOS (*Basic Input/Output System*). Since 2012,

6. These devices run on software that may contain a *backdoor*, i.e. a feature that gives secret access to the software, or even to the computer, without the user being aware of it.

7. Firmware can also be referred to as *firmware*, microcode, internal software or embedded software.

more and more computers are using a new standard called UEFI (*Unified Extended Firmware Interface*).

Among other things, this first program run by the computer allows you to choose the location of the operating system you want to use. It is usually loaded from the hard disk, but can also come from a USB stick, a CD or DVD, or even from the network.

next
page.



TO FIND OUT MORE...

To take a tour of a computer's firmware, you can follow the *Enter firmware configuration interface* in the *Start tool on a CD, DVD or USB stick* (see page 108).

1.3 Electricity, magnetic fields, noise and radio waves

Now that we've had a quick look at what makes it up, let's move on to the confidentiality of information circulating within a computer.

First of all, most information circulates in the form of electrical currents. So there's no reason why you can't install the equivalent of an *ammeter* to measure the current flow, and thus be able to reconstruct the data manipulated by the computer in one form or another.

In addition, any electric current that flows tends to emit a magnetic field. These magnetic fields can radiate to a distance of several metres or more.⁸ It is therefore possible, if you have the means, to reconstruct the contents of a screen or what has been typed on a keyboard, even from behind a wall, from the street or from the adjoining apartment. Researchers have succeeded in recording the keystrokes typed on normal wired keyboards from their electromagnetic emissions, from a distance of up to 20 meters.⁹

The same type of operation is possible by observing the slight disturbances generated by the computer on the electrical network where it is plugged in.¹⁰

Other experiments using a microphone to listen to the noise of the computer's electronic components and its power supply have made it possible, under certain conditions, to decrypt encryption keys contained on the target computer.¹¹ Corrections have since been made to the software involved to complicate this type of attack.

Finally, some peripherals (keyboards, mice, headphones, *etc.*) operate *wirelessly*. They communicate with the computer via radio waves that can be picked up and decoded by anyone in the vicinity.

To sum up, even if a computer isn't connected to a network, and whatever programs are running on it, it's still possible for well-equipped experts to "listen in" to what's going on inside.

8. In 1995, Berke Durak succeeded in capturing the electromagnetic waves [<http://lambda-diode.com/electronics/tempest/>] emitted by most of his computer's components using a simple *walkman* capable of receiving radio (link in English).

9. Martin Vuagnoux and Sylvain Pasini have produced some frightening videos [<https://lasecwww.epfl.ch/keyboard/>] to illustrate their 2009 paper *Compromising Electromagnetic Emanations of Wired and Wireless Keyboards*.

10. In 1998, Paul Kocher, Joshua Jaffe and Benjamin Jun published a report [<https://www.rambus.com/wp-content/uploads/2015/08/DPATechInfo.pdf>] explaining the various power consumption analysis techniques.

11. Clément Bohic, 2013, *Chiffrement : il suffirait d'écouter le processeur pour décoder les clefs*, [silicon.fr](https://www.silicon.fr/chiffrement-ecouter-processeur-decoder-clefs-91686.html) [<https://www.silicon.fr/chiffrement-ecouter-processeur-decoder-clefs-91686.html>].

1.4 Software

Beyond the sum of the physical elements that make up a computer, we also need to look at the less tangible elements: the software.

In the days of the very first computers, each time different processing operations had to be carried out, physical intervention was required to change the layout of cables and components. Today, however, this is no longer the case: the operations required to carry out these processes have become data like any other. This data, known as *programs*, is loaded, modified and manipulated by other programs.

A set of programs for carrying out a given task is called *software*. It is then the interaction of thousands of pieces of software with each other that will enable the complex tasks for which computers are generally used today to be carried out.

The effect produced when a button is clicked is therefore the launch of a chain of events, an impressive sum of calculations that result in electrical impulses that modify a physical object. It's like the vibrations of a loudspeaker membrane to play a sound, a screen that modifies its LEDs to afficher a new page, or an SSD disk that activates or deactivates micro-switches to create the binary sequence of data that will constitute a *file*.

1.4.1 The operating system

The purpose of an *operating system* is first and foremost to enable the various software programs to share access to the computer's hardware components and to communicate with each other. What's more, an operating system generally comes with software, at least to enable other software to be started.

The fundamental part of an operating system is its *kernel*, which coordinates the use of hardware by other software.

For each computer hardware component you want to use, the kernel activates a program called a *driver*. There are drivers for input devices (keyboard, mouse, *etc.*), output devices (screen, printer, *etc.*) and storage devices (DVD, USB key, *etc.*).

The kernel also manages the execution of the various programs, allocating parts of the processor's RAM and computing time to each.

In addition to the kernel, today's operating systems - such as Windows, macOS or GNU/Linux - also include numerous tools (or utilities), as well as graphical desktop environments that let you operate the computer simply by clicking on buttons.

The operating system is usually stored on the hard disk. However, it is also possible to use an operating system stored on a USB stick or burned onto a DVD. In the latter case, we speak of a *live* system.

1.4.2 Applications

Applications are software programs that actually let you do what you want the computer to do. Examples include Mozilla Firefox for web browsing, LibreOffice for office automation and VLC for music and video playback.

Each operating system defines a specific method for applications to access hardware, data, the network or other resources. The applications you want to use must therefore be adapted to the operating system of the computer you want to use them on.

1.4.3 The libraries

Rather than rewriting program chunks in every application to do the same thing, these chunks are grouped together in *libraries*, which software programs share.

There are libraries for graphical affichage (ensuring the consistency of what is affiché on the screen), for reading or writing file formats, for querying certain network services, *and so on*.

If you don't write software yourself, you rarely need to visit these libraries. However, it can be useful to know that they exist, if only because a problem (such as a programming error) in one library can have repercussions on all the software that uses it.

1.4.4 Packages

GNU/Linux operating systems can be organized differently depending on their distribution¹². Some distributions (such as Debian or Tails, on which most of the tools presented in this guide are based) work with *packages*.

[opposite page]

Software (*operating systems, applications or libraries*) is then installed using packages. A package is made up of several files which, among other things, enable the program to be executed, specify whether it depends on other software or other packages, enable it to be configured, provide documentation, verify its authenticity, *etc*.

The system can be run with software that automates the installation, uninstallation and updating of packages. Such software is called a *package manager*. In general, a distribution's packages are available on the Internet in so-called *repositories*. The package manager will then retrieve the necessary packages from these repositories, which are specific to each distribution.

1.5 Data storage

We've seen that a hard disk (or USB key) can be used to store data between computer power-ups.

But to be able to find their way around, data is arranged in a certain way: a cabinet without shelves in which you pile up sheets of paper isn't necessarily the most efficient form of storage.

1.5.1 Sheet music

Just as you can put several shelves in a piece of furniture, you can "slice" a hard disk into several *partitions*.

Each shelf can have a different height and a different classification, depending on whether you want to put books or files on it, in alphabetical or chronological order.

In the same way, on a hard disk, each partition can be of a different size and contain a different organization mode: this is called a *file system*.

12. The distribution of an operating system is a version of it adapted to specific uses or needs. This may be to make it particularly lightweight or easier to use, for example, but also for special functions (for a particular company or tool). Each distribution brings together an adapted and coherent collection of software, from the system to the applications, with varying degrees of functionality.

1.5.2 File systems

A file system is used to retrieve information from our huge pile of data, just as the table of contents of a cookery book takes you straight to the right page to read the recipe for the evening's feast.

Note, however, that deleting a file does not delete the contents of the file, but merely removes a line from the table of contents. If you go through all the pages, you'll always be able to find your recipe, as long as the page hasn't been rewritten - this point will be developed later.

Thousands of different formats can be imagined for storing data, and so there are many different file systems. *Formatting* refers to the creation of a file system defined on a medium.

As it is the operating system that gives programs access to data, a file system is often strongly linked to a particular operating system.



PRECISION

To name just a few: NTFS and FAT32 are the types usually used by Windows operating systems; *ext* (**ext2**, **ext4**) is often used by GNU/Linux; HFS, HFS+ and HFSX are used by macOS.

One of the consequences of this is that there may be storage spaces on a given computer that are not recognized by the operating system, and therefore cannot be easily accessed.

Nevertheless, it is possible to read a file system that is *a priori* unknown to the system you are using, provided you use the appropriate software. Windows, for example, is capable of reading an *ext3* partition, if the appropriate software is installed.

1.5.3 File formats

The data we manipulate is generally grouped together in the form of files. A file has a content - the data - as well as metadata, i.e. a name, a location (the folder it's in), a size, and other details depending on the file system used.

But within each file, the data itself is organized differently, depending on its nature and the software used to manipulate it. To differentiate between them, we speak of file *formats*.

In general, a code, sometimes called an *extension*, is placed at the end of a file name to indicate its format. You can choose one extension or another and modify it. However, this is mainly for information purposes, and does not mean that changing the extension changes the file format.

A few examples of extensions: for music, we'll often use MP3 or Ogg formats; for a LibreOffice text document, it will be OpenDocument Text (ODT); for images, we'll have the choice between JPEG, PNG and others; *and so on*.

Like software, formats can be *open or proprietary*. *Open formats* are publicly defined, in order, among other things, not to restrict their use to a single piece of software.

Some *proprietary* formats have been carefully studied to make them usable by other software, but their understanding often remains imperfect. This is typically the case for the old Microsoft Word format (DOC) or Adobe Photoshop (PSD).

page
42

page 30

page 39

1.5.4 Virtual memory (*swap*)

In theory, all the data the processor needs to access, and therefore all the programs and documents it opens, should be in RAM. But in order to be able to open lots of programs and documents at the same time, modern operating systems cheat: when necessary, they swap pieces of RAM with a dedicated space on the hard disk. This is known as "virtual memory" or "*swap* space".

So the operating system does its little cooking to ensure that the processor always has the data it really wants to access in RAM. Virtual memory is an example of storage space that we don't necessarily think of, saved on the hard disk either as a large contiguous file (with Windows and sometimes GNU/Linux), or in a separate partition (with GNU/Linux).

We'll come back to the problems posed by these questions of format and storage space from the point of view of data confidentiality in the next section.

Traces on every floor

The normal operation of a computer leaves many traces of what you do on it. Sometimes, this information is *necessary* for the computer to function. At other times, this information is collected to enable software to be "more practical".

2.1 In RAM

As we have seen, the first place where information is stored on a computer is the memory card. RAM. -----

page 18

As long as the computer is powered up, it contains all the information the system needs. It therefore necessarily retains many traces: keystrokes (including passwords), files opened, and other various events that punctuated the computer's wake-up phase.

By taking control of a computer that's switched on, it's not very difficult to transfer all the information contained in the RAM, for example to a USB stick or to another computer over the network. And taking control of a computer can be as simple as plugging in a cobbled-together *iPod* when the owner's back is turned. ¹. Once recovered, the vast amount of information contained in RAM - about who is using the computer, for example - can then be exploited.

We've also seen that this data becomes unreadable after the computer is switched off. Nevertheless, this takes more or less time, and it can suffice for an ill-intentioned person to have time to recover what's there. This is known as a *cold boot attack*: the idea is to copy the contents of RAM before it has had time to erase itself, so as to exploit it later. It's even technically possible to bring the memory of a freshly shut-down computer to a very low temperature, in which case its contents can survive for several hours or even days. ².

However, this attack must be carried out very soon after power-down in order to work. In addition, if you use a few large programs (e.g. retouching a huge image with Adobe Photoshop or GIMP) before shutting down your computer, the traces you have previously left in RAM are likely to be overwritten. What's more, there are software programs specially designed to overwrite RAM contents with random data just before shutting down the computer.

1. Fernand Lone Sang, Vincent Nicomette, Yves Deswarte, Loïc Duflot, 2011, *DMA peer-to-peer attacks and countermeasures* [https://www.sstic.org/media/SSTIC2011/SSTIC-actes/attaques_dma_peer-to-peer_et_countermeasures-lone-sang_duflot_nicomette_deswarte.pdf]. Maximilian Dornseif, 2004, *Owned by an iPod* [<https://web.archive.org/web/20100326020818/http://md.hudora.de/presentations/#firewire-pacsec>].

2. J. Alex Halderman *et al*, 2008, *Lest We Remember: Cold Boot Attacks on Encryption Keys* [<https://citp.princeton.edu/memory/>].

47

[page
25

[page
44

[page
47

[page
47

[page
25

[page

2.2 In virtual memory

uses part of the hard disk to support its RAM. This happens particularly when the computer is heavily used, for example when working with large images, but also in many other cases, in unpredictable ways.

The most annoying consequence of this very practical operation is that the computer will write potentially sensitive information from RAM to the hard disk, *which will remain readable after the computer has been switched off*.

With a computer configured in the standard way, it is therefore illusory to believe that a document read from a USB key, even when opened with portable software, will never leave traces on the hard disk.

To prevent anyone from accessing this data, it is possible to use an operating system configured to encrypt virtual memory.

2.3 Watch and hibernate

Most operating systems allow you to "pause" a computer. This is mostly used with laptops, but is also valid for desktops.

There are two main types of "pause": wakefulness and hibernation.

2.3.1 The day before

Standby (also known as *suspend to RAM* or *suspend*) consists of turning off as many computer components as possible, while keeping the power on for a quick restart.

At the very least, RAM will continue to be used to store all the data you were working on - including passwords and encryption keys.

In short, a computer in standby mode offers as little protection against data access as one that is switched on.

2.3.2 Hibernation

Hibernation, also known as *suspend to disk*, involves saving all RAM to the hard disk and then switching off the computer completely. The next time it starts up, the operating system detects the hibernation, copies the backup to RAM and starts working again from there.

On GNU/Linux systems, memory is usually copied into virtual memory (*swap*). On other systems, it may be in a large file, often *hidden*.

Since the entire contents of RAM are then written to the hard disk, this means that all open programs and documents, passwords, encryption keys and so on, can be retrieved by anyone accessing the hard disk. As long as nothing has been rewritten over it.

However, this risk is limited by hard disk encryption: the passphrase is then required to access the RAM backup.

A
s

e
x
p
l
a
i
n
e
d

e
a
r
l
i
e
r
,
t
h
e

o
p
e
r
a
t
i
n
g

s
y
s
t
e
m

s
o
m
e
t
i
m
e
s

u

2.4 Newspapers

Operating systems write a detailed history of what they do in their *system logs*.

These logs help the operating system to function, and may enable us to correct configuration problems or *bugs*.

However, these logs also store data that may raise privacy issues. For example, they record :

- the date, time and nickname of the user who logs in every time the computer is switched on;
- make and model of any removable media (external disk, USB key, etc.) connected;
- print date and number of pages ;
- software name, date and time of installation or uninstallation of an application.

By default, all these logs are stored indefinitely on the computer's hard disk, except in the case of *live* systems, which store them in RAM.

2.5 Automatic backups and other lists

In addition to these logs, it is possible that other traces of even deleted files remain on the computer. Even if the files and their contents have indeed been deleted, some part of the operating system or other program may deliberately keep a trace of them.

Here are a few examples:

- a word processor may keep a reference to a deleted file name in the "recent documents" menu. Sometimes, it may even gather temporary files with the contents of the file in question. There are dozens of programs that work in this way;
- when using a printer, the operating system often copies the pending file into the "print queue". Once the queue has been emptied, the contents of this file will not have disappeared from the hard disk;
- under Windows, when a removable drive (USB stick, external hard drive, SD card or DVD) is connected, the system often first scans its contents to suggest suitable software for reading: this automatic scanning leaves a list of all the files present on the media used, even if none of the files it contains are consulted.

It is difficult to find an adequate solution to this problem. A file, even if perfectly deleted, will probably continue to exist on the computer for some time in a different form. A search of the raw data on the disk will show whether or not copies of this data exist, unless they are only referenced there or stored in a different form, such as compressed.

In fact, only by overwriting the entire disk and installing a new operating system can you be sure that all traces of a file have been removed. On the other hand, the use of a *live* system, whose development team pays particular attention to this issue, guarantees that these traces will not be left anywhere other than in RAM. More on this later.

page 139

page 113

2.6 Metadata

In addition to the information contained in a file, there is also information accompanying it, which is not necessarily visible at first glance: date of creation, name of software used, computer, *etc.* This "data about the data" is commonly called "metadata". This "data about data" is commonly referred to as "metadata".

[page 24] Part of the metadata is recorded by the file system: the name of the file, the date and time of its creation and modifications, and often many other things. These traces are left on the computer (which can still be a problem in itself), but most of the time they are not written to the file.

[page 24] On the other hand, many file formats also store metadata *inside* the file. This metadata will be distributed with the file when it is copied onto a USB key, sent by e-mail or published online. This information can then be known by anyone who has access to the file.

The metadata recorded depends on the format and software used. Most audio file formats allow you to record the song title and performer. Word processors or PDFs will record the name of the author, the date and time of creation, and sometimes even the complete history of the last modifications.³so, potentially, information you thought you'd deleted.

Image formats such as TIFF or JPEG probably take the cake: these photo files created by a digital camera or cell phone contain metadata in EXIF format. This can include the brand, model and serial number of the camera used, as well as the date, time and sometimes even the geographical coordinates of the shot, not forgetting a mid-nature version of the image. It was this metadata that ended the run of John McAfee, founder and former head of the computer security company of the same name.⁴ What's more, all this information tends to remain even after the file has gone through photo editing software. The case of the thumbnail is particularly interesting: many photos available on the Internet still contain the entirety of a cropped photo, or even "blurred" faces.⁵ or even "blurred" faces.⁶

[page 185] For most *open* file formats, however, software is available to examine and possibly remove metadata.

[page 24]

3. Deblock Fabrice, 2006, *Quand les documents Word trahissent la confidentialité* [<https://web.archive.org/web/20190913142445/http://www.journaldunet.com/solutions/0603/060327-indiscretions-word.shtml>].

4. Big Browser, 2012, *Vice de forme - The blunder that led to John McAfee's arrest* [https://www.lemonde.fr/big-browser/article/2012/12/12/vice-de-forme-la-bourde-qui-a-mene-a-l-arrestation-de-john-mcafee_5986399_4832693.html].

5. There are even metadata search engines, such as *Stolen Camera Finder* [<https://www.stolencamerafinder.com/>], for example.

6. Maximillian Dornseif and Steven J. Murdoch, 2004, *Hidden Data in Internet Published Documents* [<http://events.ccc.de/congress/2004/fahrplan/files/316-hidden-data->

slides.pdf].

Malicious software, bugs and other spies

Every operating system leaves traces, at least when it's running. Beyond these traces, we can also find a whole host of *bugs* on our computers. They may be installed without our knowledge (enabling us, for example, to retrieve passwords or e-mail contacts), or they may be systematically present in the software we install.

These bugs can be used in a variety of surveillance techniques, from

From the "fight against piracy" of proprietary software, to the collection of data for spam and other scams, to the targeted registration of individuals, page 39. and other scams.

The range of these devices increases dramatically as soon as the computer is connected to the Internet. They are easy to install if you do nothing special to protect yourself, and the data collected can be retrieved remotely.

However, the people who gather this information are unequally dangerous: it depends on the case, their motives and their means. Perpetrators of domestic violence ²GAFAM ³ who track Internet users' data for advertising purposes, the gendarmes of Saint-Tropez, or the US *National Security Agency*... all these people or structures are often in competition with each other and do not form a coherent whole.

To gain access to our computers, they don't all have access to the same means or the same tools: for example, industrial espionage is a major reason for more or less legal surveillance. ⁴and, despite appearances ⁵don't think that Microsoft is giving all the tricks of the trade to the French police.

3.1 Legal context

However, French cops and intelligence services now have the means to set up comprehensive computer surveillance in complete legality, using several of the "bugs" presented below.

1. *Spam* is unsolicited electronic communication, usually .

2. Catherine Armitage, 2014, *Spyware's role in domestic violence* [<https://www.theage.com.au/technology/technology-news/spywares-role-in-domestic-violence-20140321-358sj.html>] talks about the use of *malware* and other technological tools by perpetrators of domestic violence (in).

3. GAFAM is the acronym for the five major US corporations - Google, Apple, Facebook, Amazon and Microsoft - which dominate the digital market.

4. To get an idea of the issues surrounding industrial espionage: Wikipedia, 2014, *Industrial Espionage* [https://fr.wikipedia.org/wiki/Espionnage_industriel].

5. Microsoft, in partnership with Interpol, has produced a toolbox called COFEE (Computer Online Forensic Evidence Extractor) available to police forces in fifteen countries. Korben, 2009, *Cofee - La clé sécurité de Microsoft vient-d'apparaître sur-la-toile.html* [<https://korben.info/cofee-la-cle-securite-de-microsoft-vient-d-apparaître-sur-la-toile.html>].

The 2016 law "strengthening the fight against organized crime, terrorism and their financing, and improving the efficiency and guarantees of criminal procedure"⁶ includes legal provisions that allow the installation of bugs to record and communicate what is affected on the screen or what the various peripherals (keyboard, webcam, scanner, cell phone...) transmit to the computer.

The "installation" of these bugs is authorized, either remotely or by entering the home of the person being monitored to install the necessary tools.⁷ The juge des libertés et de la détention (liberty and custody judge) may request this during preliminary and flagrante delicto investigations; the juge d'instruction (investigating judge) during judicial inquiries.⁸ These measures apply not only to acts of "terrorism" (such as the "proliferation of weapons of mass destruction"), but also to a number of offenses committed by several people ("organized gangs"). These can range from aiding "the illegal movement and residence of a foreigner in France", to the "criminal organization" of a "criminal organization".⁹

The 2015 Intelligence Act¹⁰ gives more or less the same powers¹¹ to "specialized intelligence services" to "research, collect, exploit and make available to the Government intelligence relating to geopolitical and strategic issues, as well as threats and risks likely to affect the life of the Nation".¹²

3.2 Malware

Malicious software¹³ (also known as *malware*) is software that has been developed for the purpose of causing harm: gathering information, hosting illegal information, relaying spam, *and so on*. Computer viruses, worms, Trojans, *spyware*, *rootkits* and *keyloggers* are all part of this family. Some programs may belong to more than one of these categories at the same time.

[page
35

3.2.1 Exploiting vulnerabilities

In order to install themselves on a computer, some malware exploit vulnerabilities in the operating system¹⁴ or applications. They rely on design or programming errors to hijack the program flow to their advantage. Unfortunately, such "security flaws" have been found in a great many software programs, and new ones are constantly being found, both by people seeking to correct them and by others seeking to exploit them.

6. Légifrance, 2016, *loi n° 2016-731 du 3 juin 2016 renforçant la lutte contre le crime organisé, le terrorisme et leur financement, et améliorant l'efficacité et les garanties de la procédure pénale* [<https://www.legifrance.gouv.fr/loda/id/JORFTEXT000032627231/>].

7. Légifrance, 2019, *Code de procédure pénale*, article 706-102-1 [https://www.legifrance.gouv.fr/codes/article_lc/LEGIARTI000038311624/2019-06-01/].

8. Légifrance, 2019, *Code de procédure pénale*, article 706-95-12 [https://www.legifrance.gouv.fr/codes/article_lc/LEGIARTI000038270130/2019-06-01/].

9. Légifrance, 2017, *Code de procédure pénale*, articles 706-73 et 706-73-1 [https://www.legifrance.gouv.fr/codes/section_lc/LEGITEXT000006071154/LEGISCTA000006138138/].

10. Légifrance, 2015, *loi n° 2015-912 du 24 juillet 2015 relative au renseignement* [<https://www.legifrance.gouv.fr/loda/id/JORFTEXT000030931899/>].

11. Légifrance, 2017, *Code de la Sécurité Intérieure*, article L853-2 [https://www.legifrance.gouv.fr/codes/article_lc/LEGIARTI000043887476/].

12. Légifrance, 2015, *Code de la Sécurité Intérieure*, article L811-2 [https://www.legifrance.gouv.fr/codes/article_lc/LEGIARTI000030939233/].

13. This whole section is largely inspired by the passage devoted to the subject in the *Surveillance Self-Defense Guide* [<https://ssd.eff.org/fr/module/comment-puis-je-me-prot%C3%A9ger-anti-malware>] from the *Electronic Frontier Foundation*.

14. According to the *Internet Storm Center* [<https://isc.sans.edu/survivaltime.html>], by 2021, an operating system on which security updates have not been installed will be compromised in less than an hour if connected directly to the Internet.

3.2.2 Social engineering

Another common method is to entice the computer user to launch the malware by hiding it in seemingly innocuous software. For example, on a social media site linked to the Syrian revolution, a simple link to a video actually led users to download a virus containing a *keylogger*.¹⁵

Adversaries then have no need to find serious vulnerabilities in common software. It is particularly difficult to ensure that computers shared by many people, or computers in public places such as a library or cybercafé, have not been corrupted: it suffices indeed only takes one slightly less vigilant person to be fooled...

3.2.3 Camouflage

What's more, most "serious" malware leaves no immediately visible signs of its presence, and can even be very difficult to detect. Perhaps the most complicated case is that of previously unknown vulnerabilities, called "zero-day vulnerabilities"¹⁶ These are vulnerabilities that antivirus software would be hard-pressed to recognize, as they have not yet been catalogued. This is exactly the kind of zero-day vulnerability exploitation that VUPEN sold to the NSA in 2012.¹⁷

Malware can be hidden in unsuspected places on the computer: for example, Intel processors have recently included a Management Engine (ME). In 2017, several security vulnerabilities were discovered in the firmware of this management engine¹⁸. They allow the installation of completely undetectable malware that resists operating system updates and has access to the entire processor and RAM.¹⁹

3.2.4 Capabilities

These programs can be used to carry out a wide range of operations: obtain credit card numbers or passwords, send spam, help attack a server by saturating it with requests, and *so on*. They can also use the computer's microphone, webcam or other peripherals. There's a real specialist market where you can buy such programs, customized for different purposes.

But they can also be used to spy on specific organizations or individuals.²⁰ for example, by exfiltrating documents stored on the computer (even encrypted documents, if they have been decrypted at some point), or by destroying⁴⁷ anonymizing devices on the Internet.

15. Eva Galperin *et al*, 2014, *Quantum of Surveillance: Familiar Actors and Possible False Flags in Syrian Malware Campaigns* [https://www.eff.org/files/2013/12/28/quantum_of_surveillance4d.pdf].

16. Wikipedia, 2016, *Zero-day vulnerability* [https://fr.wikipedia.org/wiki/Vuln%C3%A9rabilit%C3%A9_zero-day].

17. Grégoire Fleurot, 2013, *Espionnage : Vupen, l'entreprise française qui bosse pour la NSA* [<https://www.slate.fr/france/77866/vupen-nsa-espionnage-exploits>].

18. Guillaume Louel, 2017, *New Intel ME security flaw!*, Hardware.fr [<https://www.hardware.fr/news/15297/new-security-flaw-intel-me.html>].

19. Mark Ermolov, Maxim Goryachy, 2018, *How to Hack a Turned-off Computer, or Running Unsigned Code in Intel ME*, blackhat.com [<https://www.blackhat.com/docs/eu-17/materials/eu-17-Goryachy-How-To-Hack-A-Turned-Off-Computer-Or-Running-Unsigned-Code-In-Intel-Management-Engine-wp.pdf>] (in English).

20. For example, a targeted attack on Georgian institutions attributed to the Ministry of Justice of Georgia *et al*, 2012, *Cyber Espionage Against Georgian Government* [<https://web.archive.org/web/20200601112146/https://dea.gov.ge/uploads/CERT%20DOCS/Cyber%20Espionage.pdf>].



PRECISION

To give an example from the United Arab Emirates, a human rights activist, Ahmed Mansour, was the victim of a targeted attack on his smartphone²¹. An SMS containing a link to a virus was sent to him. This virus enabled the person controlling it to use the camera, microphone and monitor the victim's phone activities at any time. The attack was thwarted and dissected thanks to Citizen Lab.

page 31 French intelligence services and cops are legally entitled to use such software, which most certainly means they have it. A suite of spyware attributed to the French intelligence services has been discovered in Iran, among other places.²²

3.2.5 Risk and prevention

No one knows how many computers are infected by malware, but some people estimate that 20-50% of Windows computers are infected.²³ So it's highly likely that you'll find malware on any Windows you think you're using. Until now, using a minority operating system (such as GNU/Linux) has significantly reduced the risk of infection, as these systems are less likely to be targeted, as the development of specific *malware* is economically less profitable.

Here are a few ways to limit the risks:

- don't install (or use) any software from unknown sources: don't trust the first website you come across²⁴;
- heed operating system warnings that software is unsafe, or that a security update is required;
- finally, reduce the possibility of installing new software: by limiting the use of the administration account and the number of people with access to it.

3.3 Spy equipment

Adversaries wanting to get their hands on the secrets contained in our computers can use malware, as we've just seen, but they can just as easily use spyware. These gadgets are no match for James Bond's!

page 32

There is a whole range of more or less readily available hardware that can be used to intrude or exfiltrate information from a computer, at virtually any level. Following the publication of confidential NSA documents by Edward Snowden, a veritable catalog of computer espionage was published in the German newspaper *Der Spiegel*.²⁵

21. Andréa Fradin, 2016, "*Pegasus*", *the weapon of a shadowy Israeli firm that makes Apple tremble* [<https://www.nouvelobs.com/rue89/rue89-surveillance/20160826.RUE3689/pegasus-l-arme-d-u-ne-israelian-firm-fantasy-that-shakes-apple.html>].

22. Martin Untersinger, 2015, *Dino, le nouveau programme-espion développé par des francophones*, *Le Monde.fr* [https://www.lemonde.fr/pixels/article/2015/06/30/dino-le-nouveau-programme-espion-developpe-par-des-francophones_4664675_4408996.html].

23. SafetyDetectives, 2021, *Statistics and trends: antivirus and cybersecurity 2021* [<https://fr.safetydetectives.com/blog/antivirus-statistics-en/#review-4>].

24. This advice applies equally to GNU/Linux users. In December 2009, the *gnome-look.org* website distributed *malware* [<https://lwn.net/Articles/367874/>] presented as a screensaver. It was downloadable as a Debian package from along with other screensavers and wallpapers.

25. *Der Spiegel*, 2013, *Interactive Graphic: The NSA's Spy Catalog* [<https://www.spiegel.de/international/world/a-941262.html>].

Without going into an exhaustive list, this catalog includes fake USB connectors, enabling what passes through them to be retransmitted in the form of radio waves; tiny chips installed in the cables linking the screen or keyboard to the computer, doing the same, so that adversaries can pick up what you are typing or seeing from a safe distance. Finally, there's a plethora of spyware installed in the computer, whether on the hard disk, in the memory or on the keyboard.

firmware, etc.

page 20

The picture is not very encouraging: a meticulous check of your computer would require you to dismantle it, with very little chance of reassembling it in such a way that it could function again. One answer might be to keep your computer with you, or in a place you consider safe. That said, not all of this equipment is available to all types of adversary. What's more, there's no evidence that the use of such equipment has become commonplace, whether for reasons of cost, installation, or other parameters.

We're going to take a closer look at *keyloggers*, which can be classified as both spyware and malware.

3.4 Keyloggers, or keystroke recorders

Keyloggers, which can be "hardware" or "software", have the function of stealthily recording everything typed on a computer keyboard, in order to be able to transmit this data to the agency or person who installed them.²⁶

Once in place, their ability to record keystrokes as they are typed on

A keyboard therefore bypasses any encryption device, and gives direct access to phrases, passwords and other sensitive data on page 47.

Hardware *keyloggers* are devices connected to the keyboard or computer. They can look like adapters, expansion cards inside the computer (*PCIe* or *mini PCIe*) or even integrated inside the keyboard.²⁷ So they're difficult to spot if you're not looking for them specifically...

For a wireless keyboard, you don't even need a *keylogger* to recover the keys you've entered: you just need to pick up the waves emitted by the keyboard to communicate with the receiver, and then break the encryption used, which is quite weak in most cases²⁸. From a lesser distance, it's still possible to record and retrieve data.

decode electromagnetic waves emitted by wired keyboards, including page 21 those built into laptops...

Software *keyloggers* are much more widespread, because they can be installed remotely (*via* a network, through malware, or otherwise), and generally do not require physical access to the machine to retrieve the collected data (for example, it can be sent periodically by e-mail). Most of these programs also record the name of the current application, the date and time it was run, and the keystrokes associated with it.

A software *keylogger* was used by the Italian police in 2012 to investigate an anarchist radio station.²⁹ A US police officer was convicted of installing

26. Electronic Frontier Foundation, 2021, *Keylogger* [<https://web.archive.org/web/20220928165559/https://ssd.eff.org/fr/glossary/enregistreur-de-frappe>].

27. Many models are available over the counter, for example: a USB adapter [<https://web.archive.org/web/20210611153039/https://www.ebay.fr/itm/224463276889?hash=item34430dcb59%3Ag%3Ay8QAAOSwFpddVMrk>] or a keyboard chip [<https://web.archive.org/web/20210611153634/https://www.ebay.fr/itm/224463702020?hash=item3443144804%3Ag%3ARWgAAOSwQKdtermjy>].

28. Tom Espiner, 2007, *Microsoft wireless keyboard hacked from 50 metres* [<https://www.zdnet.com/home-and-office/networking/microsoft-wireless-keyboard-hacked-from-50-metres/>].

29. Croce Nera Anarchica, 2018, *Resoconto udienze Scripta Manent aprile-luglio* [<https://www.autistici.org/cna/2018/09/13/resoconto-udienze-scripta-manent-aprile-luglio/>] (in Italian).

a *keylogger* on the work computer of his wife, who worked at the courthouse³⁰.

The only way to spot hardware *keyloggers* is to familiarize yourself with these devices and regularly do a visual check of your machine, inside and out. Even if the NSA catalog published at the end of 2013 reports on the difficulty of finding yourself keylogging devices barely larger than a fingernail. For software *keyloggers*, the leads are the same as for other *malware*.

[page

32

3.5 Digital investigation platforms

The cops have specific hardware and software for extracting and analyzing the contents of disks, USB sticks or SD cards, as well as the contents of the random access memory³¹ running computers. These are supplied by companies specializing in digital forensics³². Their software can, for example, generate a graphical summary of computer usage, search by keyword, restore deleted data or crack passwords. Such platforms also exist for smartphones³³.

[page

18

3.6 Printing problems ?

We thought we'd covered all the surprises our computers have in store for us... but even printers are beginning to have their little secrets.

[page

42

3.6.1 A little steganography

First thing to know: many high-end printers sign their work³⁴. This steganographic signature³⁵ is based on very slight printing details, often invisible to the naked eye, which are inserted into each document. They make it possible to identify with certainty the make, model and, in some cases, serial number of the machine used to print a document. We say "with certainty", because that's what these details are there for: to find the machine from its work.

In fact, this is one of the ways in which the person who in June 2017 released *top-secret NSA* documents on the hacking of the 2016 US election by Russian hackers was found. Marks from the printer used to prime the confidential documents were still present when they were published by *The Intercept* newspaper³⁶.

Other types of wear and tear are also left on documents - and this applies to all printers. Because with age, print heads shift, slight errors appear, parts wear out, and all this gradually builds up a signature specific to the printer. Just as ballistics can identify a firearm from a bullet, so it is possible to

30. Jerome Vosgien, 2019, *Un policier installs a keylogger on his wife's computer* [<https://news.sophos.com/en-fr/2014/01/13/policier-installe-keylogger-ordinateur-epouse/>].

31. Cindy Casey, 2019, *RAM Analysis Memory Forensics* [[https://www.bucks.edu/media/bccc_medialibrary/con-ed/itacademy/fos2019/Casey-RAM-Forensics-\(1\).pdf](https://www.bucks.edu/media/bccc_medialibrary/con-ed/itacademy/fos2019/Casey-RAM-Forensics-(1).pdf)] (in English).

32. Wikipedia, 2021, *Computer forensics* [https://fr.wikipedia.org/wiki/Informatique_%C3%A9_gale].

33. Wikipedia, 2021, *Cellebrite* [<https://fr.wikipedia.org/wiki/Cellebrite>].

34. The *Electronic Frontier* Foundation attempts to maintain a list of indiscreet printer manufacturers and models [<https://www.eff.org/pages/list-printers-which-do-or-do-not-display-tracking-dots>].

35. To learn more about steganography, we recommend reading this *Wikipedia* article, 2014, *Steganography* [<https://fr.wikipedia.org/wiki/St%C3%A9ganographie>].

36. Robert Graham, 2017, *How The Intercept Outed Reality Winner* [<https://blog.erratasec.com/2017/06/how-intercept-outed-reality-winner.html>].

use these defects to identify a printer from a page that has been output.

To partly protect against this, it is interesting to know that impression details do not resist repeated photocopying: photocopying the printed page, then photocopying the resulting photocopy, suffit to make such signatures disappear. On the other hand... we're sure to leave others, as photocopiers have defects, a n d sometimes steganographic signatures, similar to those of printers. In short, we're going round in circles, and the problem becomes choosing *which* traces we want to leave...

3.6.2 Memory, again...

Some printers are suffisient to be closer to a real computer than an ink pad.

They can pose problems on another level, since they are endowed with a RAM: this, like the PC's, will keep track of the documents that page 18 have been processed for as long as the machine is switched on... or until another document overwrites them.

Most laser printers have a memory capacity of around ten pages. More recent models, or those with integrated scanners, can hold several thousand pages of text...

Worse still: some models, often used for large print runs as in the photocopy centers, sometimes have internal hard disks, to which the user has no access, and which also keep traces - and this time, even after switching off.

A few safety illusions

Good. We're beginning to get the hang of the traces we can leave behind involuntarily, and the information that ill-intentioned people could recover.

All that remains now is to challenge a few preconceived ideas.

4.1 Proprietary, open source, free software

As we've seen, software can do a lot of things you don't want it to do. So it's essential to do what you can to reduce this problem. From this point of view, free software is far more trustworthy than proprietary software: we'll see why.

4.1.1 The cake metaphor

To understand the difference between free and proprietary software, we often use the metaphor of a cake. To bake a cake, you need a recipe: a list of instructions to follow, ingredients to use and a transformation process to carry out. In the same way, a software recipe is called "source code". It is written in a language designed to be understood by human beings. This recipe is then transformed into a code that can be understood by the processor, in much the same way as baking a cake gives us the opportunity to eat it.

Proprietary software is only available "ready-to-eat", like an industrial cake without its recipe. It is therefore very difficult to know the ingredients: it can be done, but the process is long and complicated. Besides, rereading a sequence of several million additions, subtractions, reads and writes in memory, to reconstruct its purpose and operation, is far from the first thing you'd want to do on a computer.

Free software, on the other hand, comes with its own recipe for anyone who wants to understand or modify how the program works. So it's easier to know what you're feeding your processor, and therefore what's going to happen to your data.

4.1.2 Proprietary software: blind trust

Proprietary software is therefore a bit like a sealed box: you can see that it does what you want it to, has a nice graphical interface, *etc.* But you can't really know the details of how it works. But you can't really know in detail how it works. We don't know whether it just does what we ask it to, or whether it does other things as well. To find out, we'd need to be able to study how it works, which is difficult to do without its source code... so we're left to trust it *blindly*.

Windows and macOS, the former, are huge hermetically sealed boxes in which other equally hermetically sealed boxes are installed (from Microsoft Office to anti-virus...) that may do a lot of things other than what we ask of them.

In particular, they can provide information that these software programs would gather about us, or even allow access to the inside of the computer. For example, with backdoors included in the software¹ included in the software, which people with the key could use to hack into our computer. Since it's impossible to know how the operating system is written, anything is conceivable.

It's a security illusion to place the confidentiality and integrity of our data in the hands of programs that we're obliged to trust with our eyes closed. And installing other software claiming on its packaging to take care of this security for us, when its operation is no more transparent, cannot solve this problem.

4.1.3 The advantage of having the recipe: free software

The greater confidence we can have in a *free* system like GNU/Linux is mainly due to the fact that we have the "recipe" for making it. But let's not forget that there's nothing magical about it: free software doesn't cast any "protection spells" on our computers.

However, GNU/Linux offers more possibilities for making computers a little safer to use, not least by making it possible to configure the system down to the finest detail. All too often, this requires relatively specialized know-how, but at least it's possible.

What's more, the way open-source software is produced is hardly compatible with the introduction of backdoors: it's a collective production process, rather open and transparent, involving a wide variety of people. So it's not easy for ill-intentioned people to sneak in secret access.

However, beware of software described as *open source*. The latter also give access to their "recipe", but their development methods are more closed and opaque. Modification and redistribution of such software is at worst forbidden, at best formally authorized, but made very difficult in practice. As only the team behind the software will be able to participate in its development, we can assume that in practice nobody will read its source code in detail... and therefore nobody will really check how it works.

This is the case, for example, with TrueCrypt, whose development stopped in May 2014. It was an encryption software whose source code was available, but whose development was closed and whose license restricted modification and redistribution. For our purposes, the fact that software is *open source* should be seen more as a selling point than as a guarantee of trust.

Except... the distinction between free and *open source* software is increasingly blurred: people employed by Intel, Google and others write large parts of the most important free software, and we don't always look closely at what they write. For example, here are the statistics for organizations employing the people who develop the Linux kernel (which is free). They are expressed as a percentage of the total number of lines of source code modified over a given period.² :

1. On the subject of "backdoors", see the [Wikipedia article, 2014, Backdoor](https://fr.wikipedia.org/wiki/Porte_d%C3%A9rob%C3%A9e) [https://fr.wikipedia.org/wiki/Porte_d%C3%A9rob%C3%A9e].

2. Jonathan Corbet, 2021, *Some 5.12 development statistics*, Linux Weekly News [https://lwn.net/Articles/853039/].

Organization	Percentage
Linaro	17,4 %
Intel	11,5 %
Red Hat	5,5 %
Google	4,2 %
(Unknown)	4,2 %
NVIDIA	4,1 %
(None)	3,8 %
Realtek	3,3 %
SUSE	2,9 %
MediaTek	2,9 %
Arm	2,3 %
Marvell	2,2 %
AMD	2,1 %
Pengutronix	2,0 %
<i>etc.</i>	

So it's not impossible that someone who wrote part of the software in a corner, and who is trusted by the "Open Source community", could have integrated malicious bits of code. The NSA (a U.S. intelligence agency) has been thus able to create and have validated a cryptographic standard containing a vulnerability enabling it to bypass the encryption of certain secure protocols.³ [page 47]

If you only use free software delivered by a non-commercial GNU/Linux distribution such as Debian or Tails, this is unlikely to happen, but it is a possibility. In this case, you can rely on the people working on the distribution to study the operation of the programs integrated into it.

Nevertheless, this trust can only be valid if we remain vigilant about what we install on our system. For example, on Debian, the distribution's official packages are "This allows you to verify their origin. But if you install packages or Firefox extensions found on the Internet without checking them, you run the risk of all the risks mentioned for malware." [page 32]

By way of conclusion: *free or not, there is no single piece of software that can guarantee the privacy of our data*; there are only practices associated with the use of certain software. Software chosen because there are elements that allow us to place a certain level of trust in it.

4.2 An account's password does not protect its data

All recent operating systems (Windows, macOS, GNU/Linux, *etc.*) offer the possibility of having different user accounts on the same computer. But it's important to remember that the passwords that sometimes protect these accounts are no guarantee of data confidentiality.

Admittedly, it's handy to have your own space, with your own settings (bookmarks, wallpaper, *etc.*), but a person who wants to access all the data on the computer would have no trouble doing so: by simply reconnecting to the computer, you'll be able to access all the data on your computer. the hard disk on another computer, or by booting it on another operating system, page 22 she would have access to all the data written on that hard disk.

So, while using separate accounts and passwords may have some advantages (such as the ability to lock the screen when you're away for a few minutes), it's important to bear in mind that it doesn't really protect your data.

3. Julien Lausson, 2013, *The NSA is suspected of tampering with a cryptographic standard* [<http://www.numerama.com/politique/26979-la-nsa-est-suspectee-d-avoir-altere-un-standard-cryptog-raphique.html>].

4.3 About "deleting" files

[page
24
-----]

We've already mentioned that the content of a file that has become inaccessible or invisible has not disappeared into thin air. Now we'll explain why.

4.3.1 Deleting a file does not delete its contents...

... and it can be very easy to find.

Indeed, when you "delete" a file by placing it in the *Recycle Bin* and then emptying it, you're simply telling the operating system that the contents of this file are no longer of interest to you. It then deletes its entry in the index of existing files. It can then reuse the space taken up by this data to write something else.

But it may be weeks, months or years before this space is *actually* used for new files, and the old data actually disappears. In the meantime, if you look directly at what's written on your hard disk, you'll find the contents of the "deleted" files. It's a fairly straightforward operation, automated by a number of software programs for "recovering" or "restoring" data. ⁴.

4.3.2 The beginning of a solution: rewrite over the data several times

Once new data has been rewritten to hard disk space, it becomes difficult to find what was there before. But that doesn't mean it's impossible: when the computer rewrites 1 over 0, it's more like 0.95, and when it rewrites 1 over 1, it's more like 1.05 ⁵... in much the same way as you can read on a notepad what has been written on a torn-out page, by the depressions created on the blank page below.

On the other hand, it becomes very difficult, if not impossible, to recover them when you rewrite over them several times with random data. The best way to make the contents of these "deleted" files inaccessible is therefore to use software that makes sure to rewrite over them several times. This is known as "wipe" data.

4.3.3 Some limits to rewriting possibilities

Even if it is possible to rewrite several times to a given location on a hard disk to render the data it contained inaccessible, this does not guarantee its complete disappearance from the disk.

Modern" discs

Today's disks reorganize their contents "intelligently": part of the disk is reserved to replace places that would become defective. These replacement operations are difficult to detect, so you can never be sure that the place you're rewriting to is actually where the "deleted" file was originally written.

In the case of USB sticks and SSD (*Solid State Drive*) disks, it's even safe to say that, in most cases, you're rewriting to a different location. The *flash* memory used by USB sticks and SSD disks stops working properly after a certain number of times.

4. This is the case with [PhotoRec \[https://www.cgsecurity.org/wiki/PhotoRec_FR\]](https://www.cgsecurity.org/wiki/PhotoRec_FR), for example.

5. Peter Gutmann, 1996, *Secure Deletion of Data from Magnetic and Solid-State Memory* [http://www.cs.auckland.ac.nz/~pgut001/pubs/secure_del.html].

writing ⁶These contain chips that automatically reorganize their contents, distributing the information to as many different places as possible.

By taking these mechanisms into account, it becomes difficult to guarantee that the data you wish to destroy has actually disappeared.

Nevertheless, opening up a hard disk to examine its innards takes time and considerable material and human resources. Not everyone will be able to make this investment, not all the time.

For *flash* memory chips on a USB stick or SSD drive, although not immediate either, the operation is much simpler: it suffices a soldering iron, and a device for reading memory chips directly. The latter can be purchased for around \$1,500. ⁷.

File systems

Even if the contents of a file have been perfectly deleted, traces may remain elsewhere, which may be due to the file system. page 27

In fact, today's file systems keep track of successive file modifications in a "log". These file systems are therefore said to be "logged". page 24

Logging was introduced to improve the robustness of file systems. After a sudden shutdown of the computer, it allows the system to simply resume the last operations to be performed, rather than having to scour the entire disk to correct inconsistencies. But it can also leave a trail of files you'd like to see disappear.



PRECISION

Windows uses the NTFS and ReFS file systems, which are journaled. Under GNU/Linux, ext4 is the most commonly used file system. By default, it only logs file names and other metadata, but not their contents.

Some file systems have other features that leave their mark:

- *snapshots* possible with modern file systems (NTFS, ReFS, Btrfs, *etc.*);
- caching in temporary folders with network file systems (such as NFS) ;
- *etc.*

What we don't know

As far as CD-RW or DVD±RW (rewritable) are concerned, it seems that no serious study has been carried out into the efficacy of rewriting to render data irretrievable. Current recommendations are therefore to methodically destroy media of this type that may have contained data to be erased. ⁸.

6. Wikipedia, 2020, *SSD* [<https://fr.wikipedia.org/wiki/SSD>].

7. The *PC-3000 Flash* [<https://www.ancelab.eu.com/pc3000flash.php>] is sold as a professional tool for recovering data from damaged flash devices.

8. NIST, 2014, *Guidelines for Media Sanitization* [<https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-88r1.pdf>].

4.3.4 Lots of other times when we "erase"

It's important to note that you don't just delete files by putting them in the *Recycle Bin*. For example, when you use the "Clear my tracks" option in the Firefox browser, it does no more than delete the files. While the data is no longer accessible to Firefox, it can still be accessed directly from the hard disk.

Finally, it's worth emphasizing here that *reformatting* a hard disk doesn't actually erase the content that used to be on it. Like deleting files, it only makes the space where the content used to be available, but the data remains physically present on the disk until it is overwritten. In the same way, destroying a library's catalog doesn't make the books on the shelves disappear.

This means that files can still be recovered after reformatting, just as easily as if they had simply been "deleted".⁹

4.3.5 And to leave no trace?

Unfortunately, there's no simple way of radically solving the problem. The least difficult solution at the moment is to use the computer after booting it with a *live* system configured to use only RAM, such as Tails. In this case, it's possible to write nothing to the hard disk, nor to virtual memory (*swap*), and keep information only in RAM (so only as long as the computer remains switched on).

page 113

page
25

4.4 Portable software: a false solution

"Portable software" is software that is not installed on a specific operating system, but can be started up from a USB stick or external hard drive - and therefore carried around with you, so you can use it on any computer.

However, unlike *live* systems, these programs use the operating system installed on the computer where they are to be used (in most cases, they are designed for Windows).

The idea behind them is to ensure that you always have the software you need at hand, customized to your needs. But "carrying your office around with you" isn't necessarily the best way of preserving the confidentiality of your data.

Let's face it: these programs don't protect the people who use them any more than "non-portable" software does.

4.4.1 Main problems

These "turnkey" solutions therefore pose some rather unfortunate problems.

Traces will remain on the hard disk

If the software has been rendered "portable" correctly, it shouldn't leave any traces on the hard disk of the computer you're using it on. But in reality, the software never has absolute control. It is largely dependent on the operating system.

on which it is employed, which may need to write virtual memory (*swap*) to the hard disk, or record various traces of what it does in its logs and other "recent documents". All this will then remain on the hard disk.

page

28

page

29

⁹ **PhotoRec** [https://www.cgsecurity.org/wiki/PhotoRec_FR] also offers this kind of functionality.

There's no reason to trust an unknown system

We've already seen that many systems don't do what they're supposed to. However, since the portable software will use the system installed on the computer on which you launch it, you will suffer from all the bugs and other malware that might be there.

We don't know who compiled them or how.

Modifications made to software to make it portable are rarely checked, and usually not by the software authors themselves. As a result, such software is even more likely than non-portable versions to contain security flaws, whether introduced by mistake or deliberately.

Later on, we'll look at the criteria to take into account when choosing the software to install or download.

One way to protect data: cryptography

Cryptography is the branch of mathematics that deals specifically with protecting messages. Until 1999, the use of cryptographic techniques was forbidden to the general public. It has now become legal, among other things to enable Internet merchants to get paid without having their customers' credit card numbers stolen.

Cryptanalysis is the field of "breaking" cryptographic techniques, for example to recover a message that had been protected.¹

There are three aspects to message protection:

- **confidentiality**: prevent prying eyes ;
- **authenticity**: ensuring the source of the message ;
- **integrity**: to ensure that the message has not been modified.

You can want all three things at the same time, but you can also want only one or the other. A person writing a *confidential* message may wish to deny authorship (and therefore that it cannot be *authenticated*). We can also imagine wanting to certify the provenance (*authenticity*) and *integrity* of an official communiqué that will be distributed publicly (and therefore far from being *confidential*).

In what follows, we'll be talking about *messages*, but cryptographic techniques can be applied to any numbers, and therefore to any data, once digitized.

Cryptography does not seek to hide messages, but to protect them. To hide messages, it is necessary to use steganographic techniques.

Physics (such as those used by the printers mentioned earlier, or even to repudiable encryption), which we won't go into here.

page 36

page 52

]

5.1 Protect data from prying eyes

Encryption is the most serious way of ensuring that data can only be understood by those "in the know". Children who use codes to exchange words, or soldiers who communicate their orders, have understood this very well!

Encrypting a file or storage medium makes it unreadable to anyone who doesn't have the access code (often a *passphrase*). It will be

1. For a good overview of the various methods - known as "attacks" - commonly used in cryptanalysis, please refer to the [Wikipedia](https://fr.wikipedia.org/wiki/Cryptanalysis) page, 2020, *Cryptanalysis* [https://fr.wikipedia.org/wiki/Cryptanalysis].

content can still be accessed, but the data will resemble a series of random numbers, making it incomprehensible and unusable.

We often use the terms "*encrypt*" and "*decrypt*" instead of "*encrypt*" and "*decrypt*", which can lead to confusion. However, we prefer to avoid using the Anglicism *crypter*, and reserve *decrypt* for the operation consisting in thwarting an encryption system (i.e. "decrypting" a message without knowing the secret encryption code).

5.1.1 How does it work?

Roughly speaking, there are only three main ideas for understanding how to encrypt messages².

The first idea: *confusion*. The relationship between the original (unencrypted) message and the encrypted message must be obscured. A very simple example is the "Caesar cipher", which consists in shifting each letter of the plaintext by three characters in the alphabet:

plain text :	AS SAUT	IN	A	HOUR
	↓↓↓↓↓	↓↓↓↓↓	↓↓↓	↓↓↓↓↓
ciphertext:	DVVDXW	GDQV	XQH	KHXUH

A + 3 letters =
D

Except that with Caesar's cipher, it's easy to analyze letter frequencies and find words.

So the second big idea is *diffusion*. This breaks up the message to make it more difficult to recognize. An example of this technique is column transposition. So, for a three-point spread, we divide the text into three lines and then transcribe it column by column:

1	2	3	4	5	6		1	2	3	4	5	6
A	S	S	A	U	T		A	S	S	A	U	T
						→	A	S	S	A	U	T
D	A	N	S	U	N	distribution	A	S	S	A	U	T
						in 3 points	E	H	E	U	R	I
							A	S	S	A	U	T
							H	U	A	U	R	T
							E	H	E	U	R	T

What we call *encryption algorithms* are the different techniques used to transform the original text. As for the *encryption key*, in the case of Caesar's cipher, for example, it's the number of offset characters (3, in this case) or, in the broadcast technique, the number of rows in the columns. The value of this key is variable: we could just as easily have decided to use columns of 2 rows, or an offset of 6 characters.

Which brings us to the third big idea: *the secret lies only in the key*. After a few millennia, we've realized that it's a bad idea to assume that nobody will be able to understand the encryption algorithm: sooner or later, someone will figure it out. It's much easier to keep a simple encryption key or passphrase secret than an entire algorithm.

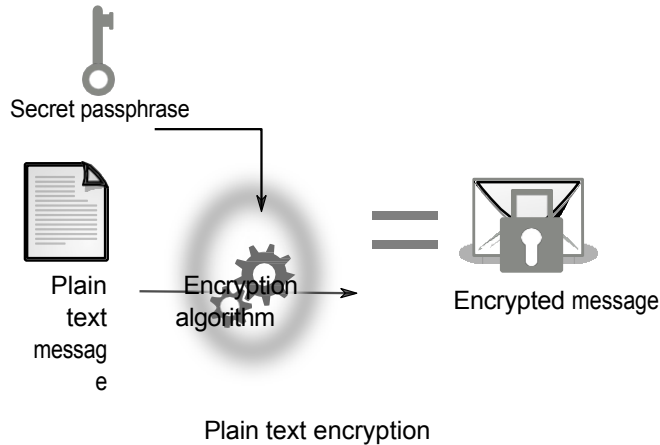
Nowadays, the algorithm can be found in great detail on Wikipedia, enabling anyone to check that it has no particular weak point, i.e. that the only way to decrypt an encrypted message is to have the *key that* was used with it.

2. The following is a very partial adaptation of [Jeff Moser's comic strip on the AES algorithm](https://www.moserware.com/2009/09/stick-figure-guide-to-advanced.html) [https://www.moserware.com/2009/09/stick-figure-guide-to-advanced.html].

5.1.2 Would you like a drawing?

In concrete terms, to ensure the *confidentiality* of our data, we use two operations: encryption, to protect the data, and decryption, to be able to read it. These operations are performed by software, such as GnuPG.

First step: encryption



For a practical example, let's take the following message³ :

The spaghetti is in the cupboard.

After encrypting this message using the GnuPG software with the AES- 256 algorithm and, as a passphrase, "*this is a secret*", we obtain, for example⁴ :

```
----- BEGIN PGP MESSAGE -----
jA0ECQMCRM0lmTSIONRg0lkBWGQI76cQ0ocEvdBhX6BM2AU6aYSPYmSqj8ihFX
u      wV1GVraWuwEt4XnLc3F+0xT3EaXINMHdH9oydA92WDkaqPEnjsWQs/
oSCeZ3WXoB 9 mf9y6jzqozEHw==
=T6eN
----- END PGP MESSAGE -----
```

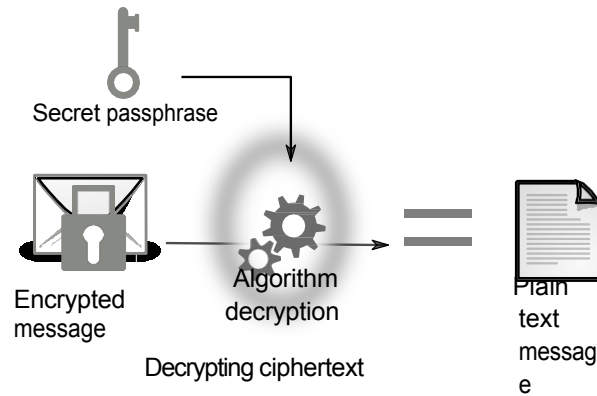
This is what a text looks like after encryption: its content has become completely incomprehensible. The "plaintext" data, readable by everyone, has been transformed into another format, unreadable by anyone who doesn't have the passphrase.

Second stage: decryption

To decrypt, we'll suffira GnuPG again, this time with our ciphertext. GnuPG will ask us for the passphrase used to encrypt our message, and if this is correct, we'll finally get the information we were missing to prepare lunch.

3. This message is of the utmost strategic importance to *peop le* you would invite into your home. It is therefore crucial to encrypt it. Joking aside, if we only encrypt "sensitive" messages, then all the encrypted messages we send are suspect; hence the importance of encrypting even innocuous messages.

4. Even if the same message is encrypted with the same passphrase, the result is different each time the operation is repeated: to prevent anyone from comparing encrypted messages (without knowing the passphrase) to find out whether or not they correspond to the same message in clear text, random data is introduced to make each encrypted message unique and distinct from the others.



5.1.3 For a hard disk...

If you want all the data you put on a storage medium (hard disk, USB key, *etc.*) to be encrypted, the operating system will have to perform encryption and decryption operations "on the fly".

This means that whenever data needs to be read from the hard disk, it will be deciphered on the way so that the software that needs it can access it. Conversely, every time a software application asks to write data, it will be encrypted before landing on the hard disk.

For these operations to work, the encryption key needs to be in RAM for as long as the medium needs to be used.

[page

18

Furthermore, the encryption key cannot be changed. Once the key has been used to encrypt the data on the disk, it becomes indispensable for re-reading it. To change the key, you'd have to reread and then rewrite all the data on the disk...

To avoid this tedious operation, most systems used to encrypt storage media have a trick: the encryption key is in fact a large random number, which is itself encrypted using a *passphrase*.⁵ This encrypted version of the encryption key is usually written on the storage medium at the beginning of the disk, as a "*header*" for the encrypted data.

With this system, changing the passphrase becomes simple, since it suffices to modify this *header* (which is usually done automatically by these encryption systems).

5.1.4 Summary and limitations

Cryptography is the perfect way to protect your data⁶ by encrypting all or part of your hard disk, any other storage medium (USB key, CD, *etc.*), or your communications. What's more, modern computers are sufficient powerful enough for us to make encryption routine, rather than reserving it for special circumstances or particularly sensitive information (otherwise, it immediately identifies the latter as important, whereas it's better to dissolve it in the mass).

[page 249

5. The LUKS system, used under GNU/Linux, even allows the use of several encrypted versions of the encryption key. Each of these versions can be encrypted with a different *passphrase*, enabling several people to access the same data without having to remember the same secret.

6. A Rue89 article on Snowden's revelations about the NSA's powerlessness in the face of the encryption: Marie Gutbub, 2014, *War crimes and data decryption: new revelations from Snowden* [<https://www.nouvelobs.com/rue89/rue89-monde/20141229.RUE7224/crimes-de-guerre-et-decryptage-de-donnees-nouvelles-revelations-de-snowden.html>].

A passphrase can be used to encrypt an entire hard disk, and/or to give certain people an encrypted part with their own passphrase. It is also possible to encrypt individual files, e-mails or attachments, with an even different passphrase.

However, while it is a powerful and essential tool for information security, **encryption does have its limitations**, and these must be borne in mind when using it.

As explained earlier, when accessing encrypted data, there are two things to bear in mind. Firstly, once the data has been decrypted, it remains in RAM *at the very least*. Secondly, as long as data needs to be encrypted or decrypted, RAM also contains the *encryption key*.

Anyone who has the encryption key can read *anything encrypted with it*, and can also use it to encrypt data themselves.

The following points should be borne in mind:

- The operating system and software have access to the data and encryption key just as much as we do, so it all depends on how much you trust them.
- so only install software you trust. page 32
- Anyone who gains physical access to the computer when it's switched on, has de facto access to the contents of RAM. When an encrypted disk is activated, it contains, in clear text, the data you've been working on since you turned on the computer (even if encrypted on disk). But above all, as mentioned above, it contains the encryption key, which can be copied. So it's best to get into the habit of switching off computers and disabling (disassembling, ejecting) encrypted disks when not in use. page 27
- In some cases, it may be necessary to provide hardware solutions to cut power quickly and easily. ⁷ encrypted disks become inaccessible again without the passphrase - unless you *could* page 27
- It is also possible that a keylogger has been installed on the computer, and that this records the passphrase. page 27

Also, the mathematics used in cryptographic algorithms are sometimes flawed. And much more often than not, the software that applies them has weaknesses or errors. Some of these problems can, from one day to the next, make it possible to decrypt in a few clicks data encrypted with what was thought to be the best protection available. ⁸...

There is also a certain **"legal" limit** to possible attacks. In France, anyone who encrypts his or her data is expected to give the access code to the judicial authorities when they request it, as explained in article 434-15-2 of the Penal Code ⁹ :



It is punishable by three years' imprisonment and a fine of €270,000 for any person with knowledge of the secret decryption agreement for a cryptographic means likely to have been used to prepare, facilitate or commit a crime or misdemeanor to refuse to hand over the said agreement to the judicial authorities or to implement it, in response to requests from these authorities issued under Titles II and III of Book I^{er} of the Code of Criminal Procedure.

7. For this reason, it may be advisable not to leave the battery connected in a laptop computer when not in use. It is then sufficient to remove the mains cable to switch it off.

8. Zythom's blog, 2015, *The encrypted hard drive* [<https://zythom.fr/2015/03/le-disque-dur-chiffre/>].

9. The legal term is "cryptology". A search on this word on Légifrance [<https://www.legifrance.gouv.fr/>] will give an exhaustive list of legal texts in this field.

If the refusal is made when the surrender or implementation of the agreement would have made it possible to avoid the commission of a crime or offence or to limit its effects, the penalty is increased to five years' imprisonment and a fine of €450,000.¹⁰

Note the words "*susceptible*" and "*sur les réquisitions*", meaning that the law is vague enough to require anyone holding encrypted data to spill the beans. We could be asked for the passphrase of a medium that isn't ours... and that we wouldn't have. However, the police, even in the person of an OPJ¹¹ must have obtained prior authorization from a magistrate.¹² On the other side of the Channel, similar customs legislation means that Muhammad Rabbani, director of the CAGE organization, could face prison for refusing to hand over his passwords at the border.¹³

Contrary to many people's expectations^{14 15} the court rejected the "right not to incriminate oneself" argument in defense of not giving out its decryption agreement. It argued that "this data, already fixed on a support, exists independently of the suspect's will".¹⁶ In addition, the Criminal Division of the French Supreme Court (Cour de cassation) has ruled that "the unlocking code of a cell phone can constitute a decryption key, if the phone is equipped with a cryptographic device".¹⁷

Some techniques also combine encryption and steganography to make the presence of encrypted data undetectable: this is known as "repudiable encryption", "plausible deniability" or "deniable encryption".¹⁸ *deniable encryption*. Software such as VeraCrypt¹⁹ offer this feature, which in theory would make it possible to deny the existence of encrypted data to a judicial authority, and thus avoid being forced to reveal the passphrase under the terms of article 434-15-2. However, there is as yet no case law on this subject, and the use of repudiable encryption does not rule out the possibility that the courts may be able to demonstrate the existence of encrypted data by other means. Caution is therefore called for.

Since 2014²⁰ the cops also have the right to requisition anyone they want.

". to be informed of the measures applied to protect the data" and "to provide them with information enabling them to access the data".²¹

10. Légifrance, 2016, *Penal Code*, article 434-15-2 [https://www.legifrance.gouv.fr/codes/article_lc/LEGIARTI000032654251/2016-06-05].

11. Officière de police judiciaire.

12. Les Numériques, 2020, *Refusing to unlock your smartphone to the police, an offence in certain cases* [<https://www.lesnumeriques.com/telephone-portable/refuser-le-deverrouillage-d-e-son-smartphone-a-la-police-une-infraction-dans-certains-cas-n155755.html>].

13. Birkbeck Law Review, 2018, *In Conversation with Muhammad Rabbani, CAGE* [<https://web.archive.org/web/20211102115950/http://www.bbklr.org/blog/in-conversation-with-muhammad-r-abbani-cage>].

14. Maître Éolas, 2014, *Allô oui j'écoute* [<https://www.maitre-eolas.fr/post/2014/03/08/All%C3%BA-oui-j-%C3%A9coute#c173067>].

15. La Quadrature du Net, 2018, *Le Conseil constitutionnel restreint le droit au chiffrement* [<https://www.laquadrature.net/2018/04/04/le-conseil-constitutionnel-restreint-le-droit-au-chiffrement/>].

16. Conseil constitutionnel, 2018, *Decision no. 2018-696 QPC of March 30, 2018* [<https://www.conseil-constitutionnel.fr/decision/2018/2018696QPC.htm>].

17. Légifrance, 2021, *Court of Cassation, Criminal Division, March 3, 2021, 19-86.757, Unpublished* [<https://www.legifrance.gouv.fr/juri/id/JURITEXT000043252997>].

18. Wikipedia, 2020, *Plausible deniability (cryptology)* [[https://fr.wikipedia.org/wiki/D%C3%A9ni_plausible_\(cryptologie\)](https://fr.wikipedia.org/wiki/D%C3%A9ni_plausible_(cryptologie))].

19. Official VeraCrypt website [<https://www.veracrypt.fr/>].

20. Légifrance, 2014, *Loi n° 2014-1353 du 13 novembre 2014 renforçant les dispositions relatives à la lutte contre le terrorisme* [<https://www.legifrance.gouv.fr/jorf/id/JORFTEXT000029754374>].

21. Légifrance, *Code de procédure pénale*, article 57-1 [https://www.legifrance.gouv.fr/codes/article_lc/LEGIARTI000032655328].

Nevertheless, some people refuse to give their encryption agreement and claim it in the name of the right to remain silent and not to incriminate themselves.²² just as others refuse to give their DNA, which is also - to a lesser extent - criminally reprehensible.

5.2 Ensure data integrity

We've seen a few ways of ensuring the *confidentiality* of our data. However, it may also be important to be able to ensure their *integrity*, i.e. to check that they have not been altered in the process (by accident or by maliciously, in order to introduce bugs, for example). We may also want to ensure the *authenticity* of our data, page 32.

5.2.1 Chopper power

Most techniques for ensuring integrity or authenticity are based on mathematical tools that cryptography has dubbed "hash functions".

These work like *mincers*, able to reduce anything to tiny pieces. And if our chopper works suffis well enough to be used in cryptography, we know that :

- with the small pieces, it's impossible to reconstitute the original object without trying every object on Earth;
- the same object, once chopped, will always produce the same small pieces;
- the probability that two different objects will produce exactly the same small pieces is astronomically low.

When these properties are met, we then suffit to compare the small pieces from two objects to see if they were identical.

The little bits that come out of our mincer are more commonly known as a *checksum* or *fingerprint*. It's usually written in a form that looks like :

```
f9f5a68a721e3d10baca4d9751bb27f0ac35c7ba
```

Our mincer works with data of any size and shape: we can just as easily reduce to small pieces - i.e., calculate their fingerprints - an image, a CD, a piece of software, *and so on*. So, for example, rather than directly comparing the contents of two DVDs byte by byte, which is likely to be long and tedious, we can simply compare their fingerprints to determine whether they are identical.

This doesn't mean our chopper is magic. It's easy to imagine that when you reduce anything to the same size cubes, you can end up with the same cubes from two different objects. This is called a *collision*. Fortunately, the probability of such mathematical collisions occurring by chance is astronomically low, except when there are algorithms to cause them... which has already happened with several hash functions after a few years of research, such as SHA-1²³ function, for example. In this case, the third property of the hash function is no longer respected, and its use should be discontinued.

22. Le Parisien, 2018, *Un gardé à vue peut garder le silence mais doit donner les codes de son smartphone* [<https://www.leparisien.fr/faits-divers/un-garde-a-vue-peut-garder-le-silence-mais-do-it-donner-les-codes-de-son-smartphone-16-04-2018-7667613.php>].

23. Marc Stevens *et al*, 2017, *Announcing the first SHA-1 collision*, Google Security Blog [<http://security.googleblog.com/2017/02/announcing-first-sha1-collision.html>].

5.2.2 Checking software integrity

Let's take an example: Ana has written a program and distributes it on CDs, which can be found in GNU/Linux user clubs. Bea wants to use Ana's program, but realizes that it would have been very easy for a malicious administration to replace one of Ana's CDs with malware.

She can't pick up a CD directly from Ana, who lives in another town. On the other hand, she met Ana some time ago, and knows her voice. So she phones her, and Ana gives her the *checksum* of the CD's contents, which she has calculated with a secure hash function:

Ana's CD	→	
	hash function	
		94d93910609f65475a189d178ca6a45f 22b50c95416affb1d8feb125dc3069d0

Bea can then compare it with the one she generates from the CD she has purchased, using the same hash function:

Bea's CD	→	
	hash function	
		94d93910609f65475a189d178ca6a45f 22b50c95416affb1d8feb125dc3069d0

As the checksums are identical, Bea is happy, because she's sure she's using the same CD as the one supplied by Ana.

Calculating these checksums doesn't take much longer than playing the whole CD... a few minutes at most.

Now let's put ourselves in the shoes of Carole, who has been paid to take control of Bea's computer without her knowledge. To do this, she wants to create a CD that looks like Ana's, but contains malware.

Carole begins by obtaining Ana's original CD. She then modifies the CD to include the malware. This first version closely resembles the original. It might fool some people who aren't careful, but she knows that Bea will check the CD's checksum before installing the program it contains.

As Ana uses a hash function that has no known flaws, all that's left for Carole to do is to try out a huge number of variations on the data on her CD, in the hope of obtaining a *collision*, i.e. the same checksum as Ana's.

Unfortunately for her, and fortunately for Bea, even with a great many powerful computers and even if she spent an extremely long time on them, Carole's chances of success would remain astronomically low.²⁴

It therefore suffices to obtain a *fingerprint*, or *checksum*, through trusted intermediaries to verify data integrity. The challenge is then to obtain these fingerprints by trusted means, i.e. to be able to verify their *authenticity*...

24. To put things into perspective, with a hash function currently considered secure (SHA-256, for example), even if Ana had a billion billion computers, each capable of calculating ten billion checksums per second, and had them calculated for a period of time equivalent to the current age of the universe (fifteen billion years), it would still be *a long way from* having a reasonable chance of finding a collision!

However, this does not take into account possible future advances in cryptanalysis that could discover weaknesses in the hash function used and propose more efficient algorithms to enable Carole to carry out her attack in a reasonable time.

5.2.3 Check password

Another example of the use of hash functions concerns verifying *the authenticity* of an access request.

If access to a computer is password-protected, for example when logging on to GNU/Linux²⁵ the computer must be able to check that the password entered is the correct one. However, passwords are not stored in clear text on the computer, otherwise it would be too easy to obtain them.

But how does the computer ensure that the password typed on the keyboard is correct?

When you choose a password for your computer, the system uses a hash function to record an imprint of the password. To verify access, it "chops" the password entered in the same way. And if the fingerprints are the same, it considers that the password was the right one.

It is therefore possible to check that the password matches, without keeping the password itself!

5.3 Symmetrical, asymmetrical?

The encryption techniques mentioned so far rely on a single secret key, which is used for both encryption and decryption. This is known as *symmetrical* cryptography.

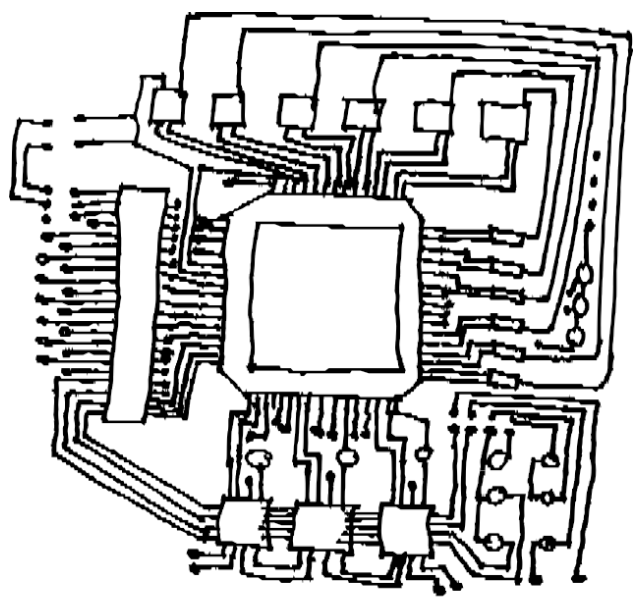
This is in contrast to *asymmetric* cryptography, which, in the context of chif-frement, does not use the same key to encrypt and decrypt a message. Also known as "public key cryptography", it is mainly used for online communication, and will be discussed in detail in the second volume.

page 249

One of the most interesting properties of asymmetric cryptography, which can be briefly mentioned here, is its ability to produce *digital signatures*. Like its paper counterpart, a digital signature can be used to affix a recognition mark to data.

These digital signatures, which use asymmetrical cryptography, are the simplest way of verifying the origin of software. We'll be using them later...

25. Remember that passwords are not used to protect data [page 41]!



PART TWO

Choosing the right answers

Introduction

To be honest, this first chapter and the understanding of how computers work is enough to make anyone panic... But burying your head in the sand is not an appropriate solution: we've already dismissed the idea that we have nothing to hide. Denial doesn't make you any less exposed to risks or threats.

After reading this, you'll certainly think back to the action movies where the heroine hides her computer equipment in a library safe that opens when you pull out a few books in a special combination... and discouragement can get the better of you.

Fortunately, there's another solution: read the rest of the guide! This new section presents typical situations, called *use cases*, and then suggests practices and tools appropriate to each situation.

Trust and risk reduction

The notions of *trust* and *risk reduction* are useful when choosing how to use digital tools. In this chapter, we'll look at sex, drugs and rock'n'roll, contexts in which these notions have already been considered. Like digital technology, these practices can be fun, but they often involve risks.

6.1 Risk reduction

Working on a building site or in an office ;
share piercing jewelry, sex toys, or toothbrushes; connect to
the Internet or use digital tools;
getting into a car or riding a bike...

All these practices entail risks... and they can be reduced!

As Act Up puts it: "information = power". Indeed, knowledge and understanding enable us to take more pleasure while taking fewer risks, in the knowledge that zero risk does not exist. The notion of risk is relative, comprising so many different aspects that we need to discuss them in order to understand and define them.¹

Disseminating information about the risks involved is all the more important given that when a virus is present, whether in the body or on a computer, there are not necessarily any visible signs. That's why, from a health point of view, it's advisable not to wait for symptoms before taking action, and to undergo regular screening. Or, on the digital front, regular updates help reduce the risk of infection by correcting security flaws.

Many people spend their time facilitating risk reduction: by distributing information leaflets, but also condoms, *straw rollers* or earplugs. In the same vein, people are developing software, writing documentation and running workshops on self-defense and digital intimacy.

To make using encryption as easy as putting on a hard hat; to know as much about the clitoris as about RAM;
to make Tor as easy to use as earplugs;
because when all is said and done, masturbating and searching for wallpaper images for hours on end makes life more pleasant and shouldn't be risky!

1. This is why, in the 90s, in the field of drug addiction, we stopped talking about *prevention* and started talking about *harm reduction* [<http://www.keep-smiling.com/?p=259>]. In the field of sexuality, people stopped using the term "*safe sex*" and started using the term "*safer sex*".

6.2 A story of trust

Trusting digital equipment, software and networks is like trusting a mechanic: you can either believe them, or not. In the same way, a software development team can be trusted to a greater or lesser degree. ².

This super application completely encrypts all your data! This helmet protects your head in the event of an accident!

Don't worry, I didn't take any chances!

Affirmations of this kind are not sufficient for building a relationship of trust. It's best to ask questions, to look beyond the obvious.

If an application claims to encrypt data completely, we might wonder what the encryption system is, and whether this method is approved by the communities we trust. We might also wonder whether this is deceptive marketing.

In the same way, when we share sexuality, we can discuss what we consider to be risk-taking, and what our limits, practices or relationship to risk reduction and screening are.

Asking yourself the following questions helps you assess the confidence you might have in a tool:

- Why was this tool developed?
- Is he free?
- What is its business model?
- Who is he accountable to?

But it's not all about risk and trust. It's only a risk when there's no intention to harm people, groups, ideas, *etc.*; otherwise, it's a threat.

The risk is to damage your hands pulling out brambles without gloves, the threat is the boss who puts his unionized employees on the back burner; the risk of drunk driving is to have an accident, the threat is to have your driving license revoked; the risk, when you haven't made a backup, is to lose all your data if a hard disk crashes, and the threat is that the police will search the place of militant activity where it is stored; the risk is a software bug that crashes the computer, the threat is Cambridge Analytica, which uses Facebook account data to influence election results.

Of course, it's more complicated: there are risks of threats, and threats can generate risks!

The question of threat is not just an individual one, since failure to take certain precautions can expose the people with whom we communicate.

2. This question of trust can be asked of just about any specialist: journalists, doctors, farmers, carpenters, engineers, cops, shopkeepers, tattoo artists, bricklayers, pilots (airplane, train, bus, *etc.*), *town planners*, hairdressers, architects, pharmacists, teachers, artists, lifeguards, nurses, bakers, nurserymen and women who work in the public sector.), urban planners, hairdressers, architects, pharmacists, teachers, artists, lifeguards, nurses, bakers, politicians, postmen and women, shrinks, receptionists (at the CAF, MSA, Pôle emploi, *etc.*), accountants, sales reps, saleswomen, *etc.*), accountants, sales reps, veterinary surgeons, human resources managers, scientists, craftswomen, cooks, security guards, bankers, workers, sportswomen, coaches, electronics technicians, educators, social workers, *and so on.*

Risk assessment

When you ask yourself what measures you need to put in place to protect data or digital communications, you soon realize that you're going in a bit of a blind alley.

In fact, the solutions we could put in place also have their drawbacks: sometimes they're a real pain to deploy, maintain or use; sometimes we have a choice between various techniques, none of which fully meets the "specifications" we've set ourselves; sometimes they're far too new for us to be sure they'll really work; *and so on*.

We should therefore start by asking ourselves a few simple questions, in order to establish a *threat model*¹ that is, the identification and prioritization of potential threats.

7.1 What do we want to protect?

What we want to protect generally falls into the broad category of *information*: for example, the content of e-mail messages, data files (photos, leaflets, address books) or the very existence of correspondence between such and such a person.

The word "protect" covers different needs:

- **confidentiality**: hiding information from unwanted eyes ;
- **integrity**: keeping information in good condition, and preventing it from being altered without our knowledge;
- **accessibility**: ensuring that information remains accessible to those who need it.

For each set of information to be protected, you need to define the requirements of confidentiality, integrity and accessibility. Given that these needs are generally in conflict, it will be necessary to prioritize and find compromises, and go easy on the (hungry) goat and the (very appetizing) cabbage...

7.2 Who do we want to protect ourselves against?

The question soon arises as to the capabilities of the people who might be after what we want to protect: intrusive parents, classmates likely to harass, thieves wanting to recover bank details, an abusive ex-spouse looking for means of control or blackmail, hierarchies that are too curious, the police in charge of quelling a social movement, civil servants who want to protect their employees, etc. The question arises as to the capabilities of the people who might be after what we want to protect.

1. See [Electronic Frontier Foundation, 2019, Your security plan \[https://ssd.eff.org/fr/module/your-security-plan\]](https://ssd.eff.org/fr/module/your-security-plan).

control migrant workers ²GAFAMs that track and sell personal data, intelligence services mandated to massively file a community or a political trend, *etc.*

But it's not easy to know what the most qualified of them can actually do, or what resources and budgets they have. By following the news, and through various other means, we can see that it varies a great deal depending on who we're dealing with. Between parents, local gendarmes and the US *National Security Agency* (NSA), there's a wide gulf in the range of actions, means and techniques employed.

The question of the opponents' means is quite broad.

There are **judicial means**: for example, the possibility that a letter rogatory authorizes the police to seize computer equipment, or the fact that you can be required to give your encryption key.

At the same time, some organizations, such as SDAT ³ or the DGSE ⁴. Nothing is certain about their capabilities: how advanced are they in breaking cryptography? Are they aware of any undisclosed flaws in their methods, which would enable them to read the data? On these subjects, there's obviously no way of knowing for sure what these entities can do.

Financial resources must also be taken into account. Some surveillance technologies are expensive, and not all intelligence services can afford them, nor will they be available for every investigation. Considering that the DGSE's annual budget was €880 million in 2021 and that the NSA's was estimated at \$10.8 billion (!) in 2013, they're not playing in the same league.

There is also the question of **political means**: for example, to what extent can the French state collaborate with the NSA?

On the other hand, completely securing a computer is an impossible task. So it's more a question of putting obstacles in the way of those who might be after our information. The greater these people's means, the more numerous and solid the sticks need to be.

Risk assessment means asking what data you want to protect, and from whom. From there, we can try to find out what means are available to those who might be interested, and define a *security policy* accordingly.

2. See Ritimo's article: [10 menaces contre les migrant·es et les réfugié·es](https://www.ritimo.org/10-menaces-contre-les-migrant-es-et-les-refugie-es) [https://www.ritimo.org/10-menaces-contre-les-migrant-es-et-les-refugie-es].

3. French police department dedicated to fighting terrorism, see [Wikipedia, 2021, Sub direction anti-terroriste](https://fr.wikipedia.org/wiki/Sous-direction_anti-terroriste) [https://fr.wikipedia.org/wiki/Sous-direction_anti-terroriste].

4. French intelligence service, see [Wikipedia, 2017, Direction générale de la Sécurité external](https://fr.wikipedia.org/wiki/Direction_g%C3%A9n%C3%A9rale_de_la_S%C3%A9curit%C3%A9_exterrieure) [https://fr.wikipedia.org/wiki/Direction_g%C3%A9n%C3%A9rale_de_la_S%C3%A9curit%C3%A9_exterrieure].

Defining a security policy

A chain is only as strong as its weakest link. There's no point in installing three huge bolts on an armored door next to a dilapidated window.

Similarly, encrypting a USB flash drive is of little use if the data stored on it is used on a computer, which will keep *unencrypted* traces of it on its hard disk.

These examples teach us something: such targeted "solutions" are of no use unless they form part of a coherently articulated set of practices. What's more, the information we want to protect is most often related to practices outside the scope of digital tools. This is why

We therefore need to assess risks globally and devise appropriate responses.

In a global, but *situated* way: a given situation corresponds to a singular set of issues, risks, know-how... and therefore possibilities for action. There is no one-size-fits-all solution that will solve every problem with a wave of a magic wand. The only practicable path is to learn sufficient to be able to imagine and implement a safety policy appropriate to one's own situation.

8.1 A matter of compromise

Digital data and communications can always be *better* protected. The possibilities for attack and surveillance are limitless, as are the devices available to protect against them. However, each additional protection we want to put in place requires an effort in terms of learning and time: not only an initial effort to get started, to install the protection, but also, very often, additional complexity of use, time spent typing passphrases, carrying out tedious and repetitive procedures, focusing our attention on the technique rather than on the use we would like to make of digital tools.

In every situation, therefore, it's a question of finding a suitable **compromise** between ease of use and the desired level of protection.

Sometimes, this compromise simply **doesn't exist**. The effort required to protect against a plausible risk would be too painful, and it's better to run that risk, or, quite simply, not to use digital tools to store certain data or talk about certain things. Other means do exist, and their efficacy has long been proven: some manuscripts have survived for centuries, buried in jars stored in caves...

8.2 What to do?

The aim is to answer the following question: what set of practices and tools should protect me sufficiently against the risks assessed above?

To do this, we can start with our current practices and ask ourselves the following questions:

1. Faced with such a security policy, what angles of attack would my adversaries use?
2. What means should my opponents use?
3. Are these resources available to my opponents?

If your answer to the third question is "yes", take the time to find out about the solutions that could protect you against these attacks, and then imagine the changes in practice that these solutions and the resulting security policy would entail. If this seems practicable, put yourself back in your opponents' shoes, and ask yourself the above questions again.

Repeat this process of reflection, research and imagination until you find a workable path, a tenable compromise.

If you're not sure, you can always ask someone more knowledgeable and trustworthy to put themselves in the opponents' shoes: they'll be delighted to see that you've done most of the thinking yourself, which will certainly encourage them to help you on the points that remain out of your reach.

8.3 A few rules

Before taking a closer look at concrete cases and the security policies that could be put in place, there are a few main principles, a few broad families of choices.

8.3.1 Complex vs. simple

When it comes to safety, a simple solution is always preferable to a complex one.

Firstly, because a complex solution offers more "attack surfaces", i.e. more places where security problems can and will occur...

Secondly, because the more complex a solution, the more knowledge is required to imagine, implement and maintain it, as well as to examine it and assess its relevance and problems. This means that, as a general rule, the more complex a solution, the less it will have undergone the sharp - and external - scrutiny needed to establish its validity.

Finally, quite simply, a complex solution that doesn't fit entirely into the mental space of the people who developed it is more likely to generate safety problems arising from complex interactions or difficult-to-detect special cases.

For example, rather than spending hours setting up devices to protect a particularly sensitive computer against network intrusions, you might as well unplug it. Sometimes you can even physically remove the network card... -----

8.3.2 Authorized list, blocked list

The usual reflex when we become aware of a threat is to try to prevent it. For example, once you've discovered that a particular piece of software leaves traces of your activities in a particular folder, you'll regularly clean that location. Until you discover that the same software is also leaving traces in another folder, and so on.

This is the principle of the blocked list ¹ a list of folders where temporary files are stored, software that sends reports, and *so on*. This list is added to as new discoveries and unpleasant surprises are made; on this basis, we try to do our best to guard against each of these threats. In other words, a blocked list works on the basis of *trust-but-in-certain-cases*.

The principle of the authorized list ² is the opposite: it's one of *mistrust-except-in-certain-cases*. *Everything is forbidden, except* what is explicitly authorized. Files are not allowed to be stored on the hard disk, except in such and such a place, at such and such a time. Software is forbidden to access the network, except for certain well-chosen software.

page 131

So much for the basics.

Any security policy based on the *blocked list* principle has one big problem: such a list is never complete, as it only takes into account problems that have already been identified. It's a never-ending, hopeless task to keep a blocked list up to date; whether we do it ourselves or delegate it to people with specialist IT knowledge, something is bound to be overlooked.

The trouble is, despite their crippling shortcomings, tools based on the *blocked list* approach are legion (as we shall see), unlike those based on the *authorized list* method, which is consequently less familiar to us.

Implementing the *authorized list* approach requires an initial effort which, while it may be significant, is quickly rewarded. Learning to use a live system that doesn't write anything to the hard disk without being asked to do so takes some time. But once you've done that, you're done with long, inefficient hard disk clean-up sessions that always have to be repeated, because they're based on the *blocked list* principle.

page 113

Another example is antivirus software, which is designed to prevent the execution of malicious programs. Since they operate on the blocked list principle, their databases have to be constantly updated, as they are systematically out of date. One answer to this problem, with the *authorized list* approach, is to prevent the execution of any program that has not been registered beforehand, or to limit the possibilities of action for each program. These techniques, known as *Mandatory Access Control*, also require lists to be maintained, but these are *authorized lists*, and the symptom of an obsolete list will be software malfunction, rather than computer piracy.

So it's much better, where possible, to rely on the widest possible authorized lists, so as to be able to do lots of cool things with computers, with a certain degree of confidence. And, when the appropriate authorized list doesn't exist, to rely on solid blocked lists of known provenance, bearing in mind the intrinsic problem with this method; blocked lists which we'll eventually help to complete, by sharing our discoveries.

8.3.3 No one is infallible

On the Internet, it is often said that "most computer problems lie between the chair and the keyboard". ³ Behind this contemptuous expression for the people who use the tools lies a reality: nobody is infallible, and human error is always a possibility.

1. Sometimes also referred to as a "black list".

The expressions "black list" and "white list" can evoke a racist dimension, both in the terms themselves and in their hierarchical structure. We have therefore chosen to replace these two terms with "blocked list" and "authorized list". Unfortunately, most programs, user manuals and other technical documentation still use these terms. That's why we find ourselves obliged to mention them.

2. Sometimes also referred to as a "white list".

3. Wiktionary, 2020, *Between the chair and the keyboard* [https://fr.wiktionary.org/wiki/entre_la_chaise_et_le_clavier].

Some practices can be devilishly efficient... until you make a mistake. Since we're bound to make one in the end, it's better to anticipate them rather than pay the piper.⁴

For example, imagine a USB key containing confidential documents. Even if you're really careful not to leave it lying around, it could end up being left on a table... and then plugged in and used by someone who has mistaken it for another, in a machine you don't trust. Encrypting the key before storing confidential documents would have significantly reduced the risks.

[page
47

In short, we're not robots. It's better to have solid material safeguards than to impose unbounded vigilance on ourselves, and it also gives us peace of mind.

8.3.4 Nothing is eternal

Once a security policy has been defined, don't forget to review it from time to time! The world of IT security evolves very quickly, and a solution considered reasonably secure today may well be easily attackable next year.

Let's not forget either, in our security policies, that it's important to monitor the life of the software on which we depend: their problems, with an impact on security; their updates, with sometimes good or bad surprises... All this takes a little time, and we might as well plan for it from the outset.

[page 175

4. In English, we use the expression "better safe than sorry". The French equivalent of "mieux vaut prévenir que guérir".

Use cases

Enough theory, let's illustrate these concepts with a few *use cases*: based on given situations, we'll suggest ways of defining an appropriate security policy. Many of the technical solutions adopted will be explained.

in the following section, to which we refer as necessary.

Page 95

Since they are all set in the offline context of this first volume, these use cases will have something of an artificial quality: they all assume that the computers in play are never connected to networks, and in particular to the Internet.

Use case: a fresh start, to stop paying the piper

(or how to clean up your computer after years of carefree use)

9.1 Context

Let's take a computer that has been used without any particular precautions for several years. This machine undoubtedly poses one or more of the following problems:

1. its disk retains unwanted traces of the past ;
2. the operating system is proprietary software (e.g. Windows), and riddled with malware.

page 27

page 31

On the other hand, troublesome files are stored on it perfectly transparently. Indeed, this computer is used for various activities (some of which are perfectly legal) such as :

- listen to music and watch movies from the Internet ;
- helping undocumented migrants prepare their files for the prefecture;
- design a pretty greeting card for his grandmother ;
- Fabricate false administrative documents that greatly simplify certain procedures (inflate pay slips when you're tired of being turned down for rental apartment after rental apartment);
- keep the family accounts up to date;
- produce "terrorist" texts, music or videos. That is, according to the European definition of terrorism ¹ threatening "*to cause massive destruction [...] to an infrastructure [...] likely [...] to produce economic losses [...]*". For example, with the aim of "*unduly coercing public authorities or an international organization to perform or refrain from performing any act whatsoever*". For example, people employed by Orange who, during a fight, threaten to disable the billing system, thereby enabling everyone to make free calls.

1. European Union, 2017, *Directive (EU) 2017/541 of the European Parliament and of the Council of 15 March 2017 on combating terrorism, replacing Council Framework Decision 2002/475/JHA and amending Council Decision 2005/671/JHA*, Article 3 [<https://eur-lex.europa.eu/legal-content/FR/TXT/?uri=CELEX:32017L0541&qid=1495634630652>].

9.2 Assessing risks

9.2.1 What do we want to protect?

[page

Let's apply the categories defined when we talked about risk assessment to the present case: -----

63

- confidentiality: to prevent an unwanted eye from falling too easily on information stored on the computer;
- integrity: to prevent information from being modified without our knowledge;
- accessibility: ensuring that this information remains accessible when needed.

Here, accessibility and confidentiality are top priorities.

9.2.2 Who do we want to protect ourselves against?

This is an important question: depending on the answer, the appropriate safety policy can vary from one thing to another.

Generous gesture, legal consequences

This computer could be seized during a search.

For example, your child generously gave a gram of *pot* to some friends, who, after being caught, informed the police of the source of the stuff... after which the public prosecutor's office prosecuted your child for drug trafficking. Hence the search.

In such cases, the computer is likely to be examined by the police, jeopardizing the objective of confidentiality. The means likely to be employed range from the gendarmes in Saint-Tropez turning on the computer and clicking around, to the forensic expert taking a much closer look at the contents of the disk. On the other hand, it is unlikely that extra-legal means will be used in this case, as they are generally reserved for special services and the military.

Burglary

This computer could be stolen during a burglary.

Unlike the police, the people who stole your computer probably don't give a damn about your secrets and won't report you. At worst, they'll blackmail you about getting your data back. It's unlikely, however, that they'll go to any great lengths to find your data on the computer disk.

9.3 Defining a security policy

Let's now ask ourselves the questions set out in the methodology, putting ourselves in the opponents' shoes.

[page 231

Everything that follows applies to offline computers. Other situations and angles of attack are conceivable if it's connected to a network, and these will be explored in the second volume of this guide.

65

9.3.1 Put yourself in the opponents' shoes

First step: when they suffit to look

1. The most practical angle of attack: plug the disk into another computer, examine its contents and find all our little secrets.

2. Resources required: another computer will enable the gendarmes in Saint-Tropez to find the bulk of our secrets; a forensic expert will also be able to find the files we thought had been deleted.
3. Attack credibility: high.

So we're going to have to adapt our practices. Encrypting the disk is page 47 the obvious response to this type of attack: installing and using an encrypted system is now relatively straightforward. -----

page 119

The steps to get there would then be :

1. Run a *live* system to perform the following operations in a relatively safe environment:
 - temporarily save the files that must survive the cleanup on an encrypted external disk or USB key;
 - eject/remove and unplug this external storage device ;
 - erase "for real" the computer's entire **internal** disk.
2. Install an open-source operating system, specifying to the installer that it is to be disk encryption, including virtual memory (*swap*).
3. Recopy previously saved data to the new system.
4. Putting in place what it takes to delete files "securely", so you can...
5. Delete the contents of the files on the temporary backup media, which may be used again in the future.

page 139

page 25

Then, from time to time, make sure that data deleted without special precautions cannot be recovered later. It's also important to ensure that the system is regularly updated, to plug any "security holes" that may arise. -----

page 32

To carry out these steps, please refer to the following recipes:

- encrypt an external disk or USB key (see page 145) ;
- use a *live* system (see page 113) ;
- save data (see page 151) ;
- erase "for real" (see page 139);
- install an encrypted system (see page 119) ;
- keep your system up to date (see page 175).

Step two: the dresser drawer was not encrypted

1. Angle of attack: the equivalent of the files we're trying to protect may be lying around in the next room, in the third drawer of the chest of drawers, on paper or on a USB stick.
2. Necessary means: search, burglary or other unannounced visit.
3. Attack credibility: high, as this is precisely the type of situation we're trying to protect against here.

Here again, we can see that a security policy needs to be considered as a whole. -----

page 65

Without a minimum of consistency in practices, there's no point in bothering to type -----

passphrases as long as a day without bread.

So it's time to sort out the papers in the chest of drawers and clean out any USB sticks, CDs or DVDs containing data that you now intend to encrypt:

- save data on an encrypted medium;
- for USB sticks and external disks: erase their contents for real;
- for CDs and DVDs: destroy and dispose of residues;
- decide what to do with previously saved data: copy it to the -----
- or archive them.

page 139

page 89

Third stage: the law as a means of coercion

page
50

1. Angle of attack: the police have the right to demand that you give them access to encrypted information, as explained in the chapter on cryptography.
2. Means required: sufficient perseverance in the investigation to apply this law.
3. Attack credibility: probable, already used on several occasions, including for narcotics cases. ².

If the police come to demand access to encrypted data, the practical question will be: does the information contained in the computer pose a greater risk than refusing to give the passphrase? After that, it depends on how you feel. Giving in in this situation doesn't undermine the whole point of encrypting your disk: at the very least, it makes it possible to know what has been revealed, when and to whom.

However, while revealing your passphrase is a personal decision, it can have wider consequences. For example, for other people whose pseudonyms are quoted in documents stored on our computer, or if in a legal proceeding all but one person gives out their passphrase. In the final analysis, whether or not to reveal your passphrase is not such an individual question, and can therefore be considered by several people. It may also be that, *on principle*, we don't want to give out our passphrase, even *though we have nothing to hide*.

page 1

Having said that, it might be a good idea to get organized so as to live through such a situation in a less delicate way: the new objective could be to have a sufficiently "clean" disk so that it won't be a disaster if we give in to the law, or if a flaw is discovered in the cryptographic system used.

As a first step, it is often possible to compromise on the accessibility of files relating to completed projects, which will often no longer be needed. This will be dealt with in the archiving use case, which can be studied afterwards.

page
page 246
89

Then there's the whole question of compartmentalization: not all our activities require the same level of security, and we don't necessarily want them all to be linked to our civil identity or to the same pseudonym. It would be possible to increase the overall level of security for all our activities, but this might be too burdensome. So, *compartmentalization* may be more appropriate. We then need to specify the respective confidentiality requirements of these various activities, and from there, sort out which, being more "sensitive" than others, should be given preferential treatment.

The next use case will look at such preferential treatment, but for now, it's best to finish reading this one!

9.3.2 Other angles of attack to consider

page
79

In addition to these situations, there are several other possible angles of attack against such a security policy.

First angle of attack: a breach in the encryption system used

As has already been explained on these pages, every security system is eventually broken. If the encryption algorithm used is broken, it will probably make the headlines, everyone will know about it, and it will be possible to react.

But if it's its implementation in the Linux kernel that's broken, it won't pass muster.

². Cour de Cassation française, 2020, *Arrêt de la chambre criminelle du 13 octobre 2020* [<https://www.courdecassation.fr/decision/5fca302e5b008f80d3ad3a35>].

page
22

not in *Libération*, and it's a safe bet that only computer security specialists will know about it.

When you're not around such people, one way to keep up to date is to subscribe to Debian's security announcements.³ E-mails received in this way are written in English, but you can find the French translation - where available - on the Debian project's "Security Information" page.⁴ page of the Debian project, where these security announcements are listed. The difficulty, then, will be to interpret them...

That said, even if the encryption system used is "broken", the adversaries still need to know about it... The gendarmes in Saint-Tropez probably won't know about it, but a forensic expert will.

Also, in the science fiction department, let's remember that it's difficult to know how far ahead the military and government agencies like the NSA are in this field.

Second angle of attack: cold boot attack

1. Angle of attack: the *cold boot attack* is described in the chapter on tracks. [page 27]
2. Means required: physical access to the computer while it is illuminated or recently switched off.
3. Attack credibility: to the best of our knowledge, this attack has never been used, at least publicly, by the authorities. Its credibility is therefore very low.

It may seem superfluous to protect yourself against this attack in the situation described here. However, it's better to adopt good habits now, rather than face unpleasant surprises in a few years' time. Which habits? Here are a few that make this attack more difficile:

- switch off the computer when not in use ;
- if using a fixed computer, ensure that the power can be cut off quickly and easily, for example by means of an easily accessible multi-socket switch;
- if using a laptop and if possible, remove the battery (it is then sufficient to unplug the power cord to switch off the machine);
- make access to your computer's RAM compartment more time-consuming and difficult, for example by gluing/welding it.

Third angle of attack: the eye and video surveillance

With the encrypted system devised in the first step, data confidentiality on [page 72] relies on the passphrase being kept secret. If it is typed in front of a video-surveillance camera, opponents with access to this camera or its A possible recorder could discover this secret, then seize the computer and gain access to the data. More simply, in a bar, a watchful eye could see the passphrase as it's being typed.

Setting up such an attack requires monitoring the people using the computer, until one of them types the passphrase in the wrong place. This can be time-consuming and costly.

To protect yourself against such an attack, you need to :

- choose a long passphrase that's hard to remember [page 103]
"on the fly" by an observer;

3. The mailing list is called [debian-security-announce](https://lists.debian.org/debian-security-announce/) [https://lists.debian.org/debian-security-announce/].

4. <https://www.debian.org/security/index.fr.html>

- check your surroundings for any unwanted eyes (human or electronic) before typing your passphrase ;
- hide your keyboard using the screen in the case of a laptop, or using a cover⁵.

Fourth angle of attack: the non-encrypted part and firmware

[page 119] As explained in the dedicated recipe, an encrypted system is not entirely encrypted: the little piece of software that asks for the passphrase to encrypt the *rest of the* data at start-up is stored in cleartext on the part of the disk known as `/boot`. A malicious person with access to the computer can easily modify this software to install a *keylogger* in just a few minutes. The keylogger will then store the passphrase as it is typed, and retrieve it later, or simply send it over the network.

[page 31

If this attack is mounted in advance, adversaries will be able to decrypt the disk when they seize the computer, during a search for example.

All in all, the means required for this attack are fairly limited: *a priori*, you don't need to be a superheroine to gain access, for a few minutes, to the room where the computer resides.

However, as far as the situation described for this use case is concerned, this attack does not appear to be the most likely.



PRECISION

One way to protect against this attack is to store the start-up programs, including that little unencrypted folder (`/boot`), on an external medium, such as a USB stick, which will be kept permanently in a more secure place than the computer. It's the *integrity* of this data, not its *confidentiality*, that needs to be protected. This requires a great deal of skill and rigor, which we won't go into here.

Such practices raise the bar for adversaries, but there remains a but: once physical access to the computer has been obtained, if `/boot` is not accessible, and therefore not modifiable, it is possible to carry out the same type of attack on the machine's firmware (BIOS or UEFI). It's slightly more difficult because the way to do it depends on the computer model used, but it's possible. We don't know of any practical way to protect against it.

Fifth angle of attack: malware

[page 31] We saw in a previous chapter that software installed on a computer without our knowledge can steal data. In this case, such software is capable of transmitting the disk encryption key to adversaries so that they can access the encrypted data as soon as they gain physical access to the computer.

Installing malware on the Debian system in question requires a higher level of skill than the attacks studied above, but also more preparation. Such an attack is therefore science fiction, at least as far as this situation is concerned. In other situations, you may need to be extremely cautious about the origin of the data and software you inject into your computer.

[page 131] To this end, the recipe for software installation gives some useful pointers on how to install new software cleanly. The second volume of this guide, devoted to networks, shows that an Internet connection adds many new angles of attack from which malware can be introduced.

5. In Laura Poitras' documentary *Citizen Four*, Edward Snowden can be seen putting a blanket over himself and his computer to type his passphrase.

Sixth angle of attack: brute force

Attacking a cryptographic system by "brute force", i.e. searching for the passphrase by testing all possible combinations one by one, is both the simplest and the slowest way. But when you can't implement any other type of attack...

For our disk encrypted in the first stage, this type of attack requires an enormous amount of time (many years) and/or money, and advanced skills... at least if the passphrase is solid.

One might think that if an organization is prepared to mobilize so many resources to gain access to our data, it would be well advised to implement one of the other attacks listed above, which are less costly and just as efficient. In particular, they could go straight to the person concerned and ask for the passphrase, whether cordially or not...



Drawing from XKCD, translated by us (<https://xkcd.com/538/>).

Use case: working on a sensitive document

10.1 Context

After a fresh start, the computer used to complete this project on page 71 was fitted with an encrypted system. Good. Then comes the need to work on a particular, more "sensitive" project, for example: page 119

- a leaflet must be drawn up;
- an affiche must be drawn;
- a book must be mocked up and then exported as a PDF ;
- an information leak must be organized to reveal a company's dreadful practices;
- a film is to be edited and burned to DVD.

In all these cases, the problems to be solved are more or less the same.

As it would be too much trouble to increase the overall level of computer security again, it was decided that this particular project should receive special treatment.

10.1.1 Vocabulary conventions

In the following, we will refer to them as :

- *work files*: all the files needed to produce the work (images or *rushes* used as bases, documents recorded by the software used, *etc.*);
- the *work*: the end result (leaflet, affiche, *etc.*).

In short, the raw material and the finished product.

10.2 Assessing risks

Let's now try to define the risks involved in working with a sensitive document.

10.2.1 What do we want to protect?

Let's apply the categories defined when we talked about risk assessment to the present case. on

page 63 :

- confidentiality: to prevent an unwanted eye from discovering the work and/or work files too easily;
- integrity: to prevent these documents from being modified without our knowledge;

- accessibility: ensuring that these documents remain accessible when needed.

Here, accessibility and confidentiality are top priorities.

Accessibility, because the main objective is to complete the work. If we had to travel to the North Pole to do so, the project would be in serious danger of falling through the (icy) cracks.

And when it comes to confidentiality, it all depends on how the work is advertised. So let's take a closer look.

Restricted works

If the content of the work is not completely public, or even perfectly secret, the idea is to conceal both the work *and* the working files.

Publicly distributed work

If the work is to be published, the question of confidentiality comes down to anonymity.

In this case, it's mainly the work files that will have to be swept under the carpet: discovering them on a computer strongly suggests that its owners have created the work... with potentially unpleasant consequences.

But that's not all: if the work, or its intermediate versions, are stored on this computer (PDF, *etc.*), their creation date is most likely recorded in the file system and in metadata. The fact that this date is prior to

page

24

30

publication of the work can easily lead opponents to draw uncomfortable conclusions about its genealogy.

10.2.2 Who do we want to protect ourselves against?

page

71

Let's take the threats described in the use case "a new beginning": the computer used to create the work can be stolen by the cops or during a burglary.

10.3 Which operating system is best for working on the document?

10.3.1 Decide which software you need

The first question is: what software will be used for this project?

- If the necessary software runs under GNU/Linux, let's continue reading this chapter to explore the options available to us.
- If these programs only work on Windows (or Mac OS), it's worth investigating whether similar programs are available for Debian GNU/Linux. If they exist, test them to see if they seem to work for this project.
- If only Windows software is really satisfactory, that's a pity. But we've come up with a practicable way of limiting the damage. So let's have a look at what this method looks like, ignoring the following paragraphs, which are devoted to GNU/Linux.

page 134

page

82

10.3.2 Use an amnesiac *live* system to leave as few traces as possible

One could imagine finely configuring an encrypted Debian system to keep as few traces as possible of our activities on the hard disk. The problem

of this approach is that it is of the "blocked list" type. We have explained [page 66](#) the limitations of this approach: no matter how much time you spend, no matter how much expertise you bring to bear, even with a particularly thorough understanding of the inner workings of the operating system, you'd always forget a small, well-hidden option. days of undesirable traces we hadn't thought of.

A chapter is devoted to [installing an encrypted Debian system](#), but it doesn't cover all the methods for limiting traces. Fortunately, some [amnesiac live systems](#) operate on the "authorized list" principle: unless explicitly requested, no traces are left on the hard disk. [page 119](#) [page 113](#)

On the basis of confidentiality alone, the *live* system beats the other by a landslide. On the other hand, while its main advantage is its amnesia, this can sometimes be a disadvantage. For example, if our favorite *live* system doesn't provide software that's essential to the project, you'll have to install it every time you start up, which can fortunately be done automatically.

If using a *live* system is therefore the safest solution, it's also the least difficult to implement, given the difficulty of installing a Debian system that leaves little trace. In the following section, we'll look at a security policy based on this solution. [next page.](#)

It should be noted that it is also possible to install a Debian system in a virtual machine to satisfy similar needs, but this solution is less suitable and will therefore not be detailed here. Documentation is available online, although it's important to take the same compartmentalization precautions for Debian as those described in the [chapter on creating a Windows virtual machine](#). [page 163](#)

10.4 Working on a sensitive document... on a *live* system

[page
79
-----]

Having set the scene at the start of this use case and decided to use a *live system*, we now need to implement this solution... and examine its limitations.

10.4.1 Download and install the *live* system

Not all *live* systems are designed for "sensitive" use. It is therefore important to choose a system specially designed to (attempt to) leave no trace on the hard disk of the computer on which it is used. This guide focuses on and documents the Tails *live* system.

If you don't already have a copy of the latest version of the Tails *live* system, follow the instructions for downloading and installing a "discrete" *live* system (see page 114).

Once our Tails device has been installed on our key, we can, if we wish, create an encrypted storage space to save some of our documents or settings. To do this, get the previously installed *live* system and start it up (see page 107). Then follow the instructions for creating and configuring a persistent volume in Tails (see page 116).

10.4.2 Install any additional software

If you need to use software that is not installed in Tails and you don't want to reinstall it each time, follow the recipe for installing additional software persistent in Tails (see page 117).

10.4.3 Using the *live* system

Each time you want to work on your document, you'll need the key containing your *live* system and its encrypted persistence to boot on it (see page 107). We then need to activate the persistent volume (see page 116).

10.4.4 Delete data

Once our project has been completed and printed or published online (see page 285), we can archive it (see page 89). We then need to delete the persistent volume (see page 116) containing the data.

10.4.5 There's more to come

We still have to clean up the metadata (see page 88) and study the limits of our approach (see page 88).

10.5 Working on a sensitive document... in Windows

[page
79
-----]

Having set the scene at the beginning of this use case, and despite all the problems of using Windows, let's now try to limit the damage a little.

10.5.1 Starting point: a colander and a box of dried patches

Let's start with a computer equipped, in the most conventional way, with a hard disk on which Windows has been installed. We won't dwell on this situation here, as the first part of this book has abundantly described the many problems it poses. In short, a sieve full of security holes.

We can therefore imagine sticking a few patches on this colander ¹. Let's take a quick tour.

A hard disk can be disassembled and hidden. That's true. But there are times when it's in use, sometimes for days or weeks at a time. This patch is based on two somewhat daring assumptions:

- *We're in luck.* It suffices in fact that the accident (search, burglary, etc.) occurs at the wrong time for all the desired confidentiality to be reduced to nothing.
- *Our discipline is perfectly rigorous.* Indeed, if you forget, or don't take the time, to "put away" the hard disk when you no longer need it, and an accident occurs at that point, it's game over.

There are also tools available for encrypting data under Windows. However much we trust them, the fact remains that they necessarily rely on the functions offered by the black box that is Windows. In any case, Windows will have *unencrypted* access to our data, and nobody knows what it might do with it.

To conclude this little tour of the court of dubious miracles, let us add that the only A possible "solution" in this case would be a blocked list approach, whose inefficacy has already been explained above

page 66

Now it's time to get down to business.

10.5.2 Second step: enclose Windows in an (almost) watertight compartment

What's beginning to look like a serious solution would be to run Windows in a watertight compartment, with a door opened, when necessary and with full knowledge of the facts, to allow it to communicate with the outside world in a strictly limited way.

In other words, set up a solution based on an *authorized list* logic: nothing can enter or leave Windows *a priori*, and from this general rule, *exceptions* are authorized on a case-by-case basis, with thought given to their impact.

Virtualization ² makes it possible to set up this type of system. It's a set of hardware and software techniques that enable several operating systems to run separately on a single computer, (almost) as if they were running on separate physical machines.

Nowadays, it's relatively easy to run Windows **inside a computer.** a GNU/Linux system, thereby cutting off all access to the network.
- and, in particular, by isolating it from the Internet.



Please note: it's advisable to read this entire chapter before rushing into the practical recipes; the description of the hypothesis that follows is quite lengthy, and its limitations are explored at the end of the chapter, where countermeasures are considered. It would be a shame to spend four hours following these recipes, before realizing that a completely different solution would, in fact, be more appropriate.

Let's start by summarizing the proposed hypothesis.

1. [Archive INA, La passoire des Shadoks](https://www.youtube.com/watch?v=1Duiup2tWKA) [https://www.youtube.com/watch?v=1Duiup2tWKA]
2. For more information, see the [Wikipedia](https://fr.wikipedia.org/wiki/Virtualization) page, 2020, *Virtualization* [https://fr.wikipedia.org/wiki/Virtualization].

[page 71]

The idea is therefore to run Windows in an *a priori* watertight compartment, **inside** an encrypted Debian system such as the one set up after reading the previous use case. What will serve as Windows' hard disk is in fact a large file stored on the hard disk of our encrypted Debian system.

Install the Virtual Machine Manager

We therefore first need to follow the recipe for installing the Virtual Machine Manager (see page 163). This software will be used to launch Windows in a watertight compartment.

Installing a "clean" Windows in the Virtual Machine Manager

Let's prepare a *clean* virtual disk image: to do this, follow the recipe for installing a virtualized Windows (see page 165). This recipe explains how to install Windows in the Virtual Machine Manager, cutting off all network access from **the outset**.

From then on, Windows is referred to as the *guest* system by the encrypted Debian system, which is the *host* system.

Install the necessary software in "clean" Windows

You might as well install all the *non-com-promising* software you need to create your premeditated works right now in your "clean" Windows.³ It'll save you having to redo it at the start of each new project... and, let's hope, avoid using a "dirty" Windows image for a new project, one day when time is running out.

Since the Windows *guest* is not allowed to leave the box to fetch files himself, it is necessary to send him the necessary software installation files from "outside".

We'll come back to this later, when we'll be sending it all kinds of files. For the moment, as we're in the process of preparing a "clean" Windows image to be used as the basis for each new project, let's not mix everything up, and content ourselves with sending it only what's necessary to install the desired non-compromising software.

Let's create a folder on the host system called *Windows Software*, and copy **just** the files needed to install the desired software.

Then share this folder with the Windows *guest*. To do this, follow the recipe for sharing a folder with a virtualized system (see page 171).

And as far as installing software inside the *guest* Windows: anyone sufficient enough to Windows to read these pages is, without a doubt, more competent than those writing these lines.

Please note: once this step has been completed, you must do **nothing** else in this virtualized Windows.



Take a snapshot of "clean" Windows

Now let's take a *snapshot* of the *clean* virtual machine we've just prepared. In other words, let's save its state in a corner. In future, this snapshot will serve as a starting point for each new project.

You therefore need to follow the recipe for taking a snapshot of a virtual machine (see page 168).

3. For example, if you want to hide the fact that you make films, having video editing software can be compromising.

New project, new start

Let's say you're starting a new project requiring Windows; here's how to proceed:

1. Restore the snapshot of the virtual machine containing the Windows installation.
2. The virtual machine can now be started up in its sealed compartment. It will be used **exclusively** for the new project, and now becomes a *dirty* virtual machine.
3. Within this new *sale* virtual machine, a new Windows user is created. The name assigned to it must be different **each time** a new project is started, and this user will be used **exclusively** for this new project. This is because software tends to register the name of the active user in the metadata of the files they save, page 30 and that it's best to avoid making unfortunate cross-checks possible.

The technical details of the first step are explained in the recipe for restoring the state of a virtual machine from a snapshot (see page 168). As far as the creation of a new user on the Windows version in use is concerned, the person reading this page will certainly be able to find it in the *Control Panel*.

Now that we have a watertight compartment, let's see how to open doors in it selectively, according to need.

How do I send files to the Windows guest? Since the *guest* Windows user is not allowed to leave the box to fetch files himself, it may be necessary to send him files from "outside", for example :

- raw material (*rushes*, images or texts from other sources);
- software required for the new project, and not present in the "clean" virtual image just restored.

We've already seen how to do this, but that was in a very specific case: installing new software in a "clean" *guest* Windows. Sharing files with a "dirty" Windows requires more thought and precautions, which we'll now explore.

The procedure is slightly different, depending on the medium on which the files to be imported are originally located (CD, DVD, USB stick, folder on the encrypted system's hard disk), but the usual precautions are the same:

- Windows should **only** have access to the files you want to import, and that's it. There's no question of giving it access to a folder containing a jumble of files relating to projects that shouldn't overlap. If that means starting with a sorting and tidying phase, so be it.
- When Windows needs to *read* (copy) the files contained in a folder, it is given *read-only* access to that folder. The less right you give Windows to write here and there, the fewer annoying traces it will leave.

To avoid mixing brushes, we recommend :

- create a **single** import folder for each project ;
- name this folder as explicitly as possible; for example: *Windows-readable folder* ;
- never share folders other than this one with the Windows *guest*.

Practical explanations are given in the recipe for sending files to the virtualized system (see page 171).

How do I get files out of the sealed Windows? By default, the Windows *guest* is not allowed to leave any traces outside its watertight compartment. But almost inevitably, the time comes when it's necessary to get files out of it, at which point we need to explicitly authorize it. For example :

- to take an exported PDF file to the copy-box, or to the printer;
- to screen the newly-released film on DVD.

To do this, we will export these files to an empty folder, dedicated to this purpose, and stored on an encrypted volume that can be :

- an encrypted USB key, activated under Debian by typing the corresponding passphrase ;
- the hard disk of the encrypted Debian, which here acts as office of the *host* system.

This dedicated folder will be shared with the Windows *guest*. Let's emphasize the words **empty** and **dedicated**: Windows will be able to read and modify everything in this folder, and it would be a shame to allow it to read files, when all you need to do is export a file.

If you need to burn a DVD, you can do so from Debian.

To avoid mixing brushes and limit contagion, we recommend :

- create a **single** export folder for each project ;
- name this folder as explicitly as possible; for example: *Folder where Windows can write* ;
- never share folders other than this one with the Windows *guest*, apart from the import folder recommended in the previous paragraph.

The recipes for sharing a folder with a virtualized system (see page 171) and for encrypting a USB key (see page 145) explain how to proceed in practice.

When the project is finished

When this project is finished, it's time to clean up, but first :

1. the resulting work is exported to the appropriate medium (paper, DVD, *etc.*), with the help of the previous paragraph, which explains how to output files from the *guest* Windows ;
2. working files are archived if necessary (the following use case coincidentally deals with this issue).

[page 89

Then it's time for the big clean-up, eliminating as many traces of the completed project as possible from the *host* system:

- the virtual disk image is restored to its "clean" state using the recipe for restoring the state of a virtual machine from a snapshot (see page 168);
- after a final check that everything that needs to be kept has been archived elsewhere, folders shared with Windows are deleted "for real" (see page 141);
- traces left on the hard disk are erased "for real" (see page 143).

Another new project?

If a new project comes along that also requires the use of Windows, let's **not** reuse the same *dirty* Windows. Instead, let's go back to the "new project, new start" stage.

[previous
page.

10.5.3 Third stage: possible attacks and countermeasures

The hypothesis we have just described is based on the use, as a system *host*, an encrypted Debian. All attacks on this encrypted Debian are therefore applicable to the present solution. Now it's time to study the attacks that can be used against this system.

Traces left on our encrypted Debian

Most of the most obvious traces of this project are separated from the rest of the system: all working files are stored in the file containing the virtual disk image. The name of the virtual machine, its configuration and periods of use, on the other hand, will leave other traces on our Debian system.

If disaster strikes during project execution The hard disk of the computer used contains the working files inside the virtual disk image.

If disaster strikes later Since the virtual disk image is properly cleaned when the project is completed, if disaster strikes later (giving in to the law, discovering a problem in the cryptographic system), the residual traces on the hard disk will be less obvious, and less numerous, than if we had proceeded in the ordinary way.

Even if the disaster occurs after the end of the project, i.e. after clean-up advised here, it would be unwise to feel immune, for as the beginning of this use case explains, the major drawback of the method described here is that it is based on the blocked list principle, a principle much decried on these pages... there will therefore always remain undesirable traces, which we hadn't thought of, on the hard disk of the computer used, in addition to those we are now familiar with: logs, random access and "virtual" memory, automatic backups.

If, despite these concerns, the hypothesis we have just described seems to be an acceptable compromise, it is now necessary to find out about the limitations shared by all the solutions considered in this use case.

If not, let's dig in.

Going further

Let's assume that one of the attacks described from the third stage of the use case

A "fresh start" seems credible. If successful, the contents of the *host* system's encrypted hard disk would be readable, in clear text, by the attacker. But our *guest* Windows... which is a silly file stored on the *host* system's hard disk. These work files, along with any traces recorded by the software used in Windows, then become readable by the attacker.

We're going to look at two ways of limiting the damage. One is to

One is a "blocked list", the other is an "authorized list".

Storing the virtual disk image off the *host* system's hard disk One idea is to store the virtual disk image used by the *guest* Windows system off the *host* system's hard disk. For example, on an encrypted external hard disk. This way, even if the *host* system disk is decrypted, our working files remain inaccessible... provided that the external hard disk containing them is not within the reach of the adversary at that time.

This is a "blocked list" approach, with all the problems that entails. Working files and the Windows system are not saved on the hard disk.

on the *host* system. But don't forget that this data will be used by the Virtual Machine Manager, which itself runs on the *host* system. As explained in the "Traces on all levels" chapter, various traces will therefore inevitably remain on **the internal hard disk of the computer in use.**

[page

27

To follow this trail :

[this page

- find out about the limits shared by all the solutions considered in this use case;
- see the recipe for encrypting an external hard disk.

[page 145]

Using a *live* system as a *host* system The counterpart to this approach "Blocked list" is an "authorized list" solution, combining the use of a *live* system and storage of the virtual disk image on an encrypted external hard disk.

To follow this trail :

[this page

- find out about the limits shared by all the solutions considered in this use case;
- See the recipe for encrypting an external hard disk, and the recipe for using a *live* system.

[page 145]

[page 113]

10.6 Clean up the metadata of the finished document

Once our document is complete, we export it in a format suitable for document exchange - for example, a PDF to print a text, an AVI or MKV file to publish a video on the Internet, *etc.*

Let's assume that we publish our document without taking any further precautions. Opponents who don't like it will probably start by downloading the document in search of any metadata that might link it to the people who produced it.

[page 185]

Despite the precautions we've already taken, it's a good idea to clean up any metadata that may be present.

10.7 Limits common to these safety policies

Any security policy based on this use case is vulnerable to a number of attacks, regardless of whether it uses a *live* system or the infamous Windows.

⌋ *The angles of attack* in the New Beginnings chapter examine some of the imaginable attacks, more or less science fiction, depending on time, place, protagonists and circumstances. The time has come to re-read them with fresh eyes.

[page

74

⌋ In addition, the "Issues" section of this volume dealt in relatively general terms with a number of surveillance methods, which it may be useful to revisit in the light of the concrete situation we're dealing with here; in particular, let's mention the issues of electricity, magnetic fields and radio waves, as well as the effects of various bugs.

[page

13

[page

21

[page

31

Use case: archiving a completed project

11.1 Context

A sensitive project is nearing completion; for example, a book has been designed and printed on page 79, or a film has been edited, compressed and burned to DVD.

In general, it will no longer be necessary to have permanent access to work files (high-resolution iconography, uncompressed *rushes*). On the other hand, it may be useful to be able to retrieve them at a later date, for example for a re-edition, an updated version, etc.

Since the more frequently a system is used, the more likely it is to be *attacked*, it's best to extract seldom-used information from the computer you use every day. What's more, it's easier to deny any link with files when they're stored on a USB stick in the woods, than when they're stored on the computer's hard disk.

11.2 Is this really necessary?

The first question to ask before archiving such files is: is it *really* necessary to keep them? When a piece of information *is* no longer available, no matter how hard anyone tries, no one will be able to provide it, and sometimes that's the best solution.

11.3 Assessing risks

11.3.1 What do we want to protect?

What happens to the requirements defined when we talked about risk assessment on page 63 as applied to this case?

- confidentiality: to prevent an unwanted eye from falling too easily on archived information;
- integrity: to prevent information from being modified without our knowledge;
- accessibility: ensuring that this information remains accessible when needed.

Here, accessibility is secondary to confidentiality: the whole idea of archiving is to make a compromise, by making access to data more difficult *for everyone*, in order to offer them better confidentiality.

11.3.2 Who do we want to protect ourselves against?

The risks envisaged in our "new beginning" are valid here too: a page 71

burglary, a search for reasons not directly related to the information we wish to protect.

Added to these risks is the possibility that the book or film produced may displease some commissioner, minister, CEO or similar. It happens. Let's say :

- this authority has learned of clues leading it to suspect who committed the masterpiece;
- this authority is able to dispatch a cohort of cops in the early hours of the morning to the homes of suspected criminals.

Such an untimely intrusion will, at the very least, result in the seizure of any computer hardware that may be discovered. This equipment will then be handed over to a computer expert, who will perform a kind of autopsy aimed at uncovering the data stored on it... or having been stored on it.

[page

42

11.4 Method

The simplest method at present is :

1. create an encrypted USB key or external hard drive (see page 145);
2. copy files to be archived to this device ;
3. delete and overwrite the contents of work files (see page 139).

Once these operations have been carried out, the key or hard disk can be stored in a place other than the computer you use most often.

CDs or DVDs could be considered for their low cost, but it is more complex to encrypt data correctly on these media than on USB sticks, which are now commonplace and easy to obtain.

11.5 What passphrase?

[page 103]

Since the files will be archived in encrypted form, it will be necessary to choose a passphrase. However, given that the purpose is archiving, this passphrase will not be used very often. And a seldom-used passphrase is likely to be forgotten... making it virtually impossible to access the data.

There are a number of possible solutions to this problem.

11.5.1 Write the passphrase somewhere

The difficulty lies in knowing where to write it down, where to store it so that it can be found again... without others being able to find it and identify it as a passphrase.

11.5.2 Use the same passphrase as for your daily system

[page 119]

The passphrase for your daily system, if it's encrypted, is one you type in regularly, and are likely to remember.

On the other hand:

- if you are forced to reveal the common passphrase, access to the archive also becomes possible;

- you need to have a **high level of** confidence in the computers you use to access the archives. Otherwise, the passphrase can be "stolen" without your knowledge, and used to read not only the archived information, but also all the data stored on your everyday computer.

11.5.3 Sharing the secret with others

A secret can be shared by several people. This requires several people to be present in order to access the archived content. This has to be weighed up: it can complicate the task, both for desired and unwanted access.

page 157

11.6 A hard drive? One key? Several keys?

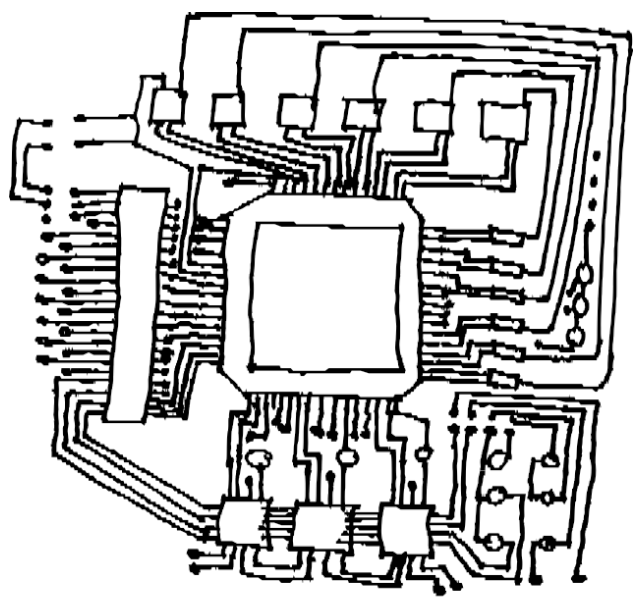
Depending on the choices you've made above, and in particular with regard to passphrases, you may be wondering which media to use. From a technical point of view, the simplest solution at present is to use a single passphrase per medium.

An external hard drive can hold more data than a USB stick, and is therefore sometimes necessary: to archive a video project, for example.

Archiving several projects on the same medium simplifies the task, but makes it difficult to separate projects according to the desired levels of confidentiality. Indeed, those who can access the archives of one project also have access to the others, which is not necessarily desirable.¹

Moreover, if the passphrase is a shared secret, we might as well make access easier for people who share the secret, by having a medium they can pass on to each other.

1. The subject of compartmentalization is developed in the chapter on contextual identities [page 246].



PART THREE

Tools

Introduction

In this third section, we'll explain how to apply some of the above ideas in practice.

This section is a technical appendix to the previous sections. Once you understand

Once you've chosen the answers on page 59 that are right for you, there's still the question of "How do you do it?" to which this appendix provides some answers.

For most of the recipes presented in this guide, we assume that you're using GNU/Linux with the GNOME desktop; these recipes have been written and tested under Debian GNU/Linux version 11 (nicknamed Bullseye) ¹ and Tails version 5 ² (*The Amnesic Incognito Live System*).

However, these are generally adaptable to other Debian-based distributions, such as Ubuntu ³ or Linux Mint ⁴.

If you are not yet using GNU/Linux, you can consult the use case a new start or use a live system.

[71](#) or use a live system.

Procedures are presented step-by-step, and wherever possible, the meaning of the proposed actions is explained.

The order in which each recipe is detailed is important. Unless otherwise stated, it is recommended not to skip a step and then go back. The result could be very different from the one expected.

Finally, it's important to use the most up-to-date version of this guide, as software evolves. It can be found on the <https://guide.boum.org/> website.

1. <https://www.debian.org/releases/bullseye/index.fr.html>
2. <https://tails.boum.org/index.fr.html>
3. <https://www.ubuntu-fr.org/>
4. <https://linuxmint.com/>

Using a terminal

C *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*


C *Duration: Fifteen to thirty minutes.*

A personal computer is often operated by clicking on menus and icons. However, there's another way to "talk" to it: by typing bits of text called "commands". The key to typing these commands is called "terminal", "*shell*" or "command line".

As often as possible, this guide seeks to circumvent the use of this tool, which can be rather confusing if you're not used to it. However, its use has sometimes proved indispensable.

12.1 What is a terminal?

A detailed explanation of how to use command lines is beyond the scope of this guide, and the Internet is full of tutorials and courses that do just that.¹ However, it did seem necessary to lay down a few basics on how to use them.

So let's start by opening a terminal: on a GNOME 3 desktop, open the Activities overview by pressing ( on a Mac), then type **term** and click on *Terminal*. A window showing :


```
IDENTIFIER @ THE-MACHINE-NAME :~$
```

At the end is a square, called a "cursor", which corresponds to the place where you enter the command text. In concrete terms, with the *rabouane* identifier on a machine named *debian*, you'll see :

```
rabouane@debian:~$
```



From this state, known as the "command prompt", you can directly type the commands you want the computer to execute.

The final effect of these commands is often the same as that obtained by clicking in the right place in a graphic interface.

For example, if you write **firefox** in the terminal you've just opened and then press **Enter** () to open the *Firefox* web browser.

However, we won't be able to enter a new command in our terminal until we quit the browser. We could have done exactly the same

1. Among others, a [page on ubuntu-en.org](https://doc.ubuntu-fr.org/console) [<https://doc.ubuntu-fr.org/console>] which itself ends with other links.

to do this, press  ( on a Mac) and type `nvram`, then click on *Firefox ESR*.

For the purposes of this guide, the main advantage of the terminal is that it allows you to perform actions that no other graphical interface currently offers.

12.2 About controls

Commands are like orders given to the computer via the terminal. These "command lines" have their own language, with their own words, letters and syntax. A few remarks on this subject are therefore useful.

12.2.1 Syntax

Take, for example, this command, `sfill`, which performs much the same operations as `nautilus-wipe`, a graphics tool that will be introduced later:

page 143

```

┌───┐ sfi  ┌─┐ ┌─┐ ┌───┐ /home
ll      option option argument
order


```

In this command line, we can see, in order :

- the *command* we call is `sfill`. The command is usually a program installed on the system;
- two *options*, `-l` and `-v`, which modify the behavior of the `sfill` program. These may be optional depending on the program (and begin with one or two dashes to distinguish them);
- a `/home` *argument* that specifies what the command will work on. There may be several, or none, depending on the command.

Each of these elements must be separated from the others by one (or more) space(s). So there's a space between the command and the first option, between the first option and the next, between the last option and the first argument, between the first argument and subsequent arguments, *and so on*.

There's no mystery when it comes to knowing the options and arguments of a command: each one normally has a man page. To access it, simply type `man` followed by the command name in Terminal, then press *Enter*.

( or `return`). The latter, however, can be difficult to understand because of their technical aspects, and are sometimes only available in English.

12.2.2 Inserting a file path

When using a terminal, you often need to specify folders and files. The term "path" is used to describe the folder and sub-folder in which a file is located. To separate a folder from its contents, we use the `/` character (pronounced "slash").



To give an example, here is the *path* to the `recette.txt` document in the Documents folder of the `alligator` account's personal folder:

```
/ home/ alligator/ Documents/ recette. txt
```

As many commands expect file names as arguments, it quickly becomes tedious to type their complete paths by hand. There is, however, a simpler way of inserting a path: when you grab a file icon with the mouse and drag it to drop it on the terminal, its path is written where the cursor is.

However, this only works with "real" files or folders. You'll get a weird name that won't work, for example, with trash files, the *Personal Folder* icon on the desktop or USB stick icons.

12.2.3 Execution



Once you've typed a command, you ask the computer to "execute" it, by pressing the *Enter* key () or ().

12.2.4 End or interrupt order

Command execution takes varying amounts of time. When it's finished, the terminal always returns to the state it was in before the command was issued, the "command prompt":



```
rabouane@debian:~$
```



The terminal is then said to "give back".

If you wish to interrupt the execution of a command before it has finished, you can press the  key, and while holding down the key press on the . This stops the command immediately, in the same way as when close a program window.



















12.2.5 Typography

Most of the symbols used to enter complete commands are common symbols. When a command uses the symbol "-", it is a "dash".

which can be obtained by typing (on a French keyboard) the . For a " ' " (right apostrophe) key. .

Other symbols are rarely used outside the terminal, but are available on standard keyboards. They are even indicated on the keyboard, and can be accessed using the button  on the right, . Here, based on a PC keyboard noted

French standard, the correspondence of some keys with the symbols they write, and their names (very few will actually be used in this guide):

Keys	Results	Symbol name
 + 	~	tilde
 + 	#	hash
 + 	{	left brace
 + 	[left hook
 + 		<i>pipe</i>
 + 	\	backslash
 + 	@	at
 + ]	straight hook
 + 	}	right brace

12.2.6 Names to be replaced

Sometimes, we specify that we're going to name something we've found so we can reuse it later. For example, we'll say that the identifier is LOGIN. Let's say we're working under the identifier daisy. When you write "type LOGIN, replacing LOGIN with your account ID", you'll actually be typing paquerette.

12.3 Administrative privileges

Some commands that modify the system require administrative rights. They will then have unrestricted access to the entire system, with all the risks that this entails.

To run a command with administrative rights, put `pkexec` before the command name. A window will then prompt you for a password before executing the command.

12.4 Warning

Even more so than with the recipes mentioned above, commands must be typed with the utmost precision. Forgetting a space, omitting an option, mistyping a symbol or being imprecise in an argument all change the meaning of the command.

And since the computer does *exactly* what is asked of it, if you change the command, it will do *exactly the opposite...*

12.5 An exercise

We'll create an empty file named "essai", which we'll then delete (without overwriting its contents).

In a terminal, enter the command :

```
> touch test
```

And press *Enter* (`↵`) or `return` to make the computer run it.

The `touch` command creates an empty file; the `essai` *argument* gives the name of the file. No options are used.

You can then check that this file has been created by issuing the `ls` command (which stands for "list):

```
> ls
```

Once the command has been issued, the computer responds with a list. On the one used for testing, this gives :

```
Test
office
```

`Bureau` is the name of a folder that already existed before, and `essai` the name of the file we've just created. Another computer might have responded with many other files in addition to `Bureau` and `essai`.

What the `ls` command answers is just another way of seeing what else is available. By clicking on the *Personal Folder* icon on the desktop, you'll see a new icon appear in the file browser, representing the `test` file you've just created.

We're now going to delete this file. The command line to do this has the general syntax :

```
rm [ options ] DELETE-FILE-NAME
```

We're going to use the `-v` option which, in the context of this command, asks the computer to be "verbose" about the actions it's going to perform.



To insert the name of the file to be deleted, we'll use the tip given above to indicate the file path. We'll then :

- type `rm -v` in our terminal,
- type a space to separate the `-v` option from the rest,

- in the *Personal Folder* window, drag the icon of the `test` file and drop it into the terminal.

At the end of this operation, we should obtain something like :

```
> rm -v '/ home/ LOGIN/ essai'
```

You can then press the *Enter* key ( or *teur* ) and note that the ordina- responds:

```
"home/ LOGIN/ test" deleted
```

This indicates that the requested file has been deleted. You can still check its absence by running a new `ls` :

```
> ls
```

We can see that there are no `tests` in the list returned by the command. On the same computer as before, this now gives :

```
Office
```

And the icon must also have disappeared from the file browser. Apparently, the file has therefore been deleted, even though, as explained in Part 1, its contents still exist on disk. As it was an empty file named "essai" we can say that it's not a big deal.

12.6 Watch out for tracks!

Most *shells* automatically save the command lines you type in a "history" file. This is handy for later retrieving the commands you may have used, but it also leaves a "history" file on the disk. trace of our activities.

The standard *shell* in Debian is called `bash`. With `bash`, to temporarily disable history recording in the terminal you're using, you suffit to do :

```
> unset HISTFILE
```

In addition, commands are stored in the hidden `.bash_history` file (located in the *Personal Folder*). You may therefore want to clean it out from time to time. page 141

12.7 Further information

This first experience with this window full of small print could be the start of a long passion. To nurture it, there's nothing better than taking the time to read the tutorial "Linux in text mode: console yourself!"² from the book *Linux aux petits oignons*, or the section "La console, ça se mange?"³ in the tutorial "Take back control with Linux!"

2. <https://fr.calameo.com/read/005322362565c72e1efe8>

3. <https://web.archive.org/web/20210920080224/https://sdz.tdct.org/sdz/reprenez-le-control-a-l-aide-de-linux.html#Laconsoleasemange>

Choose a passphrase

🔄 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

🕒 *Duration: Approximately ten minutes.*

A "passphrase" is a secret used to protect encrypted data. This is what is used to encrypt a hard disk or documents, or even, as we shall see in the second volume of this book, cryptographic keys.

We use the term "*phrase*" rather than "password", because a single word, however bizarre and complicated, is much less resistant than a simple multi-word phrase. A passphrase is considered to be made up of at least ten words. But the more, the better!

An important criterion that is sometimes overlooked: a good passphrase is one that you can *remember*.¹ This eliminates the need to write it down on paper, a serious mistake that renders obsolete the value of creating a concrete passphrase. But, just as importantly, a good passphrase should be *as difficult to guess as possible*. So let's avoid the passphrase made up of 15 words of random characters that you'll have forgotten barely 15 minutes after finding it, as much as the lyrics to an 80s disco hit.

One technique for finding a good passphrase that's difficult to guess, but no less easy to remember, would be to craft a phrase that doesn't come from an existing text. Indeed, whether it's song lyrics, the line of a poem, or a quotation from a book, tools like Project Gutenberg² make it increasingly easy to test passphrases taken from existing literature.³

However, the use of the expression "passphrase" can lead to a desire to choose a phrase that makes sense, which would have the disadvantage of losing the randomness that reinforces password security.

So you have to use your imagination when creating a passphrase, and here are a few tips on good habits to adopt when choosing a passphrase:

1. Choose ten words at random that have nothing to do with each other, for example by opening one or more books at random and keeping the first word on which our eyes fall.

1. Randall Munroe, 2014, *Password complexity* [<https://xkcd.lapin.org/index.php?number=936>].

2. Wikipedia, 2017, *Project Gutenberg* [https://fr.wikipedia.org/wiki/Projet_Gutenberg].

3. Dan Goodin, 2013, *How the Bible and YouTube are fueling the next frontier of password cracking* [<https://arstechnica.com/security/2013/10/how-the-bible-and-youtube-are-fueling-the-next-frontier-of-password-cracking/>].

2. Software often requires us to enter numbers or special characters. It is then possible to find things to modify in these words. Knowing that this step is really not necessary from a security point of view and risks above all making the sentence more difficult to remember. This can involve adding slang, words from different languages, putting capital letters or spaces where you wouldn't expect them, replacing characters with others, letting your imagination run wild when it comes to spelling, *and so on*.
3. Create a mnemonic to remember it. Example: embroidery
a narrative structure with these words can help to remember the pass phrase.

It's best to use only the characters found on all keyboard variants; in other words, avoid accented characters or any other symbols specific to local languages. This can avoid problems of missing or difficult to find keys, and above all of incorrect character encoding, if we have to type our passphrase on a different keyboard from the one we're used to.



TO FIND OUT MORE...

You can also use the KeePassXC password manager (see page 355) to generate a passphrase of ten random words.

By default, this tool includes an English word list, but you can specify another word list⁴ by adding it, in the form of a simple text file containing one word per line, to the `/usr/share/keepassxc/wordlists` folder. This operation must be performed as a superuser.

Next, start KeePassXC and go to the *Tools* menu, then *Password Generator*. In the *Passphrase* tab, you can choose the list of words to use (if several are available) and the number of words. The generated passphrase appears above.

The number of words required to make a passphrase difficult to guess varies according to the size of the word list. The *Entropy* indicator, located on the right, below the passphrase, thus gives a measure of this difficulty: the greater the entropy, the better. A good passphrase requires an entropy of around 128 bits.

For example, find ten words at random:

seem bridge brake payante outgoing ostrich date licenses degauchir
piller

If a program requires you to add symbols or numbers, you can make sentences without complicating your life too much. For example:

Sembler bridge frein payante. Out ostrich, dater licenses! degauchir
+piller-1984

And we can imagine a sentence, with these words, that serves as a mnemonic:

It may seem that playing bridge puts a damper on things, because it's not free. Take out your ostrich and date your licenses! Planing and plundering, unsupervised

4. For example, you can use [the list of French words proposed by mbelivo \[https://raw.githubusercontent.com/mbelivo/diceware-wordlists-en/master/wordlist_fr_5d.txt\]](https://raw.githubusercontent.com/mbelivo/diceware-wordlists-en/master/wordlist_fr_5d.txt). However, it needs to be adapted to the format used by KeePassXC, which can be done with the command executed in a terminal from the folder where the file in question is located:

```
cut -d' ' -f2 < wordlist_fr_5d.txt > wordlist_fr_5d_keepassxc.txt
```

You can then choose to use the list of random words only, or the complete sentence as your passphrase. In the latter case, however, you'll need to be careful about the use of special characters, as mentioned above.

Once the data has been encrypted with our new passphrase, it's a good idea to use it a dozen or so times to decrypt the data. You can even write it down on a piece of paper when you create it, to make sure you remember it when you use it for the first time (of course, you'll need to destroy the paper afterwards). This will allow you to teach your fingers how to type this new phrase, and thus memorize it mentally and physically.

Finally, let's not forget that if finding such a passphrase is not effortless, you also need to find a different one for each medium you're encrypting. Using the same passphrase, or worse, the same password, for a variety of different things, can prove disastrous if it is revealed.

What's more, you should never use a passphrase as a password for an online service that is also used to lock a cryptographic secret. Indeed, if this online service were to be hacked, our passphrase would then be known to the hackers and potentially sold to others.

Booting from CD, DVD or CD-ROM USB key

🔄 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

🕒 *Duration: Approximately one minute to twenty minutes.*

Here we'll look at how to boot a computer from external media, such as a Debian installation CD, or a *live* system on a USB stick.

Sometimes, especially on modern computers, it's quite simple. Other times, it's a bit hair-raising...

When a computer is started up, the firmware (BIOS or UEFI) runs first. **20** runs first. As we've seen, it's this that allows you to choose the device (hard disk, USB key, CD or DVD, *etc.*) where the operating system to be booted is located.

page

14.1 Try naively

Start by putting on the external media, then (re)start the computer. Sometimes it works by itself. If that's the case, you're in luck: there's no need to read the rest of this chapter!

14.2 Attempt a one-time selection of the boot device

With recent firmware, it's often possible to choose a boot device on a case-by-case basis. But this isn't always possible, especially for certain computers equipped with Windows (version 8 onwards), for which the handling is more complicated. Among other things, you'll have to disable *Secure Boot*¹ and probably search the Internet to find out how to boot onto a USB stick with this particular model of computer.

(Re)start the computer, looking carefully at the very first messages that appear on the screen. Look for messages in English that look like :

- Press [KEY] to select temporary boot device
- [KEY] = Boot menu
- [KEY] to enter MultiBoot Selection Menu

These messages tell you to use the KEY key to select a boot device.

This key is often **F2** or **F12** or **F9** or **Escap**.

On Macs, there is an equivalent to this possibility: immediately after the allu-computer, press and hold the key **alt** key (sometimes also

1. [How to disable secure boot \[https://doc.ubuntu-fr.org/desactiver_secure_boot\]](https://doc.ubuntu-fr.org/desactiver_secure_boot).

`[option]` marked `[]`). After a while, you should see the *Startupmanager*².

But back to our PCs. Often, the firmware goes too fast, so you don't have time to read the message, understand it and press the key. Once the right key has been identified, restart the machine and press the key in question (don't hold the key down, but press and release it several times) as soon as the computer is switched on.

If all goes well, a message like this will affiche :

```
+-----+
| Boot Menu |
+-----+
|
| 1: Removable Devices
| 2: Hard Drive
| 3: DVD - ROM
| 4: Network boot
|
| <Enter Setup
|
+-----+
```

If it works, you're in. Choose the right entry in this menu, moving around with the keyboard arrows `[↑]` and `[↓]`, then press *Enter* (`[↵]` or `[return]`). Often, we have to guess the term used by the firmware to designate our device. For example, to boot on a USB stick, select **Removable Devices**. The computer will boot on the selected device. There's no need to read on!

14.3 Modify firmware parameters



Firmware is used to configure the computer's hardware operation. It's a good idea not to make so many changes all at once, but to write them down on a piece of paper. That way, if the computer stops working, you can go back to the way things were. If in doubt, exit without saving and start again.

If choosing a temporary boot device doesn't work, you'll need to enter the firmware to manually set the boot order. The firmware tests the devices in the configured order and boots the first operating system found. The aim of this modification is to put our external media at the top of this list.

To spice things up a bit, the firmware programs are almost all different, so it's impossible to give a recipe that always works.³

14.3.1 Enter firmware configuration interface

Once again, it's a matter of (re)starting the computer by looking carefully at the first messages that affich on the screen. Look for messages in English that look like :

- Press `[KEY]` to enter setup
- Setup: `[KEY]`
- `[KEY]` = Setup
- Enter BIOS by pressing `[KEY]`

2. http://support.apple.com/kb/HT1310?viewlocale=fr_FR

3. Illustrated tutorials for some BIOS are available on [this page \[https://www.hiren.info/pages/bios-boot-cdrom\]](https://www.hiren.info/pages/bios-boot-cdrom).

- Press [KEY] to enter BIOS setup
- Press [KEY] to access BIOS
- Press [KEY] to access system configuration
- For setup hit [KEY]

These messages tell you to use the KEY key to enter the firmware.

This key is often (←→) (Delete) (Delete Del) or (F2) sometimes (F1), (F10), (F12), (Escap), (Tab) or (→).

Here's a table summarizing firmware access keys for some common computer manufacturers ⁴.

Manufacturer	Keys observed
Acer	F1, F2, Delete
Compaq	
Dell	F1
Fujitsu	F10
HP	F2, F2, F10, F12, Escap
IBM	F2
Lenovo	F1 Input ()
NEC	F1
Packard Bell	F1, ↓
Samsung	F1, F2, Delete
Sony	F2, F2, F12, Escap
Toshiba	F1

Often, the firmware goes too fast, and you don't have time to read the message, understand it and press the key. Once you've identified the right key, restart the machine again by pressing the key in question (don't hold down the key, but press and release it several times). Sometimes the computer gets lost and crashes. In this case, restart and try again...

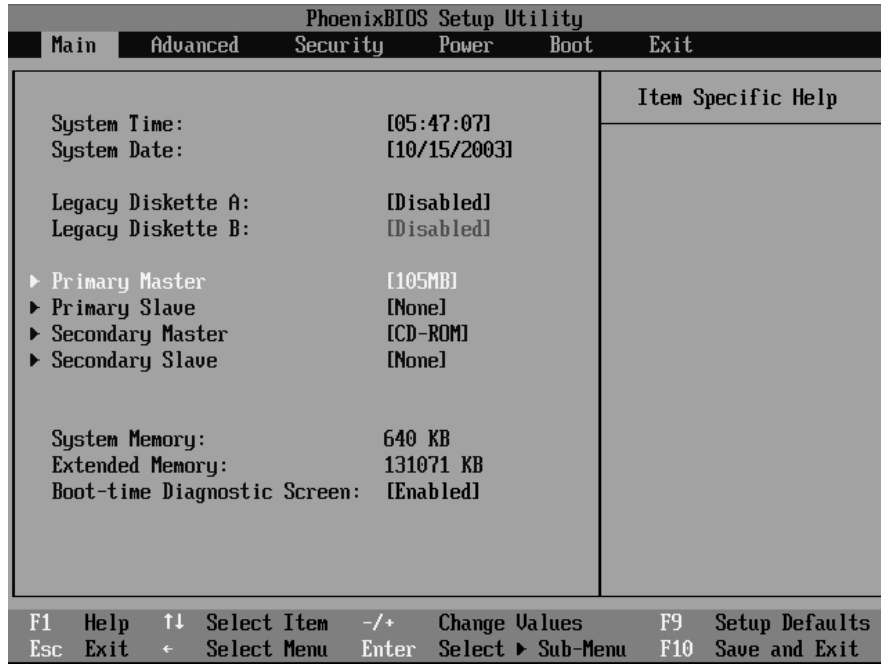
If an image affiche instead of the message you're hoping to see, it may be that the firmware is configured to afficher a logo rather than these messages. Try press (Escap) or on (Tab) (←→ or →) to view messages.

If the computer starts up too quickly for you to read the messages it affiche, it is sometimes possible to press the key (Break) key (often at the top to the right of the keyboard) to freeze the screen. Pressing any key again can to "unfreeze" the screen.

14.3.2 Using the firmware configuration interface

Once inside the firmware, the screen is often blue or black, full of menus and sometimes the mouse doesn't work. Usually, an area at the bottom or right of the screen explains how to navigate between options, how to change tabs *and so on*. It's often in English: help is "help", key is "key", select is "select", value is "value" and modify is "modify". The keys to be used to move are usually described too, for example ←↑↓→: Move (in English, "move"). These are the keyboard arrows ↓ | and ↑ | and/or ← and →. Sometimes, the (Tab) (←→ or →) is also useful.

⁴ Tim Fisher, 2019, *BIOS Setup Utility Access Keys for Popular Computer Systems* [[https:// web.archive.org/web/20200227083303/https://www.lifewire.com/bios-setup-utility-access-keys-fo r-popular-computer-systems-2624463](https://web.archive.org/web/20200227083303/https://www.lifewire.com/bios-setup-utility-access-keys-for-popular-computer-systems-2624463)] (archive), and Michael Stevens Tech, 2014, *How to access/enter Motherboard BIOS* [https://web.archive.org/web/20201128221653/http://michaelsteventech.com/bios_manufacturer.htm] (archive).



A BIOS screen

14.3.3 Modify startup sequence

The idea is to rummage around until you find something that contains the word boot, and looks like :

- First Boot Device
- Boot Order
- Boot Management
- Boot Sequence

If not, try something like Advanced BIOS Features or Advanced features.

Once you've found the right input, you need to figure out how to modify it. For example, Enter: Select or +/-: Value. The aim is to put the CD/DVD or USB key first, depending on which you want to boot from.

Sometimes, you need to enter a sub-menu. For example, if there is a Boot order menu and it is written in Enter: Select help, press Enter () or once () a the menu has been selected.

Other times, options can be changed directly. For example, if there's an option like First boot device and it's written in the +/-: Value, press the key key or the key until the correct value is displayed, e.g. IDE DVDROM, is selected. Sometimes the Page Down key is used instead, () and Previous Page (Page Up, or) are used. Sometimes also, keys like F5 and F6. And other times, these keys are used to move the device up and down in a list corresponding to the startup order.

14.3.4 Choosing your new configuration

Once you've managed to select the right media for booting, you need to ask yourself whether you want to leave it in forever or not. If you want to leave it, it may be useful to place the internal disk second in the boot sequence. In this way, if the medium placed first is absent, the computer will boot from that disk.

If you don't include the internal disk in the boot sequence, the computer won't boot from it, even in the absence of a CD, DVD or USB key.

However, leaving your computer to boot up on an external medium can have unfortunate consequences: it becomes a little easier for a malicious person to start it up using this medium, for example, to carry out an attack.

It's true that firmware can be used to set up a password for access to the computer, which must be entered before starting up. But there's no point relying on this to protect anything: most of the time, this protection can be easily bypassed, for example by removing the battery from the motherboard for a few minutes.

14.3.5 Save and exit

Once the new configuration has been set, save and exit. Once again, read the on-screen help, such as **F10: Save**. Sometimes, you may need to press one or more times **Esc** to get the right menu. A message will then appear asking (in English) if you are sure you want to save and exit. For example:

```
+-----+
|           Setup Confirmation           |
+-----+
| Save configuration and exit now       |
|                                     |
|      <Yes >           <No>         |
|                                     |
+-----+
```

We want to save, so we select **Yes** and press *Enter*.

(**↵** or **return**).

Using a *live* system

🔄 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

🕒 *Duration: Thirty minutes to an hour, plus about thirty minutes download time.*

A *live* system is a GNU/Linux system that runs without being installed on the computer's internal hard disk.

Please note that this does not mean that there will be no traces on the internal disk:

For example, many *live* systems use virtual memory (*swap*) if they detect the possibility of doing so. In addition, some *live* systems allow automatic access to the contents of the internal disk, which is also likely to leave some residues.

traces.

15.1 Discrete *live* systems

On the other hand, some *live* systems are specially designed to (attempt to) leave no trace on the hard disk of the computer on which they are used, unless they are expressly asked to do so. This is the case, for example, with Tails (*The Amnesic Incognito Live System*).

There is then (if the people behind the *live* system have got it right) nothing written to the internal disk. Everything done from the *live* system will be only written in RAM, which is more or less erased for real when the computer is turned off, at least after a certain period of time.

Using such *live systems* is therefore one of the best ways of using a computer without leaving any traces. Here, we'll look at how to get a *live* system, and how to boot into it.

The usual way to use a *live* system is to install it on a USB key or burn it onto a DVD.

It's generally advisable to use Tails on a USB stick: this allows you to use certain features not available on DVD, such as automatic updates and persistent space.

However, given that it is possible to write data to a USB key, but not to a DVD, this makes it possible for malicious people to modify our *live* system to, for example, record our passwords or keystrokes. If, for these reasons, you choose to use a DVD, you'll need to make sure you update it manually, otherwise you'll be using a

system with known vulnerabilities!

15.2 Download, check and install Tails

We'll explain here how to download the latest version of Tails from its official website, then how to check its authenticity before installing it on a USB stick or burning it to a DVD. We rely mainly on the official wizard available on the <https://tails.boum.org/install/index.fr.html> web page, which offers several different documentations depending on the operating system you're using.

If you already have an installation of the latest version of Tails, you can simply duplicate it. To do this, follow the Tails clone tool.



Please note: this guide provides further explanations on verifying the authenticity of the Tails image. When you get to the "Verify your download" section of the official Tails documentation, refer to the [verify live system authenticity](#) part of this chapter.

15.2.1 Download Tails

Tails can be downloaded in two ways: either directly *via* a web browser (in HTTPS), or using BitTorrent.

Whichever method you use, you need a disk image of the Tails system ¹ image of the Tails system and the corresponding OpenPGP signature to [verify its authenticity](#).

With a web browser, the two will have to be downloaded separately, whereas BitTorrent will retrieve them at the same time.

In all cases, you'll need to follow the [Tails installation wizard \[https://tails.boum.org/install/index.en.html\]](https://tails.boum.org/install/index.en.html) for the operating system you're using.

15.2.2 Verify the authenticity of the *live* system

The Tails installation official wizard (if you're not using the command-line method) offers an automatic tool for checking the integrity of the downloaded file. It tells you to click on a *Select your download...* button and then performs an initial check of the downloaded image. In particular, it guarantees that ² that the image corresponds exactly to the one distributed by the Tails site. However, it does **not protect** against an [attack on the Tails site](#).

The *live* system image we've just downloaded is digitally signed using OpenPGP. We're going to use this signature to verify its authenticity more robustly. If you haven't already downloaded this signature, you can obtain it by clicking on the *OpenPGP signature* link in the *Check your download* section, then on *OpenPGP signature* again in the box that appears.

Next, you need to download the OpenPGP key for signing Tails. To do this, always in the *Check your download* section, first click on *OpenPGP signing* to display the corresponding box (if not already visible), then click on *OpenPGP signing key*. This key is associated with the address `tails@boum.org`.

Once downloaded, we import this OpenPGP public key into the desktop keychain. We can then afficher the fingerprint of this key by double-clicking on it in *Kleopatra*. The fingerprint observed by the people who wrote this guide is as follows (assuming it's an original copy we have in our hands):

1. A disk image is an *archive file* containing an identical copy of a storage system (CD, DVD, hard disk, USB stick, *etc.*). It is often used to transfer and duplicate system installation files. A disk image can have different formats, such as `.img` or `.iso` (referred to as an ISO image).

2. The threat model addressed by Tails' download verification system is documented [on the Tails website \[https://tails.boum.org/contribute/design/download_verification/\]](https://tails.boum.org/contribute/design/download_verification/).

```
A490 D0F4 D311 A415 3 E2B B7CA DBB8 02 B2 58 AC D84F
```

If the observed fingerprint is the same as this one, then the digital signature of the image can be verified. *Kleopatra* can afficher *Impossible to verify the data Signature created on [...] With the certificate: Tails developers (offline long-term identity key)*

page 345

<tails@boum.org> (DBB8 02B2 58AC D84F). This means that the file is indeed signed by the key in question, but that we haven't confirmed the authenticity of this key... it's no big deal because we've just verified its fingerprint.

If the signature is valid, there's a very high probability that the Tails download just performed is a good one. Firstly, its integrity has been verified, so the image is exactly the same as the one proposed by the site. What's more, it is signed with a key whose fingerprint can be verified in this guide, i.e. elsewhere than on the Tails site. As the probability that the site and the guide have been corrupted in the same way is very, very low, you can continue with the installation.

15.2.3 Install Tails on the chosen support

Return to [the Tails installation wizard \[https://tails.boum.org/install/index.en.html\]](https://tails.boum.org/install/index.en.html) to find the instructions for installing Tails on a USB key for our operating system.

If, on the other hand, you prefer to install Tails on DVD, go to the [dedicated page \[https://tails.boum.org/install/dvd/index.fr.html\]](https://tails.boum.org/install/dvd/index.fr.html).

15.3 Cloning or updating a Tails key

Once you have a DVD or USB key with Tails, you can duplicate it, for example to create a USB key with persistence corresponding to a new contextual identity, to give a Tails key to an acquaintance, or to update a USB key containing an older version of Tails.

previous
page.

To do this, we follow the official Tails documentation, which is available from any Tails DVD or key, even without an Internet connection.

Start Tails first. Then double-click *the Tails documentation* icon on the desktop. Look for the *Downloading, installing and updating* section, then the *Installing by cloning from another Tails* item. Click on *For PC*, or *For Mac*, depending on your computer. Follow the steps indicated.

this page

To update the key thus created, you will then need to follow the *Upgrading automatically* page, located under the *Upgrading a Tails USB stick* item.

15.4 Booting on a live system

As soon as copying or burning is complete, you can restart your computer, leaving the *live* system media inside, and check that the copy has worked. Provided, of course, that you have configured your computer's firmware to boot on the correct media: see [the recipe explaining how to boot on external media](#) for details.

page 107

On startup, Tails affiches a screen that lets you choose, among other options, the affichage language and keyboard layout.

15.5 Using Tails persistence

[this page] When using Tails from a USB stick, it is possible to create an encrypted persistent volume on the stick's free space.

The data contained in this persistent volume is backed up and remains available from one Tails session to the next. The persistent volume can be used to back up personal files, encryption keys, configurations or software not installed in Tails by default.

[this page] Once the persistent volume has been created, you can choose whether or not to activate it each time Tails is started.

[this page] Finally, you can delete it when you no longer need to access the data it contains.

However, using a persistent volume is not without consequences in terms of the traces left behind. That's why you should start by reading the warning page concerning the use of persistence.

To do this, double-click on the *Tails Documentation* icon on the desktop. Look for the section *Getting started with Tails* and click on *Warnings about persistent storage*, located just below the *Persistent storage* item.

15.5.1 Creating and configuring a persistent volume

The aim of this recipe is to create and configure a persistent volume on a Tails key.

To do this, we'll follow the official Tails documentation, which is available from any Tails USB stick or DVD, even without an Internet connection.

[previous page.] Start Tails first. Then double-click on the *Tails Documentation* icon on the desktop. Look for the section *First steps with Tails* and click on *Persistent storage*. On this documentation page, follow the sections *Create persistent storage* and *Configure persistent storage*.

If you already have a persistent volume and simply wish to modify its parameters, such as the passphrase, go directly to the *Advanced topics* section at the bottom of the documentation summary page.

15.5.2 Activate and use a persistent volume

The aim of this recipe is to activate the newly created persistent volume on our Tails key.

To do this, we'll follow the official Tails documentation, which is available from any Tails USB stick or DVD, even without an Internet connection.

[previous page.] Start Tails first. Then double-click on the *Tails Documentation* icon on the desktop. Look for the section *First steps with Tails* and click on *Persistent storage*. On this documentation page, follow the section on *Using persistent storage*.

15.5.3 Delete a persistent volume

The aim of this recipe is to delete the persistent volume previously created on our Tails key.

To do this, we'll follow the official Tails documentation, which is available from any Tails USB stick or DVD, even without an Internet connection.

[previous page.] Start Tails first. On the desktop, double-click on the *Tails Documentation* icon. Look for the section *First steps with Tails*, click on *Delete persistent storage* located under the *Persistent storage* item, then follow this documentation page.

15.5.4 Installing additional persistent software in Tails

Tails contains software suitable for most common Internet and document creation tasks. However, for specific projects, you may need to install specific software in Tails, such as electronic circuit design and simulation software.

When Tails is installed on a USB stick, a persistent volume can be set up so that one or more specific software programs are installed automatically at each startup.

Find the name of the package to be installed We need the exact name of the package to be installed. To find it, follow the [find a package recipe](#). For example, our electronic circuit design software is provided by the `geda` package.


[page 135]


Installing the additional software To install the package thus identified, we'll follow the official Tails documentation, which is available from any Tails USB key or DVD, even without an Internet connection.

Start Tails first. On the desktop, double-click on the *Tails documentation* icon. Look for the section *First steps with Tails*, click on *Install additional software* and then follow this documentation page.

[page 115]

Installing an encrypted system

 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

 *Duration: One day, with several (sometimes long) waiting periods.*


We've seen that all computers - with the exception of certain *live* systems - leave traces all over the place, of files opened, jobs carried out, Internet connections, and so on. We have also seen that one way of exposing a little less stored data on the computer and the traces we leave on it is to encrypt the entire system on which page 47 we're working.

It is possible to install a GNU/Linux operating system such as Debian or page 22 Ubuntu, on an encrypted part of the hard disk. Each time it starts up, the computer will request a passphrase, after which it unlocks the disk's encryption, giving access to the data and thus enabling the system to boot. Without this passphrase, anyone wishing to consult the contents of the disk will be faced with indecipherable data. This is what we intend to do in this recipe.

Installing a new operating system can delete all the data on your hard disk. The first step is to back up the data you want to keep. Then, if you consider that the hard disk contained sensitive data, you can erase them "for real" to make their recovery as difficult as possible. page 151

page 139

16.1 Limits

 **Please note:** this simple encrypted installation does not solve all confidentiality problems with a wave of a magic wand. It only protects data under certain conditions.

16.1.1 Limits of an encrypted system

We highly recommend the following background reading:

- the chapter on encryption (and its limitations),
- the use case of a new start, which examines in detail the practical limits of such a system and possible attacks on it. on page 47 page 71

Without this, installing an encrypted system can create a false sense of security, which can lead to many problems.

16.1.2 Limits of a new installation

When installing a new system, you're starting from scratch. There's no easy way to check that the installation medium you're using is reliable, and doesn't contain malware, for example. You may only find out *later* - and then it may be too late...

16.1.3 Limits to equipment handling

Using a free operating system like Debian has one disadvantage: hardware manufacturers generally pay little attention to it. It can therefore be difficult, if not impossible, to use a computer or one of its peripherals with Debian.

[page

20

The situation has improved over the last few years: hardware operation is tending to become more homogenous, and above all, the spread of open-source systems is increasingly encouraging manufacturers to help, directly or indirectly, to ensure that their hardware works.¹

[page 113

However, before you replace an operating system, it's a good idea to make sure that the necessary hardware is running properly, using a *live* system. The Tails system, for example, is based on Debian. Hardware that works with one should therefore work with the other without too many difficulties. Bear in mind, however, that Tails includes non-free firmware, whereas you have to install it explicitly to have it in Debian.

[this page

16.2 Download installation media

The easiest way to install the system is to use a USB stick, CD or DVD. Debian offers several variants, so it's a good idea to start by choosing the method that best suits your situation.

16.2.1 With or without non-free firmware?

[page

20

In order to function, certain computer peripherals require "*firmware*" from the system. But free versions are not always available...

A micro-what?

These are programs that run on electronic chips inside the device, rather than on the computer's processor. This is the case, for example, with the program that controls the movement of the mechanical parts of a hard disk, or the operation of the radio system on a Wi-Fi card. We don't necessarily realize that they exist, as most hardware is delivered with the firmware already installed.

[page

16

For other peripherals, however, the operating system must send the firmware to a component during initialization.

Free firmware is supplied with the Debian installation program. As most firmware is not free, we have to provide the installer with any non-free firmware needed to run the computer: this is typically the case for certain Wi-Fi cards.

Another story of compromise

[page

39

If we install our encrypted system on a laptop, it's very likely that additional firmware will be needed to get Wi-Fi working, or even to have a good-quality affichage.

1. For some hardware, problems can arise from faults in the operation of built-in firmware. These problems are sometimes corrected by updates provided by manufacturers. It may therefore be a good idea to update the firmware (BIOS or UEFI), the *Embedded Controller* or other components before proceeding with installation. Unfortunately, these procedures differ too much from one hardware to another to be detailed in this book, but can generally be found on the manufacturer's website...

On a fixed computer without Wi-Fi, it's quite plausible that our encrypted system should work correctly without necessarily having non-free firmware.

Although we know of no evidence of its use, it's conceivable that a Wi-Fi card's proprietary firmware could spy on us without our knowledge... except that without firmware, it simply won't work. Once again, it's a question of compromise.

16.2.2 Network installation image

The quickest way is to use a network installation image. This contains only the very first parts of the system. The installer then downloads the software to be installed from the Internet. The computer on which you wish to install Debian must therefore be connected to the Internet, preferably via a network cable (and not via *Wi-Fi*, which will rarely work inside the installer).

There are several files (also called "images") containing a copy of the installation image, depending on processor architecture. In most cases, [page 16](#) you'll need to download the one ending in `amd64-i386-netinst.iso`, known as multi-architecture, which is suitable for both 32-bit and 64-bit architectures, and which will work with all your processor architectures. on most home computers manufactured after 2006².

Choose between :

- the completely free version³
- the version containing non-free firmware⁴.

16.2.3 The image with the graphic environment

If it's not possible to connect the computer on which you want to install Debian to the Internet, you can download an image containing the entire base system and the usual graphical environment. This requires access to a DVD burner or a USB key of at least 4 GB.

In the same way as for the network installation image, you must choose between⁵ :

- the completely free version⁶
- the version containing non-free firmware⁷.

Only the first DVD is required for installation. The name of the file to be downloaded ends with `-amd64-DVD-1.iso` (64-bit).

16.3 Check the footprint of the installation image

It's a good idea to make sure that the image download has been successful by checking the installer's fingerprint, to ensure its integrity and authenticity. We [page 47](#) will proceed in two stages, the first ensuring its integrity, the second its authenticity.

authenticity.

To do this, you need to boot on a system that's already installed. If you have access to a GNU/Linux computer, such as a friend's, you're in the clear. If you only have a *live* system, for example, you can install the image

2. Laptops using ARM processor architecture [[page 16](#)] are appearing, but the authors of this guide have never yet tested one.

3. <https://cdimage.debian.org/cdimage/release/current/multi-arch/iso-cd/>

4. <https://cdimage.debian.org/cdimage/unofficial/non-free/cd-including-firmware/current/multi-arch/iso-cd/>

5. These DVDs work with computers with x86-64 processor architecture, i.e. the vast majority of computers manufactured after 2012.

6. <https://cdimage.debian.org/debian-cd/current/amd64/iso-dvd/>

7. <https://cdimage.debian.org/images/unofficial/non-free/images-including->

downloaded to a USB stick, then check the fingerprint from the *live* system.

To verify the integrity and authenticity of the ISO image, two small files are required:

- the checksum SHA512SUMS ;
- the signature of this checksum SHA512SUMS.sign.

Download them from the page on which you found the ISO image above by right-clicking and selecting *Save link target as...*


16.3.1 Check the integrity of the installation image

[page 161] To do this, follow the checksum tool. It will be necessary to calculate the SHA512 checksum of the installation image (the ISO image), and then check that it matches that contained in the SHA512SUMS file.

16.3.2 Verify the authenticity of the installation image

If the integrity check has been successful, i.e. if the two calculated checksums match, the process can be continued to verify its authenticity. Indeed, adversaries could provide corrupted installation media and checksums. The previous check would simply show us that the downloaded file is the one available on the website, not the one we hope to have.

The second volume explains how to ensure the authenticity of the downloaded installer, since the fingerprint is signed with GnuPG, which uses asymmetric cryptography. The following tools are required:

- Download the public key used to sign the installation media from <https://keyserver.ubuntu.com/pks/lookup?op=get&search=0xdf9b9c49eaa9298432589d76da87e80d6294be9b> and save it with  then *Save as...* Choose **debian.asc** as file name and *Save*.
- Import this key into the office keychain. Check fingerprint:
 - if you have access to a trusted Debian installation, you can install the package **debian-keyring**, then use a terminal and type the following command:

[page 343]

[page 135]

[page 97]



```
gpg --keyring /usr/share/keyrings/debian - role -
keys.gpg - default - keyring --
S fingerprint debian -
cd@lists.debian.org
```

- if you trust the book in your hands, it claims that the print is: DF9B 9C49 EAA9 2984 3258 9D76 DA87 E80D 6294 BE9B.

[page 345]

- Check the signature of the SHA512SUMS file, contained in the file SHA512SUMS.sign previously downloaded. The notification must afficher *Signature valid butnot reliable of Debian CD signing key <debian-cd@lists.debian.org> on [...]*.

16.4 Prepare installation support

Once the installation media image has been selected, downloaded and verified, all that remains is to install it on a USB key, CD or DVD.

16.4.1 Create an installation USB key

If you have an empty USB stick, or one containing only data you don't really want, and you have access to a GNU/Linux-based system such as Debian⁸ or Tails, this is the fastest option.

[page 113]



8. Occasionally, the computer may fail to boot from the USB stick produced by following the instructions described here. However, from what we've been able to experience with





Please note: any data on the key will be lost. On the other hand, if this key was not initially encrypted, it would be possible to carry out an analysis.

to find files whose contents have not been overwritten before...


page 42

Open Disks from the Activities overview: press  ( on a Mac), then type `disk` and click on *Disks*.

Once Disks is open, you can plug in your USB stick. An entry corresponding to it should appear in the list on the left. Click on it to select it.

Then click on the  menu in the top right-hand corner (or ) and select *Restore disk image*.... In *Image to restore*, select the previously downloaded ISO image. Click on *Start restoration*....

A window asks *Do you really want to write the disk image to the device?* Check that the size and model of the device in question correspond to the size and model of our USB key. If so, click on *Restore*.

You will then be asked for your administration password. Type it in and *authenticate* to start writing the installation key. Once restoration is complete, click on  to eject the key.

16.4.2 Burn the installation image to CD or DVD

If you don't have a USB key or access to a GNU/Linux system, you can burn the installation image onto a CD or DVD.

The downloaded file is an "ISO image", i.e. a file format that most burning programs recognize as a "raw CD image". In general, if you insert a blank disc into your drive, the burning software will take care of transforming this image by writing it to the blank disc - at least, it works with Tails, and more generally under Debian or Ubuntu.

Under Windows, if you haven't already installed software capable of burning ISO images, the freeware InfraRecorder⁹ (will do the trick.

16.5 The installation itself

To install Encrypted Debian from the installation media (CD, DVD or USB stick), you need to boot from it, following the corresponding recipe.

page 107

From there, the actual installation can begin: allow yourself some time and a few crossword puzzles, as the computer will be able to work for a long time without any particular supervision.

In the case of a network installation image, check that the cable connecting the computer to the network is securely plugged in. And if it's a laptop, check that the power cable is plugged in, as there are no low battery notifications during installation.

The Debian installation program has its own documentation¹⁰. If you have any doubts about the steps described below, it may be worth taking a look. In addition, for most of the choices it asks us to make, the installation program will automatically suggest an answer that usually works...

At the time of writing, keys created in this way from Tails seem to work correctly.

9. <http://infrearecorder.org/>

10. The installation manual is available in several versions [<https://www.debian.org/releases/stable/installmanual.en.html>]. We will follow the one corresponding to the processor architecture [page 16].

16.5.1 Launching the installer

Boot from the installation media (CD, DVD or USB key). The *Debian GNU/Linux installer menu* appears. Press the

Enter (`↓`) or (`return`) to launch the rest of the installation program.

If you have chosen a multi-architecture CD, the option automatically selected by the installer will normally be *Graphical install*, and a *32-bit install* option will be available; in this case, the installer has detected that the processor is compatible with the `amd64` architecture, which offers a number of security advantages.

[page

16

16.5.2 Select language and keyboard layout

- After a little patience, a menu named *Select a language* appears: the installer offers to choose a language for the rest of the installation. Select *French*. To move on to the next step, select *Continue* each time.
- A menu asks for the country, to fine-tune the system's adaptation. Choose your geographic location.
- In *Configure keyboard*, the default choice *French* is appropriate if you have a French "azerty" keyboard.
- The installer then loads the files it needs.

16.5.3 Firmware and network hardware

After a load time, the Debian installation program will detect the network cards present in the computer.

As we saw earlier, some hardware requires the system to provide firmware in order to operate.

[page 120]

If the installation medium has been previously prepared with the firmware required for the system, a screen will appear asking you to accept a `SOFTWARE LICENSE AGREEMENT` or similar. After reading it, you can answer *Yes* to continue installation.

If the installation medium contains only free programs, you may see a message indicating a list of *missing firmware files*: these are non-free firmware programs that are useful for our computer, but which are not supplied by the installation medium. The installer suggests inserting a removable medium containing them. Selecting *No* will allow you to continue with the installation without installing these non-free firmware items.¹¹ Choosing *Yes*, on the other hand, instructs the installer to search for firmware files or packages containing such firmware on available devices - and thus to revert to the previous choice of a completely free installation.

[page 120]

¹¹. Missing microcode can be installed later after activating the repositories. non-free [page 136].



TO FIND OUT MORE...

You can prepare such a device by copying the most common firmware programs gathered by the Debian community [in an archive \[https://cdimage.debian.org/cdimage/unofficial/non-free/firmware/bullseye/current\]](https://cdimage.debian.org/cdimage/unofficial/non-free/firmware/bullseye/current) to be decompressed in a `firmware` directory at the root of a FAT-formatted USB key.

This *firmware* archive is available in three versions, corresponding to three different types of compression (`.cpio.gz`, `.tar.gz` or `.zip`). Depending on the format(s) our system is capable of decompressing, we'll choose the corresponding file. If you don't know the answer to this question, you can download all three files until you find one that you can decompress. As with the ISO image, it is advisable to check the integrity (see page 122) and authenticity (see page 122) of each downloaded file.

If the message appears again, the key does not contain the necessary¹². It is beyond the scope of this guide to indicate how to obtain all the firmware that may be useful. Finally, don't hesitate to answer *No...* In most cases, installation will continue without further problems, thanks to the wired connection, which dispenses with the firmware needed to operate the Wi-Fi card.

16.5.4 Network configuration and machine name

- The installer then takes a little time to configure the network. If your computer has several network cards, you'll need to choose the one you're going to use for installation. The default choice is generally the right one, if it's an *Ethernet* network card.
- We are then asked for the *Machine name*. Choose a small name for your computer, bearing in mind that this name will then be visible on the network, and may also appear in files created or modified with the system you're installing. It may therefore be a good idea to give it a generic name, like *debian*, for example.
- The installer then asks for a *Domain*. Without going into too much detail, it's best to leave this field empty (i.e. delete anything the program may have pre-filled).

16.5.5 Create users and choose passwords

The installation program now asks us to choose the *root password*. This is a password that would be needed to perform computer administration tasks: updates, software installation, major system modifications, *etc.*

However, it's simpler to save yourself an extra password, and allow the first account created on the system to have the right to perform administration operations¹³ by requesting the password again. To do this, it suffices not to enter a password for "root": simply leave the box empty and choose *Continue*, then again for *Password Confirmation*.

- In *Full name of new user*, choose the name associated with the first account created on the system. This name will often be recorded in documents created or modified in this session, so it may be useful to choose a new pseudonym.

12. For example, filenames beginning with `b43` are firmware for a particular type of Wi-Fi card, and are not redistributed directly by Debian. To make them work, you'll need to try installing one of the following packages once the system is up and running: `firmware-b43-installer`, `firmware-b43-lpphy-installer` or `firmware-b43legacy-installer`.

13. This mode is called *sudo*, because in the terminal it will be possible, by adding `sudo` to the beginning of the line, to execute a command as root, i.e. as superuser.

- In *Login for user account*, choose a *login* for this account. It is pre-filled, but can be modified. The installer warns you, in case you want to change it, that it must start with a lower-case letter and be followed by any number of numbers and lower-case letters.
- The installer asks for a password for the user. This is the person who will have the right to administer the computer, if you have decided not to enter a password.
"root" previously. (Don't forget to find a way to remember this password).

16.5.6 Partitioning disks

[page 20] If the installation medium has been booted in UEFI mode, the installer may ask: "*Force UEFI installation?*" This means that it has detected another system already installed on the hard disk, which uses the "BIOS compatibility mode" (the ancestor of UEFI) to boot. Since we're going to erase all traces of this old system anyway and put Debian in its place, we can answer *Yes* to this question.



TO FIND OUT MORE...

The likelihood of having a problem with UEFI is very low, but some motherboards or troublesome firmware may work better in BIOS compatibility mode.

If at the end of the installation the system does not boot into UEFI, you can restart the installation by answering *No* to this question, in order to install Debian in BIOS compatibility mode.

The installation support then starts the partitioning tool. It detects the partitions present, and proposes to modify them.

- In the *Partitioning method* menu, select *Assisted - use an entire disk with encrypted LVM*.
- In *Disk to partition*, select the disk on which to install Debian GNU/Linux. If you want to remove the currently installed system, this is usually the first disk on the list. The size of the disk is an indication of its suitability, so that you don't try to install Debian on the USB key containing the installer, for example.
- The installer then offers a choice of *partitioning schemes*. Select *All in one partition*.
- The installer then warns that it will apply the current partitioning scheme, which will be irreversible. Now that you've backed up what you want to keep, answer *Yes* to *Write changes to disks and configure LVM?*
- The installer will then replace the old disk contents with random data. This takes a very long time - several hours on a large disk - and leaves plenty of time to do other things!
- The installer then asks for a *secret passphrase*. Choose a good passphrase and type it in, then confirm the passphrase by typing it in a second time.
- The installer then proposes the size to be used on the disk in *Quantity of space on the volume group for assisted partitioning*. You can keep the default value, which corresponds to the maximum usable disk size.
- The installer shows a list of all the partitions it will create. You can trust it by leaving *Finish partitioning and applying the selected changes*.

- The installer warns that it will write changes to the disk. The entire disk has already been filled with random data, so if it contained important data it has already been erased. Answer *Yes* to *Should changes be applied to disks?* The installer then creates the partitions, which may take a little while.

16.5.7 Basic system installation

The installer will now install a minimal Debian GNU/Linux system. Let it do the rest...

16.5.8 Configuring the package management tool

Depending on the version of the installer used, it may ask different questions:

- If the installer asks *whether to scan installation media other than the one used to start the installer*, the default choice, *No*, is appropriate.
- If the installer asks *whether to use a mirror on the network*, the default choice, *No*, is appropriate. However, if you have a good Internet connection, you can also choose *Yes*: this will install an updated version.

If you're using a network installation (also known as "*netinst*", for *network install*), or if you answered *Yes* to the previous question, the installer will ask you from which server to download the :

- The installer first asks you to choose the *Country of the Debian archive mirror*. Select the country you are in.
- It then asks for *the Debian archive mirror* to be used. The default choice, *deb.debian.org*, is very good.
- The installer asks if an *HTTP proxy* is required. Leave blank.
- The installer then downloads the files it needs to continue.

16.5.9 Software selection

The next question concerns the *Configuration of popularity-contest* and asks *Would you like to participate in the statistical study of package usage?* Answer *No*, unless you agree to provide Debian with a list of the software you install.¹⁴

The installer then asks which *software to install*. It usually suggests the following: *Debian desktop environment*, *GNOME* and the *usual system utilities*.



TO FIND OUT MORE...

Most of the tools described in this guide are based on the GNOME desktop environment. However, GNOME is a little demanding in terms of power, and other lighter environments will be better suited to not very powerful computers: *LXDE*, *Xfce* or *MATE*.

¹⁴ Communicating our list of software installed in Debian facilitates the work of the people who develop and maintain this distribution, by giving them a vision of which software is most widely used. It also lets them know that this software is important to us and that we want it to continue to be maintained in Debian. However, the list of software we use is still personal data: if there are security breaches on Debian's servers, this data could be divulged. What's more, answering *No* to this question is also part of the construction of a collective political culture of refusal to communicate our personal data and opposition to governance by numbers.

The installer then downloads the rest of the Debian GNU/Linux system (or recovers it from the installation media) and installs it. It takes a long time, so there's plenty of time to do other things.

System services may need to be restarted when updated. If the installer ever suggests *Restart services automatically on update*, you can answer *Yes* to avoid the system asking for manual confirmation each time.

16.5.10 Installing the GRUB startup program

If you have chosen to install Debian in UEFI mode, the installer automatically installs the GRUB boot program, which allows you to start GNU/Linux.

Otherwise, the installer proposes to install GRUB on a part of the hard disk called the "boot sector":

- To the question *Install GRUB boot program on main disk*, answer *Yes*.
- The installer then asks for the *Device where the boot program will be installed*. Choose the internal hard disk, which will usually be `/dev/sda`. If in doubt, a good clue is to choose the first disk in the list whose name contains *ata* or *sata*.

When it has finished, the installer suggests restarting the computer, checking that the installation media (CD, DVD, USB key) is no longer inserted when restarting. Select *Continue*.

16.5.11 Reboot to the new system

The computer then boots on the new system. At one point, it asks for the passphrase on a black screen: "Please unlock disk". Type it in and press *Enter* (`↵`) or `return` at the end¹⁵.

After starting a number of programs, a screen appears with the words *debian 11* and the name of the account previously entered. Select the latter, then enter the associated password.



Here's a new Debian encrypted system ready for use. If you've never used one before, it might be a good idea to take a stroll through it to familiarize yourself. The Activities overview, which can be opened by clicking on *Activities* in the top left-hand corner of the screen, or by pressing the `Alt+F1` key (`⌘+F1` on a Mac), gives access to the many software packages already installed. To find a program, you can type in a word describing its function (e.g. *image* to find programs that work with images). To afficher all installed software, click on `Alt+F2` at bottom left. Help pages containing numerous tips and tricks can be accessed by typing *Help* in the activity overview.

16.6 Setting up Debian's main package repository

Once the installation is complete, depending on the image you used to install Debian, you may need to go to *Software & Updates* to add the main Debian package repository.

¹⁵ If you're not very comfortable with typing, you'll often make a typing error in the first few sentences. Don't worry about repeated mistakes, and keep at it until you manage to type the sentence without error... after a while, it will have "sunk in", and typing errors will become rarer. That said, it doesn't hurt to check that you've got it right.

has not inadvertently left the key `Alt+F2` pressed down, in which case we could go on and on long on the keyboard, but still unable to unlock the hard disk.

To do this afficher the activity overview by pressing  ( on a Mac), then type `software` and click on *Software & Updates*. In the *Debian Software* tab, select *Officially supported (main)*. Since this software modifies which programs we trust, we're reassured that it asks us for our password.

If you used a DVD image to install Debian, you also need to deactivate this repository so that your system no longer uses it. To do this, in the *Other Software* tab, uncheck all lines beginning with `cdrom:`. If you don't do this, Debian will insist that you always have the installation media inserted in your computer, so that you can update the list of available software.

To close the *Software & Updates* window, click on the *Close* button. It's possible that an *out-of-date software information* window will appear, in which case click on *Refresh*. A *Cache Refresh* window appears, showing the progress of the download of available package lists. This window and the *Software & Updates* window close automatically when the refresh is complete.

16.7 A few ideas to keep you going

It may now be useful to learn how to save data... and how to delete it. "for real".

page 151

page 139

It's also important to learn how to keep your system up to date. Software problems are discovered regularly, and it's important to install fixes as they become available.

page 175

16.8 Documentation on Debian and GNU/Linux

Here are some references to documentation on Debian and GNU/Linux :

- The Debian officiel reference guide ¹⁶ ;
- The Debian officielle documentation home page ¹⁷ ;
- Debian Administrator's Workbook ¹⁸.

There's a lot of documentation available on how to use GNU/Linux. If they are often very useful, they are, like many things on the Internet unfortunately, of uneven quality. In particular, many of them will stop working when a part of the system is modified, or have little regard for the privacy we expect from our system. So we need to think critically and try to understand them before we apply them.

Having said that, here are a few more wiki and forum references:

- The Debian officiel wiki ¹⁹ (partially translated from English) ;
- The French forum on Debian `debian-fr.org` ²⁰ ;

16. <https://www.debian.org/doc/manuals/debian-reference/index.fr.html>

17. <https://www.debian.org/doc/user-manuals.fr.html>

18. <https://debian-handbook.info/browse/fr-FR/stable/>

19. <https://wiki.debian.org/fr/FrontPage>

20. <https://www.debian-fr.org/>

Choosing, checking and installing software

This section offers a few recipes for managing your software :

- What are the criteria for choosing software? There are times when you need to choose software to perform a certain task, and you can get lost in the multitude of solutions available... In this chapter, we'll look at a few criteria to help you make the right decision.
- How do I find and install software with Debian? When you want to perform new tasks with your computer, you need to install new software. In this chapter, we'll give you some tips on how to find what you're looking for in Debian.
- How do I install packages on Debian? Sometimes you need *packages* to complement software or to serve their own purpose.
- How do I access Debian repositories? Software downloaded by the Debian system is stored in so-called *repositories*. While the repositories supplied with Debian contain almost all the software you may need, it is sometimes useful to add new ones.

[page 22]

[this page]

[page 134]

[page 135]

[page 136]

17.1 Selection criteria

C *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

🕒 *Duration: Half an hour to an hour.*

When it comes to choosing software to perform a certain task, it's easy to get lost in the multitude of solutions available. Here are a few criteria to help you make the right decision.

The benefits of using free software versus proprietary software or even

open source has already been explained. The remainder of this text will therefore focus solely - page 39 on the free software available.

17.1.1 Distribution

It's generally preferable to install software provided by your GNU/Linux distribution (e.g. Debian). There are two main reasons for this.

First of all, a practical question: the distribution provides the tools to install and update, in a more or less automated way, a set of software packages; it alerts us when one of the software packages we're using needs updating, for example to correct a security flaw. But as soon as you install software that is not supplied by your distribution, you have to think about updating it yourself, keeping abreast of any security flaws that are discovered in it, managing dependencies between software, *and so on*. It takes effort, time and skill.

On the other hand, a question of security policy: when you choose your GNU/Linux distribution, you have implicitly decided to place a certain amount of trust in a set of people, in a production process. Installing software not supplied by your distribution implies making a similar decision about a new set of people, a new production process. Such a decision is not to be taken lightly: when you decide to install software not supplied by your distribution, you broaden the set of people and processes you trust, and therefore increase the risks. For example, without a few precautions, you could quickly find yourself downloading a virus.

[page

61

17.1.2 Maturity

The lure of novelty is often a trap: software in full development may contain major problems that have not yet been discovered.

Whenever possible, it's best to choose software that has been actively developed and has reached a certain level of maturity. In software that has been developed and in use for at least a few years, chances are that the biggest problems have already been discovered and corrected... including security flaws.

To find out, you can consult the software's history. You can usually find them on their website by searching for terms like *historique*, *release*, *news* or *changelog*. If there are a lot of updates, especially recent ones, this means that the software is still being maintained.

17.1.3 Production processes and community

The *free software* label is an essentially legal criterion, which must never suffice to inspire confidence.

Of course, the fact that software is placed under a free license opens up the possibility of trust-inspiring development methods. But the people developing the software may well, intentionally or not, discourage cooperation and work in isolation. What does it matter if the program is *legally* free, if in fact no one else will ever read its source code?

We therefore need to take a quick look at the software production process, using the following questions to gauge the dynamism of the process:

- Who develops? One person, several people, a whole team?
- Is the number of people contributing source code increasing or decreasing?
- Is development active? We're not talking about pure speed here, but about reactivity, long-term follow-up and resilience. Software development is an endurance race, not a *sprint*.

And about the collective communication tools on which development relies (mailing lists and chat rooms, for example):

- Is there easy access to the discussions guiding software development?
- Do these discussions bring many people together?
- Do these people participate in its development, or do they simply use it?
- What's the atmosphere? Flat calm, dead silence, joyful cacophony, chilling seriousness, open arms, implicit hostility, tender complicity? (But also: sexist jokes, racist remarks?)
- Has the volume of discussions over the last few months/years been decreasing or increasing? More than the raw volume, it's the proportion of messages getting answers that's important: mature, stable, well-documented software won't necessarily be a source of discussion, but if no one's around to answer questions from newbies, that can be a bad sign.

[page

40

- Do you have any feedback or suggestions for improvement? If so, are they taken into account?
- Are the answers always given by a small number of people, or are there more widespread self-help practices?

17.1.4 Popularity

Popularity is a tricky criterion when it comes to software. The fact that the vast majority of desktops currently run on Windows in no way indicates that Windows is the best operating system available.

However, if this software isn't being used by many people, there are doubts about its long-term viability: if the development team were to stop working on it, what would become of it? Who would take up the torch?

A general rule of thumb is to choose software used by a sufficiently large number of people, but not necessarily the most widely used.

To measure the popularity of a software application, you can use the same criteria described above for the dynamism of the "community" formed around it. You can also look at the rating of an application in *Logiciels*, relying not only on the score but also on the number of people who have voted. For example, an application with a three-star rating and 295 votes will be preferred to one with five stars but only 19 votes. Debian also publishes the results of its popularity contest ¹This allows us to compare not only the number of people who have installed a given software, but also, and even more importantly, the evolution of its popularity over time.

17.1.5 Security past

You can also take a look at the security tracker ² offered by Debian. If you search for a program by name, you'll find a list of the security problems that have been discovered and, in some cases, resolved.

If this software has a perfectly clean security history, it could mean: either that nobody cares, or that the software is written in an extremely rigorous way.

If security flaws have been discovered in the software studied, there are several implications, some of them contradictory.

These vulnerabilities have been discovered and corrected:

- so they no longer exist, *a priori*;
- so one person was concerned with finding them, and another with correcting them: we can assume that attention is being given to this issue.

But these loopholes did exist:

- the software may have been written without security being a particular concern;
- other loopholes may remain undiscovered or, worse still, unpublished.

In order to affiner our intuition with regard to this software, it may be a good idea to look at the "time" criterion. For example, it's not dramatic if a few flaws were discovered early on in the development of a software product, and if none have been discovered for a few years, then we can put this down to errors.

1. [Debian.org, 2014, Debian Popularity Contest](http://popcon.debian.org/) [<http://popcon.debian.org/>].

2. Debian's security team maintains information for each package, which can be viewed on the *security tracker* [<https://security-tracker.debian.org/tracker/>], where you can search by software name.

of youth. On the contrary, if new flaws have been discovered regularly, for years and until very recently, it's quite possible that the software still has many security problems that are totally unknown... or unpublished. Just as a relatively high number of flaws, even recent ones, can indicate an active development community, and be a better sign than no security flaws at all for software that very few people actually deal with.

17.1.6 Development team

Who wrote the software? Who maintains it? Once we've managed to answer these questions, there are a number of clues that can help us determine how much trust we can place in the development team. For example:

- The same people have also written another piece of software that we already use extensively; our impressions of this other software are entirely relevant to this study.
- Members of the development team have addresses ending in `@debian.org` and therefore have the right to modify the software provided by Debian GNU/Linux; if we use this distribution, we already de facto trust these people.
- Members of the development team have addresses that end in `@google.com`, which shows that Google is paying them; while there's no doubt about their technical skills, it's questionable how much of their work is remote-controlled by their employer, who can't be trusted to know what they're up to when it comes to our personal data.

17.2 Find and install software

As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.



Duration: From five minutes (if you know the name of the software) to half an hour (if you're starting from scratch), plus download and installation time (from a few seconds to several hours, depending on the size of the software to be installed and the speed of your connection).

Sometimes, we already know the name of the software (also known as an *application*) we want to install - because it's been recommended to us, because we've found it on the Internet - and we want to know if it's in Debian. Other times, we just know the task we'd like the software to perform. In any case, the database of software available in Debian will certainly answer your questions.

To help you make the right choice when several software packages are available to perform the same task, see the chapter on software selection criteria.

page 131

- Open the *Software* application: afficher la vue d'ensemble des Activités en appuyant sur la touche (on a Mac), then type `logiciel` in the search bar and click on *Logiciels*.
- Then there are two techniques for finding an application:
 - or click on the icon in the top left-hand corner. Type the name of the application in the search bar. It's also possible to type in keywords, but application descriptions aren't always translated into French. With a basic knowledge of English, it's often a good idea to try out keywords in that language.
 - or select a category at the bottom of the page (e.g. *Games*).
- The search results are displayed. By clicking on a software icon, its description appears.

Once you've found the software you want, you can install it. You need to be connected to the Internet, as software is installed from packages that are downloaded online from so-called *repositories*.

page 23

- Click on the *Install* button below the software logo and title.

- Since we're going to install a new application, we're asked for our password.
- *Software* installs the new application.
- Exit the *Software* application.





17.3 Find and install a Debian package

- 🔄 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*
- 🕒 *Time: Ten minutes, plus download and installation time (from a few seconds to several hours, depending on the size of the packages to be installed and your connection speed).*

Packages are sometimes needed. Packages can be used to install software, but they can also be used to complement software or serve their own purpose.

To install *packages*, you can use the *Synaptic Package Manager* software.

17.3.1 Find a Debian package

- Open the *Package Manager*: afficher la vue d'ensemble des Activités en appuyant sur la touche  ( sur un Mac), puis taper `paquet` et cliquer sur *Gestionnaire de paquets Synaptic*.
- Since the package manager allows you to modify the software installed on your computer, and therefore to choose which programs you trust, it's reassuring that it asks for your password to open.
- Once in the package manager, let's start by reloading the list of available packages by clicking on the  *Reload* icon. The package manager then downloads the latest information on available packages from a Debian server.
- Then there are two techniques for searching for a package:
 - or click on the  *Search* icon on the right-hand side of the toolbar. Here, check that *Description and Name* is selected in *Search in*. Type keywords or an application name in the *Search* box (e.g. "German openoffice dictionary"). Descriptions of less current applications are rarely translated into French. With some basic English, it's often worth trying out keywords in that language;
 - either click on *Category* in the left-hand column, and choose the category that seems appropriate for the package.
- The search results or the packages in the category are then displayed in a list. By clicking on the name of a package, its description appears in the frame at the bottom. All that remains now is to install the corresponding package.

17.3.2 Select the package to install

For the actual installation of the package found above, there are different ways of doing things, depending on whether you want to use the default version, available in the official repositories of your distribution, or a package from another repository, for example to have a more recent version.

To install the default version

Normally, the desired package is now somewhere in the package list. Once you've found the corresponding line, right-click on it, and in the menu that appears choose *Select for installation*.

Sometimes, for the package to function properly, it is necessary to install other packages. For example, if several programs use the same package,

so that it is installed only once, it is not contained in each of the software packages, but exists separately and is referred to by the software packages. If the package to be installed depends on other packages, the manager opens a window asking whether it is necessary to *make any other changes?* Generally speaking, these suggestions are relevant, and you can accept them by clicking on *Add to selection*.

To install a specific version

[this page]

Sometimes, you'll want to install a particular version of a package from among those available, for example, if you've added specific repositories. Instead of choosing *Select for installation* from the context menu, select the desired package by left-clicking on its name, without clicking on the checkbox. Then go to the *Package* drop-down menu, and select *Force version....* Select the desired version. If this option is greyed out, it means that it is not available, as there is only one version of the package. The rest remains unchanged.


17.3.3 Apply changes

The last two steps can be repeated to install several packages at the same time. Once these installations have been prepared, all that remains is to launch them by clicking on *Apply* in the toolbar. The package manager then opens a *Summary* window, listing everything it's going to do. After a quick look to make sure you haven't made a mistake, click on *Apply*.

The package manager then downloads the packages from the Internet, checks them and installs them. Occasionally, the manager may indicate that some packages could not be verified: **this information is not to be taken lightly**. In such a case, it's best to cancel the download, click on *Reload* in the main menu, and repeat the package selection operation. If the message appears again, it may be the result of an attack, technical failure or configuration problem. But it's best to refrain from installing new packages until you've identified the source of the problem.

Finally, if all has gone well, the package manager displays a window stating that *the changes have been applied*, so you can click *Close*. Finally, close the package manager to prevent it falling into other hands.

17.4 Add deposits

 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

Duration: Quarter to half an hour.

The Debian packages containing the programs are located in so-called *repositories*. Although the repositories supplied with Debian contain almost all the software you could possibly need, it is sometimes useful :

- install software newer than that contained in the latest stable release of Debian, known as *backports*;
- install *non-free* software (e.g. firmware) or software provided by third parties (e.g. the Tor browser).



Warning: adding a new Debian repository to a computer means deciding to trust the people who maintain it. While the *backport* repositories we're talking about here are maintained by Debian members, this isn't the case for many other repositories. The decision to trust them should not be taken lightly: if the repository in question contains *malware*, it could be possible to install it on the computer without even realizing it.

[page 32]

17.4.1 Open Software & Updates

Open *Software & Updates*: to do this afficher the activity overview by pressing  ( on a Mac), then type `softw` and click on *Software & Updates*.

17.4.2 Disable local installation media

As mentioned in the previous chapter, depending on the installation image used to install Debian, the package management system may request that you always have this installation media plugged into your computer, so that you can update the list of available packages.

page 128

To avoid this, disable repositories for this installation medium: in the *Other Software* tab, uncheck all lines beginning with `cdrom:`.


17.4.3 Configure repository location

To install backported software

Click on the *Other Software* tab, then on the *Add* button.

In *APT line*, enter the APT directory to be added:

```
deb http:// deb.debian.org/ debian/ bullseye - backports main
```

In this case, the *repository version* is *bullseye-backports* and the *category* is *main*. Once this is done, click on  *Add an update source*.

Since this software modifies which programs we trust, we're reassured that it asks us for our password.

To install non-free or third-party software

- In the *Debian Software* tab, select according to your needs
 - *contrib* to add third-party software ;
 - *non-free* to add non-free software.

Since this software modifies which programs we trust, we're reassured that it asks us for our password.

17.4.4 Update available packages

To close the *Software & Updates* window, click on the *Close* button. It's possible that an *out-of-date software information* window will appear, in which case click on *Refresh*. A *Cache Refresh* window appears, showing the progress of the download of available package lists. This window and the *Software & Updates* window close automatically when the refresh is complete.

To install a package from the backports, follow the *install a package* tool and choose to install a particular version when the question arises.

page 135

Deleting data "for real"

We saw in the Understanding section that when you delete a file, its contents on page 42 are not really deleted. However, there are programs that allow you to delete files *and their contents*, or at least attempt to do so, with the limits explained earlier.

18.1 A little theory

18.1.1 The Gutmann method

The documentation for the *secure-delete* package, which we'll use in the next recipe, inspired by a publication by Peter Gutmann published in 1996¹ tells us:

The deletion process works as follows:

1. *the overwriting procedure (in secure mode) replaces the contents of the file [...]. After each pass, the disk cache is emptied;*
2. *the file is truncated, so that an attacker does not know which blocks on the disk belonged to the ;*
3. *the file is renamed, so that an attacker cannot draw any conclusions about the contents of the deleted file from its name;*
4. *finally, the file is deleted. [...]*²

For a magnetic hard disk less than 20 years old³: it suffices to overwrite the data a few times with random data.

The NIST (*National Institute of Standards and Technology*, the U.S. government body that defines the security protocols used, among others, by that country's administrations) has published a 2006 study by the NSA, which seems to conclude that on recent magnetic hard drives, data is so tightly bound together that it becomes virtually impossible to perform magnetic analysis to find traces of erased data.⁴ by the NSA, which seems to conclude that on recent magnetic hard drives, data is so tightly glued together that it becomes virtually impossible to perform magnetic analysis to find traces of deleted data.

Consequently, in the recipes that follow, we'll confine ourselves to a few random rewrites.

However, this method is not suitable for SSD disks. Nowadays, SSD disks are tending to replace hard disks...

Once again, it's a question of finding the right compromise, on a case-by-case basis, between speed page 65

1. Peter Gutmann, 1996, *Secure Deletion of Data from Magnetic and Solid-State Memory* [http://www.cs.auckland.ac.nz/~pgut001/pubs/secure_del.html].

2. Source : file README.gz file from *secure-delete* installed on a Debian at /usr/share/doc/secure-delete.

3. Using PRML technology [<https://fr.wikipedia.org/wiki/PRML>], introduced in 1990 [<http://www.datadoctor.biz/datarecoverybook/chapter-2.html>] (in English).

4. NIST, 2006, *Guidelines for Media Sanitization*
[<https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-88.pdf>].

and the desired level of protection, depending on the size of the data to be overwritten, the age of the hard disk, and the trust placed in NIST.

18.1.2 For USB sticks, SSD disks and other *flash* memories

For USB sticks and other *flash* memory devices - such as SD cards or SSD disks - a study dating from 2011 ⁵ showed that the situation was really problematic.

This study shows that it is impossible, no matter how many times a file is rewritten, to be sure that all its contents have been overwritten. Even if this makes the data inaccessible by simply plugging in the key, it is still visible to anyone looking directly into the *flash* memory chips.

The only method that worked consistently was to rewrite *the entire* USB stick several times. In most cases, two passes suffi, but on some models, twenty rewrites were needed before the data disappeared for good.

[page 145] Based on these observations, the preventive response seems to be to systematically encrypt USB sticks, an operation that makes it really difficult to extract information directly from *flash* memory chips. And when it comes to cleaning up after the event, despite its limitations, full overwrite still protects against purely software attacks.

The only way to make the data on these media unreadable is to physically destroy them.

18.1.3 Other limits to "secure" deletion

There may still be information about the file that can be used to retrieve it, especially if you use a journaled file system such as *ext4*, Btrfs, HFS+, ReFS, NTFS, a write system, compression or backup (whether on disk, for example with RAID ⁶ or *via* a network). See Part 1.



[page 43] 18.2 On other systems

We've seen that it's illusory, if you're using a proprietary operating system, to seek real privacy. Although there is software that supposedly deletes files with their contents on Windows and macOS, it's therefore far more difficult to trust them.

[page 39] 18.3 Let's go

Contents can be deleted:


- of individual files (see next page) ;
- of an entire peripheral (see opposite page);
- of previously deleted files (see page 143).

5. Michael Wei *et al*, 2011, *Reliably Erasing Data From Flash-Based Solid State Drives* [[http s://www.usenix.org/legacy/events/fast11/tech/full_papers/Wei.pdf](http://www.usenix.org/legacy/events/fast11/tech/full_papers/Wei.pdf)].


6. RAID stands for *Redundant Array of Independent Disks*. It's a system that distributes data over several disks in order to improve either performance, security or fault tolerance (Wikipedia, 2021, *RAID (computing)*) [[https://fr.wikipedia.org/wiki/RAID_\(computing\)](https://fr.wikipedia.org/wiki/RAID_(computing))]

18.4 Deleting files... and their contents

 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

 *Duration: Five minutes of preparation, then a few seconds to several hours of waiting, depending on the size of the file to be deleted and the method used.*

Here's how to get rid of your files, taking care to render their contents unreadable.

 **Please note:** this method only works with mechanical hard drives. After overwriting the contents of files on a USB stick (or any other storage medium using *flash* memory, such as an SD card or SSD disk), there's a good chance that they're still written to an inaccessible region of the device!

18.4.1 Install the necessary software

If you haven't already done so, install the `nautilus-wipe` package (see page 135), then restart your computer.

This package is present in Tails by default.

18.4.2 Deleting files and their contents from the file browser

In Tails

To delete files and their contents using Tails, consult the documentation by clicking on *the Tails Documentation* icon on the desktop. In the index that opens, look for the *Encryption and privacy* section and click on the *Secure file deletion and disk space cleanup* page.

With an encrypted Debian

To delete files and their contents from the File Browser, navigate to the file, right-click on it and select *Overwrite*. A window opens, asking you to confirm the deletion, and also proposing a few *Options*.

We can choose the number of passes to be made in order to cover the data on our device, as well as some behavior options when erasing the data. The default options are sufficient for magnetic hard disks.

Then click on *Overwrite*. When deletion is complete, an *Overwrite Successful* window opens, stating that *the element(s) have been successfully overwritten*.

18.5 Deleting an entire disk "for real"

Before disposing of a hard disk, recycling it, reinstalling a clean page 71 system, or sending a broken computer to the after-sales service department, it may be wise to put obstacles in the way of people who want to recover the data it contained. The best solution is to rewrite the entire disk with random data.

Before using this recipe, think twice and carefully back up the data to be stored. If applied correctly, it makes the data very difficult to recover, even when analyzing the disk in a laboratory. page 151

18.6 Delete the entire contents of a disc

🔄 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

🕒 *Time: Five minutes' preparation, then several hours' waiting depending on disc size.*

next
page.

To erase an entire volume (disk or partition), use the `shred` command to overwrite the entire data set three times with random data. In addition to deleting files, this command covers the deleted space in such a way as to make it virtually impossible to find what was there before.

page 113

To cover the contents of a disk, you need to be away from it... if it contains the operating system you normally use, you need to put the hard disk in another computer or use a *live* system. `shred` is a standard tool, so any *live* system should do the trick.

The command is very simple. All you need to know is the location of the device (its path) you want to delete, and then be patient, as the process takes several hours.


18.6.1 Find device path

First and foremost, you need to be able to identify the path used by the operating system to designate the storage medium you wish to erase.

If you wish to erase an internal drive, first disconnect any external hard drives, USB sticks, memory card readers or other storage devices connected to the computer. On the one hand, this will prevent them from being erased by mistake; on the other, it will make it easier to find the internal drive.

Of course, you shouldn't do this if you want to make the contents of an external drive inaccessible.

Open the disk management utility

Open Disks: afficher la vue d'ensemble des Activités en appuyant sur la touche  (⌘ sur un Mac), puis taper `disque` et cliquer sur *Disques*.

Find device path

The section on the left shows the list of disks known to the system. You can click on any of them to see more information on the right-hand side. The icons, size and names of the disks should help you identify the one you're looking for.

If this doesn't suffice, you can take a look at the partition organization, by looking at the table that appears on the right-hand side:

- if the disk to be erased contained an unencrypted GNU/Linux system, there must be at least two partitions, one with a *Swap* file system, the other usually *Ext3* or *Ext4* ;
- if the disk to be erased contained an encrypted GNU/Linux system, there must be at least two partitions, one with an *Ext2* file system, the other *LUKS* ;
- if the disk to be erased contained a Windows system, there must be one or more partitions marked *NTFS* or *FAT*.

Furthermore, the device corresponding to the internal disk is usually the first on the list.

Once the disk has been found and selected, you can read the disk path in the bottom right-hand corner, next to the *Device* label.

The device path begins with `/dev/` followed by three letters and possibly a number, the first characters in most cases being `sd`, `hd` or `mmcblk`: for example, `/dev/sdx1`. Make a note of the path somewhere, without the number (e.g. `/dev/sdx`): you'll need to write it down later, instead of LE-PÉRIPHÉRIQUE.



Please note: this path may not always be the same. It's best to repeat this short procedure after restarting the computer, plugging in or unplugging a USB key or hard disk. This will avoid unpleasant surprises... such as losing the contents of another hard disk.



18.6.2 Run the shred command

Open a Terminal: open the activity overview by pressing [page 97](#)  ( on a Mac), then type `term` and click on *Terminal*.

Enter the following command, replacing THE-DEVICE with the device path determined above:

```
> pkexec shred -n 3 -v LE-PÉRIPHÉRIQUE
```

If you prefer to use Gutmann's original method (more time-consuming, and perhaps safer), replace `-n 3` with `-n 25` in the command line.

Once the command has been typed and checked, press the *Enter* key ( or ). A password is requested, as this command requires [page 99](#) administration privileges, enter it. The `shred` command will then write `-----` to the terminal, which it does (since it has been asked to do so by adding the `-v` option to the `shred` command, which means, in the context of *this* command, that the computer must be "verbose" - i.e. "talkative"):

```
shred: / dev/ sdb : pass 1/3 (
random)... shred: / dev/ sdb : pass
2/3 ( random)... shred: / dev/ sdb :
pass 3/3 ( random)...
```

At the end of the procedure, the terminal again displays the `$` sign, symbolizing the command prompt. The terminal can now be closed.

18.6.3 Reuse the disc

Warning: this method not only deletes the data of an entire volume, but also,

At the end of the operation, the disk no longer has a partition table or file system, [page 23](#)
To reuse it, you need to create at least one new partition `-----` and its file system, using the
Disks application for example. [page 24](#)

⋮

18.7 Make previously deleted data irretrievable



As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.



Time: Five minutes of preparation, then several minutes to several hours of waiting, depending on the size of the disc to be cleaned and the method used.

When files *have already* been deleted without any special precautions, the data they contained is still on the disk. The aim of this recipe is to recover any remaining data, by overwriting the free space on a hard disk. This method does not delete any files visible in the file browser.



Warning: like the other ways of deleting a file "for real", this doesn't work with certain "intelligent" file systems which, to be more efficient, won't show all the free space to the software responsible for overwriting the traces there. Nor should you trust this method for USB sticks, SD cards or SSD disks, preferring to cover the entirety of the data they contain several times over.

 _ page 43
 page 142

----- **In Tails**

The `nautilus-wipe` package is already installed in Tails by default. We therefore suffice to consult the documentation, by clicking on *the Tails Documentation* icon on the desktop. Then, in the index that opens, look for the *Encryption and Privacy* section and click on the *Securely delete files and clean up disk space* page.

With an encrypted Debian

If you haven't already done so, install the `nautilus-wipe` package (see page 135), then restart your computer.

Next, open a file browser and navigate to the disk you wish to clean. Right-click in the right-hand part of the file browser and select *Overwrite free disk space*. A window opens, asking us to confirm the deletion of available disk space, and also proposing a few *Options*.

We can choose the number of passes made in order to recover data from our device, as well as some behavior options when erasing data. The default options are sufficient for today's magnetic disks.

Then click on *Overwrite available disk space*. Overwriting may take some time. In some cases, the administration password is required.

Note that a file called `tmp.XXXXXXXXXX` is created inside the folder. Nautilus Wipe will create this file inside and increase its size as much as possible, in order to use all available free space, then overwrite it securely. Once deletion has been completed, a *Successful Overwrite* window opens, stating that

The available disk space on partition or device "... " has been successfully overwritten.

Partitioning and encrypting a hard disk

We're now going to look at the encryption of a device, in order to store data on it confidentially.

Once a disk has been encrypted, the data it contains can only be accessed once a passphrase has been entered to decrypt it. For further information

For more information, see the section on cryptography.

page 47

When the passphrase is entered, the system has access to the data of the device in question, so don't type the passphrase just anywhere, but only-

ment on computers and systems in which one has suffisamment confidence.

page 65

Not only will they have access to the decrypted data, but page 27 traces of the device's presence will also be kept on the computer. For this reason, we recommend- that you use it on an encrypted GNU/Linux system or on an encrypted PC.

page 119

an amnesiac live system.

page 113

It could be a hard disk, an SSD, a USB stick, an SD card, or even just part of one of these devices. In fact, a hard disk or USB stick can be cut into several independent pieces, known as

scores.

page 23

In the following, unless otherwise specified, we'll use the term disk to refer to both internal and external hard disks, or any type of *flash* memory device, such as a USB stick, SSD drive or SD card.

If you want to have a place on the disk to put data that will not be confidential, and that can be accessed on untrustworthy computers, you can split the disk into two partitions:

1. an unencrypted partition, where only non-confidential data is stored, such as music, and which can be used from any computer without typing the passphrase;
2. an encrypted partition containing confidential data, to be opened only on trusted computers.

19.1 Overview

19.1.1 Encrypting a disk with LUKS and dm-crypt

We will explain how to encrypt a disk using the standard methods under GNU/-.

Linux, called dm-crypt and LUKS, which are open-source software. This system is well integrated with desktop environments, so most operations are possible without the need for special tools.

19.1.2 Other software we don't recommend

page 39 To encrypt a disk, we advise against using proprietary software that cannot be trusted, such as FileVault, BitLocker, Stormshield Endpoint Security or Symantec PGP Whole Disk Encryption. There are also freeware products, such as page 22 VeraCrypt [<https://www.veracrypt.fr/>], which can run on proprietary operating systems. However, if you use software, even free software, on a proprietary operating system, you are implicitly trusting the latter, since it inevitably has access to the decrypted data.

19.1.3 Stage overview

If the disk has already been used, it may be a good idea to start by recovering the data (see page 141).

If the disk to be encrypted has no free space, start by formatting it (see this page). This may involve deleting all data on the disk.

Then, if you wish to encrypt only part of the disk, you must first create an unencrypted partition (see opposite page).


If you already have unpartitioned space on your disk, you can proceed directly to the encryption stage (see page 148).

All that remains is to initialize it to contain encrypted data (see page 148).

And now it's ready to use (see page 148).

19.2 Preparing a disk for encryption

 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

 *Duration: Approximately ten minutes.*

In the following, we'll always use the term disk to refer to an internal or external drive, as well as a USB stick, SD card or SSD disk, unless we specify otherwise.

The procedure explained here involves deleting all the data on the disk.¹ If you already have unpartitioned space on your disk, you can proceed directly to the encryption stage (see page 148).

19.2.1 Install the necessary packages

To encrypt a disk, we need the following packages: `secure-delete`, `dostfstools` and `cryptsetup`. With Debian 11, you need to install the `secure-delete` package (see page 135), the other two being installed by default. If you're using Tails, these three packages are already installed.

19.2.1 Format the disk with the Disks utility

Formatting means erasing all the data on the disc.

To open the Disks application from the Activities overview: press the  key ( on a Mac), then type `disks` and click on `Disks`.

In the Disks application window, the left-hand side lists the disks known to the system; the right-hand side allows you to perform actions.

1. However, it is possible to *resize* an existing partition while keeping the files on it.


Select device

On the left is the list of disks. If the computer in use contains an encrypted system, the encrypted volumes of our system are also shown.

The icons, the size indicated and the names of the disks should enable us to identify the one we're looking for.

Once the disc has been located, select it from the list. The information displayed on the right of the window should confirm that you have selected the correct disc.


Dismantling volumes

If the volume is mounted, a square icon  is visible in the right-hand section, below the graphic representation of the disk in the *Volumes* section. Click on this button to unmount the volume.

If this disk contains several volumes, unmount them one by one: select them in the graphical representation in the *Volumes* section, then unmount them as previously explained.

Reformat disc

Warning: formatting a disk means deleting all the files on it.

In the software's top bar, click on the  icon, then on *Format disk...*


A window opens, offering the choice of deleting or not deleting the data on the media, and of format the disk. Depending on the context, and the limits discussed above, on [page 42](#), choose whether or not to erase the data. Leave *Compatible with all systems and devices* in *Partitioning*, then click on the *Format...* button.

Disks asks if you really want to format the device. Now's the time to check that you've chosen the right device before making a mistake. If so, confirm by clicking on *Format*.

Formatting may take some time, and a progress bar will appear in the Disks application. Wait until the task has been completed before unmounting or unplugging the drive.


19.3 Create an unencrypted partition

 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

 *Duration: Two minutes.*

If you wish, this is the time to create an unencrypted partition where you can store data that is not confidential, and which you can use from any computer without having to type in a passphrase.

If you wish to encrypt the entire disk, you can proceed directly to the next step (see next page).

Still in the Disks application, select the desired disk, then on the right-hand side, click on the *Available space* area of the *Volumes* diagram. Underneath, click on the  symbol.

Select the desired size for the unencrypted partition in the dedicated field. The space left free will be used for the encrypted partition. Click on *Next*.

You can choose a name for this partition. In *Type*, select *Compatible with all systems and devices (FAT)*. Once this has been done, click on *Create*.

19.4 Creating an encrypted partition

- 🔄 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*
- 🕒 *Duration: Ten minutes + between a few minutes and several hours to fill the free space, depending on the size of the partition.*

19.4.1 Create encrypted partition

Still in Disks, with the target disk selected, right-click on the *Available space* area of the *Volumes* diagram. Then click on the **+** symbol below.

Choose the partition size: keep the maximum size, since we want to create a single encrypted partition in this available space. Click on *Next*.

A name can be given to the future encrypted partition. It is not necessary to activate the *Erase* option. Erasure will take place in the next step, via random data filling, which will be more reliable. In the *Type* section, select *Internal disk for use with Linux systems only (Ext4)*, then tick *Password-protected volume (LUKS)*. Click on *Next*. Choose a suitable passphrase (see page 103) for the encrypted volume and type it into the two appropriate fields. Finally, click on *Create*.

19.4.2 Fill the partition with random data

Finally, we fill the empty space on the encrypted disk with random data. This hides the location of our data, making it more difficult for anyone to decrypt it.

On the *Volumes* diagram, locate *Partition [...] LUKS* and select the *File System* below it. Under the diagram, click on ▶

At the bottom of the window, in *Contents*, a link appears after *Mounted on*. Click on this link to open the folder, then follow the tool to make previously deleted data irretrievable (see page 143).

The process takes from a few minutes to a few hours, depending on the size and speed of the drive (for example, two hours for a 4 GB USB key).

19.4.3 Cleanly unplug the disc

In the file browser, click on the **▲** symbol, then physically disconnect the disk (if applicable).

The encrypted disk is now usable.

19.5 Use an encrypted hard disk

- 🔄 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*
- 🕒 *Duration: Two minutes, a few hours... or never, if the passphrase escapes us.*

To enable the system to access data on an encrypted disk, you need to specify a passphrase (which is just what we wanted!). But this operation is more or less straightforward, depending on the environment.

19.5.1 With Debian (or other GNU/Linux)

On a GNU/Linux system with a desktop environment configured to open external media automatically, a window appears asking for the passphrase when an external disk containing encrypted data is plugged in.

If this is not the case, this window appears when you ask the system to open the encrypted partition, for example from *Files* by clicking on the disk name in the left-hand column.

To close the encrypted partition, you suffice to unmount the disk as you normally would.

19.5.2 With other systems

We know of no simple way of accessing the encrypted disk partition under Windows or macOS. Even if solutions do exist² it's a good idea to trust that these are proprietary operating systems, in which there's no reason to trust. page
39 no reason to trust.

If you want to put data on the disk that you want to access on computers you don't trust, then the best thing to do is probably to provide a second, unencrypted partition on the disk, as explained above. page 147

² For older versions of Windows (up to Vista), it was possible to use FreeOTFE (<https://sourceforge.net/projects/freetofe.mirror/>).

Saving data

Backing up is a relatively simple operation in principle: make a copy of the files you don't want to lose, on a storage medium other than the one where the data is located.

Of course, if we take the trouble to put our work data on encrypted hard drives or USB sticks, these copies need to be encrypted too.

Two other points to bear in mind when implementing a good *backup policy*:

- define a method for **regular** backups,
- check from time to time whether backups are still readable.

This last aspect should not be overlooked. Losing original data is often painful. And then to find that the backups can't *restore* what you've lost turns the situation into a catastrophe.

It's also a good idea to store back-ups in a different location from the original data, to avoid everything being destroyed at the same time (fire, water damage...).

20.1 Special case of Tails persistent storage

When using Tails, there is a method for backing up the entire persistent volume of a Tails key.

To do this, we'll follow the official Tails documentation, which is available from any Tails USB stick or DVD, even without an Internet connection.

[page 115]

Start Tails. On the desktop, click on the *Tails Documentation* icon. Look for the section *Getting started with Tails* and in the section *Encrypted persistent storage* click on *Create a backup of your persistent storage* and follow this documentation page.

20.2 With file manager and encrypted storage

Making backups is above all a question of rigor and discipline. In simple cases, you can dispense with software specifically designed for making backups, and simply make copies to an encrypted storage medium using your file manager.

20.2.1 Backing up

- 🔄 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*
- 🕒 *Duration: for the first time, time to encrypt the storage medium and decide which files to back up; after that, it depends on the amount of data to be backed up.*

page 145

The encryption of our backups is ensured by the encryption of the external storage medium, USB key or hard disk.

To make copies regularly and without spending too much time on them, we recommend :

- to have somewhere a list of files and folders to back up;
- make yourself a little calendar of the days or weeks when you'll be making your saves, with boxes that you check off after you've made them.

A good practice is to create a folder (on the backup storage medium) with the date of the backup and copy the data into it. This makes it easy to keep several backups if desired, and to delete previous backups just as easily.



PRECISION

When choosing which files to back up, bear in mind the data in certain programs (such as those in the Thunderbird e-mail program¹), which are sometimes in hidden folders. In *Files*, they can be affichés by clicking on then *Afficher les fichiers cachés*.

20.2.2 Restore a backup

- 🔄 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*
- 🕒 *Duration: depends on the amount of data to be restored.*

In the event of loss of the original data, restoration is just as simple as backup: by making copies in the opposite direction.

20.2.3 Make sure backups are always readable

- 🔄 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*
- 🕒 *Time: Approx. five minutes, then wait for verification.*

If you've backed up your data on an external storage device, you'll need to connect it to your computer first.


Perhaps the most obvious way of ensuring that backups are always readable is to simulate a restore. However, there is a drawback: you need to have enough free space at your disposal to copy all the backed-up data to a temporary folder, which you then delete.

page 97


Here's another method, perhaps less easy to implement, but which doesn't have this constraint. It requires the use of a Terminal.



1. Vincent, Goofy *et al*, 2021, *Profiles - where Thunderbird keeps your messages and other user data* [<https://support.mozilla.org/fr/kb/profils-thunderbird-conserve-donnees-utilisateur>].

Start the command by typing (**without** pressing *Enter*,  or ):

```
 tar -cPf / dev/ null
```

Next, add a space and indicate the folder containing the backups, by attracting the folder icon with the mouse and bringing it to the terminal window. After releasing the button, what's affiché should look like this:

```
 tar -cPf / dev/ null '/ media/ external/ backups
```



Playback starts as soon as you press *Enter* ( or ). The following line should remain empty until the end of the operation.

If error messages appear in the meantime, such as "*Input/output error*" or "*Erreur d'entrée/- sortie*", this means that the backup is corrupt. In this case, you need to make a new backup on a new storage medium (USB key or hard disk), check it and then dispose of the faulty storage medium.

After a little patience and the return of the command prompt \$, you can close the terminal.

Note: these two methods share the shortcoming of not verifying data integrity. page 53
Setting up a mechanism to do this is difficult without resorting to more complex backup software.

20.3 Using Déjà Dup

-  *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*
-  *Time: Five minutes to install the software.*



Alternatively, you may prefer to use specialized backup software. One such program, Déjà Dup, is easy to use and produces encrypted backups. These backups are also "in-cremental", i.e. only new files and modifications are saved; files unchanged since the previous backup are not copied again, so it is possible to access files as they were at each backup.

What makes it so simple may be a limitation: when you configure the software, you choose the folders to be backed up and the medium on which to store them. But you can't have multiple configurations that would allow you to save certain folders on a hard disk with one passphrase, and other data on a server, for example, with another passphrase. Déjà Dup is therefore ideal for backing up the contents of your personal folder on a regular basis, but not much more.

In addition, it is not supplied with the default environment, so you need to install the software (see page 134) *Déjà Dup Backups* before you can use it.

20.3.1 Making a backup

- 🔄 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*
- 🕒 *Time: about 15 minutes for setup, from a few minutes to several hours for backup, depending on the size of the file to be copied.*

Open *Backups* from the Activities overview: press  ( on a Mac), then type *save* and click on *Backups*.


The first time you launch the program, you'll be greeted by two buttons, one for *Create my first backup*, the other for *Restore from a previous backup*. Click on the first button to define what you want to backup and where. A *Backup* window appears, showing several steps:



1. Leave *Folder to be backed up* set to *Personal folder* with account name, which is sufficient in most cases. Add folders to ignore containing files that are often large but easier to find, such as *Videos* or *Music*, to *Folders to ignore*. Then select *Next*.
2. In *Storage location*, select the backup location. To store the backup on an external drive, connect the drive in question to the computer, then select it from the *Storage location* list. Choose a name for the backup folder in *Folder*. Then select *Next*.
3. The *Protect backup with password* choice is selected by default: we then enter a passphrase (see page 103) in *Encryption password* to encrypt² our new backup. Note that encryption only concerns the actual content of the files to be backed up: Dup does not encrypt the names of the files and directories being backed up. What's more, the passphrase cannot be modified once it has been defined. Click on *Next* to start the backup.

Once the backup is complete, the *Backup* window closes, giving way to the *Backup Overview*. This displays a notification message about the date of the last backup and the next scheduled backup.

Backup automation can be activated via the *Save automatically* button, which turns blue when enabled.

By default, automation is weekly and the backup retention period is permanent.

All these parameters can be modified via the *Preferences*, accessible from the  menu:


- The backup location can be changed in the *General* tab.
- Backup retention time can be limited to three months, six months, one year or indefinitely from the *General* tab.
- Backup automation can be activated from the *General* tab, and you can choose the *frequency of automatic backups* between *Every day* and *Every week*.
- *Folders to be backed up* are listed in the *Folders* tab: the  button adds an additional folder to be backed up; the  button removes the corresponding folder from the backup.
- *Folders to be ignored* are also listed in the *Folders* tab and are added and deleted in the same way.

2. If the external medium is encrypted, you may decide not to encrypt the backed-up files. This means one less passphrase to invent and remember. However, you lose the ability to compartmentalize access, should the external media be used for purposes other than backups.

When scheduled backups are activated and the indicated time since the previous backup has elapsed, Déjà Dup affichets a notification message to let us know that the scheduled backup is delayed and will start as soon as the external media is plugged back into the computer. And as soon as this happens, a window will automatically open asking you to enter the *passphrase* needed to update the backup.

20.3.2 Restore a backup

 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

 *Duration: Five minutes for configuration, from a few minutes to several hours for restoration, depending on the size of our backup.*

Open *Backups* from the Activities overview: press  ( on a Mac), then type *save* and click on *Backups*.

Connect the disk containing the backups, and open it from *Files* if it is encrypted.

The restore operation is started by clicking on the *Restore* button.

If this is the first time you've used *Backup* (for example, to restore your personal folder after the loss of a hard disk), you'll be asked to specify the folder where the backups were made. Otherwise, it uses the backup parameters already configured.

After a short delay, *Backups* affdisplays the list of files and directories from the last backup, along with its date. Another backup date can be chosen from the *Date* drop-down list. The *Restore* button can be used as soon as all or some of the directories and files to be recovered have been selected.

Next, you need to specify the folder where the files from the backup will be written. You can either *Restore files to their original locations* (which may replace some files with their old version from the backup), or *Restore to a specific folder*.

After clicking on *Restore*, writing of the files from the backup begins in earnest, after asking for the passphrase if the backup was encrypted. If all goes well, the window affiche *Your files have been restored successfully*.

20.3.3 Make sure backups are always readable

 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

 *Duration: From a few minutes to several hours, depending on the size of our backups.*

Déjà Dup's incremental operation superficially ensures that previous saves are readable. However, this is no guarantee.

Unfortunately, the best method currently available with Déjà Dup to ensure that you can restore your backups is... to restore to a temporary folder that you'll delete afterwards. This is far from practical, and you need access to a suffisufficiently large encrypted hard disk.

However, you can ensure that files containing backups remain readable by using the same methods as those described above.

Sharing a secret

C *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

🕒 *Duration: Approximately one hour.*

Sometimes you want several people to share a secret, without each person having the whole secret.

A number of cryptographic techniques have been invented for this purpose. Using slightly different mathematical calculations, they can all be used to decode a secret into several pieces, which can then be reconstructed by putting a few of them together. ¹.

21.1 Share a passphrase

The most practical use is to share the passphrase of an encrypted medium as a secret. page 145

Ideally, this step should be carried out using a *live system*, so as not to leave any traces of the secret you're about to share. page 113

21.1.1 Install the necessary package

To share the secret, use the `ssss-split` program. This program is one of those supplied with the Tails *live* system. However, to use it on an encrypted Debian, you need to install the Debian `ssss` package. page 135

The tools contained in the `ssss` package are to be used on the command line. All operations will therefore have to be carried out in a Terminal, without the powers of administration. page 97

21.1.2 Generate a random passphrase

In our case, no one should be able to remember or guess the passphrase that will be used for encryption. So we'll generate a completely random passphrase by typing the command :

```
➤ head -c 32 / dev/ random | base64
```

The computer will reply something like :

```
7 rZw00u+8 v1stea980uyU1efwNzHaKX9CuZ/ TK0bRWY=
```

1. For more details, see the Wikipedia article on [distributed secrets](https://fr.wikipedia.org/wiki/Secret_r%C3%A9parti) [https://fr.wikipedia.org/wiki/Secret_r%C3%A9parti].

If you wish to vary the length of the passphrase, replace 32 with the desired number of characters. Select this line with the mouse and copy it to the clipboard, by right-clicking and then clicking on *Copy*.

21.1.3 Cutting out the secret

Before cutting the secret, we need to decide how many pieces it will be cut into, and how many pieces will be needed to reconstitute it.

Next, still using our terminal, we need to use `ssss-split` as follows:

```

> ssss-split -t NUMBER-OF-MORRORS-NECESSARY -n
S NUMBER-OF-TOTALS

```

The `NUMBER-OF-PASSPHRASES-NECESSARY` is the number of pieces that need to be assembled to find the original passphrase. The `TOTAL-NUMBER OF PARTS` is the number of pieces into which the passphrase will be cut. The `WARNING: couldn't get memory lock` message can easily be ignored if you're using a *live* system.

When it asks for the secret, you can paste the contents of the clipboard, by right-clicking and then clicking on *Paste*. Then press *Enter* (`↵`) or `return`) to validate the order.

Each person sharing the secret will have to keep one of the lines affichés next. And that in their **entirety**, also taking careful note of the first number followed by the dash.

Here's an example with the random key generated earlier, shared between six persons and requiring three of them to get together to find it:

```

$ ssss - split -t 3 -n 6
Generating shares using a (3 ,6) scheme with dynamic security
Enter the secret , at most 128 ASCII characters: Using a 352 bit
S security
1- b8d 576 a1a8091760 b18f125 e12bb6f2 b1f 2 dd9d93f 7072 ec 69
2- 129b2785b0797506e 7b 4399
3- af83f 0af 05 fc 207 e3b 466 caef30ec4 d39c 060800371 feab93 594350
4- 7699598db0c 71 ed9cd 2
5- 471873b 58873 dab 22d24e 526931 b 061 a6 ac331613 d8fe 79b 2172213
6- a 76729aa627d ec0e6cf 77b6
7- 143 a1efcde 7f4f 5658415 a150 fcac 6da 04f 697 ebfef9427 b59 dca
8- 7b50 10e735610 594 e6
9- fca1250 b5 cbec 40 ab14964 d2cd 7463 af34 c389f81158 d1707 b6
10- 838a 500773d85e fb 79266
11- ebf 7a305 f14 bf3143 b801a 222 cc1c857b7e8582119374925274 f9f335d
12- 83677c 102 f8d 68 bcce 722
ebba1f

```

21.1.4 Create encrypted media

page 145 The encrypted medium can then be created. When you enter the passphrase, you can copy the contents of the clipboard, as before, or transcribe it with it in front of you.

21.2 Reconstruct the passphrase

In order to reconstitute the passphrase (the secret), you need at least as many pieces as the minimum number decided at the time of cutting (three in our example).

page 113 Ideally, this step should also be carried out using a *live* system, so as not to leave any traces of the shared secret.

21.2.1 Install the necessary packages


As before, if the program is not available on the system, you need to install the `ssss` package and open a terminal.

page 135

21.2.2 Recombining the secret

To recombine the secret, use the `ssss-combine` program. You need to tell it how many pieces you have available:

```
2- $ ssss-combine -t NUMBER-OF-MORRORS-A-DISPOSITION
```

The program then prompts you to enter the available tracks. You need to type `Enter` () after writing each one. If all goes well well, the aff program will then display the complete passphrase.

To return to the previous example, this gives :

```
$ ssss -combine -t 3
Enter 3 shares separated by newlines:
Share [1/3]: 4 -143 a1efcde7 f4f5658415 a150 fcac6 da04f697
sbfeb947b5059d755510 b0e57 ccc594
Share [2/3]: 2-af83f0 af05fc207 e3b 466caef30 ec4 d39 c060800371f
ab9359350 b7699a8db9594bfc71 ed9 cd2bf314
Share [3/3]: 6- ebf7 a305 f14 bf3143 b801a222 cc1c857
b738
$ 7e858225277f9f335 d283677 f4c002f8
Resulting secret: 7 rZw00u+8 v1stea980uyU1efwNzHaKX9CuZ/
TK0bRWY=
```



Caution: if one of the pieces has been mistyped, the error that affiche is not necessarily very explicit:

```
$ ssss -combine -t 3
Enter 3 shares separated by newlines:
Share [1/3]: 4 -143 a1efcde7 f4f5658415 a150 fcac6 da04f697
sbfeb947b5059d755510 b0e57 ccc594
Share [2/3]: 2-af83f0 af05fc207 e3b 466caef30 ec4 d39 c060800371f
ab9359350 b7699a8db9594bfc71 ed9 cd2bf31
Share [3/3]: 6- ebf7 a305 f14 bf3143 b801a222 cc1c857
a738
$ 7e858225277f9f335 d283677 f4c002f8
Resulting secret: ..... L.fm.....6 _.... v.. w.a....[....
zS..... WARNING: binary data detected , use -x mode instead.
```

21.2.3 Open encrypted media

Once you've obtained the passphrase, you can copy and paste it to unlock the encrypted medium, or transcribe it with it in front of you.

Using checksums

🔄 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

🕒 *Duration: Five to ten minutes.*

In Part 1, we talked about *checksums*: "numbers" on page 53 that can be used to verify the integrity of a file (or any other data). The principle is that it is virtually impossible to have an identical checksum for all files.

two different files. If, in a letter, Ana tells Bea that on her site she can download a program that has the checksum SHA256 171a0233a4112858db23621dd5ffa31d269cbdb4e75bc206ada58ddab444651f and that the file Bea downloads has the same checksum, then it's almost certain that nobody has tampered with the program along the way. So she can run the program without too much fear.

There are several algorithms - or *hash functions* - for creating checksums. These include :

- MD5 is no longer secure and should be avoided;
- SHA-1 was widely used until 2017, when there was an actual attack on this algorithm. Since then, it has been used less and less. It should be abandoned;
- Those in the SHA-2 family (SHA-224, SHA-256, SHA-384 and SHA-512) will still be secure in 2022. We'll be using SHA-256 here, but the method also works with the other algorithms in this family.

22.1 Get the checksum of a file

Whether you want to check the integrity of a file, or enable your recipients to

To do this, we need to calculate the checksum of this file.

[page 53]

You can use a graphics tool just as well as a terminal to perform such calculations. However, we won't go into the details of using a terminal here.

22.1.1 Install the necessary software

If not already installed, install the `nautilus-gtkhash` package (see page 135), then restart your computer. This package is installed by default in Tails.

22.1.2 Calculate checksum

Open *Files* from the Activities overview: press  ( on a Mac), then type `files` and click on *Files*.

Select the file for which you wish to obtain checksums, then right-click on it. In the contextual menu that appears, select *Properties*, then go to the *Prints* tab.

Numerous *hash functions* are available, with three default selections: MD5, SHA1, SHA256. If a checksum other than these is required, tick the appropriate box. Click on *Hash*. The checksums now appear in the *Fingerprint* column.

22.2 Check file integrity

The checksum of the original file must be obtained by a secure means other than the one used to receive the file. For example, if you download the file, you can receive the checksum in a letter or by telephone - the best way, of course, being by word of mouth.

Similarly, to enable other people to check the integrity of a file we send them, we send them the checksum using the same methods.

Finally, using the method explained above, calculate the checksum of our copy of the file. Be careful to use the same hash function as the one used by our correspondent. If we use SHA1 and she uses SHA256, we obviously won't get the same checksum. If our correspondent offers us several checksums, prefer the hardest algorithm to break (see previous page), as mentioned at the start of this chapter.

Check that the two checksums are identical - it's a bit time-consuming and tedious. It's often easier to do this in pairs, or by pasting them one below the other in a text file.

Installing and operating a system virtualized

The aim of these recipes is to use a virtual operating system, i.e. to run several operating systems on a single computer, almost as if they were running on separate physical machines. The virtual system (called *guest*) runs inside our GNU/Linux system (called *host*): this is called *virtualization*. This technology, together with a security policy using it are described further in the use case on page 82, which explains how to work on a sensitive Windows document.




TO FIND OUT MORE...

There may be other reasons for using a virtualized system. For example, it is possible to boot a Tails key (see page 113) into a virtual system and use it without having to reboot the computer. It is even possible to install Tails directly in the *Virtual Machine Manager*¹, in the same way as other operating systems. It is important, however, to carefully consider and analyze the traces you may leave in the host system, in the guest system or in the metadata of documents created and shared.

23.1 Install the Virtual Machine Manager

 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

 *Duration: Approximately 15 minutes.*

23.1.1 Principle



The purpose of this recipe is to install the *Virtual Machine Manager*² which will enable us to run a virtual Windows system (or any other system) inside our Debian GNU/Linux system.

1. Documentation for this can be found on the Tails official site [<https://tails.boum.org/install/vm/index.en.html>].

2. Earlier editions of this *Guide* recommended the use of VirtualBox software. However, this is no longer available in Debian. If you used this tool before, you will need to either reinstall your virtual machine or migrate it from VirtualBox to Virtual Machine Manager. This procedure is not documented in this *Guide*, but you can get started by following the instructions available on the web: Malte Gerken, 2017, *Migrate a VM from VirtualBox to libvirt* [<https://maltegerken.de/blog/2017/01/migrate-a-vm-from-virtualbox-to-libvirt/>] (en English).


23.1.2 Install and launch the Virtual Machine Manager

page 134 The next step is to install the *Virtual Machine Manager* software.

Then, to launch *the Virtual Machine Manager*, open the Activities overview by pressing  ( on a Mac), then type `virt` and click on *Virtual Machine Manager*. You'll be asked for your administration password, which is normal.

23.2 Enable hardware virtualization

 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

 *Duration: Approximately 15 minutes.*

23.2.1 Principle


The vast majority of today's processors incorporate special hardware support for virtualization, known as *hardware virtualization*, so that virtualized systems run as smoothly as if they were running on a real physical machine. However, this feature is sometimes disabled by default on certain computers, making virtual machines extremely slow.

23.2.2 Test whether hardware virtualization is enabled

To check whether hardware virtualization is enabled on our computer, we can use a little program supplied with the Virtual Machine Manager.

To do this, use a terminal and type the following command:

page 97

```
 virt - host - validate
```

The `aff` command then displays several diagnostic lines. Look for the one entitled *QEMU: Verification for hardware virtualization* (normally the first one):

- If it's affiché *PASS* (in green) on this line, hardware virtualization is indeed enabled. We can then move straight on to the next part of this chapter.
- Otherwise, if it's affiché *FAIL* (in red) on this line, it means that hardware virtualization is disabled. We'll continue reading this section to activate it.

opposite page




23.2.3 Enable hardware virtualization in firmware

To activate this feature, you need to modify the computer firmware configuration:

page 108

- First, restart the computer, then enter the firmware configuration interface.
- If you are unfamiliar with the firmware configuration interface, please refer to the description in a previous chapter.
- Once in the firmware, look for something like *Virtualization Technology*, *VT-x* or *AMD-V* (which are the names of the hardware virtualization technologies in Intel and AMD processors, respectively). These options are usually found in the *Advanced* or *System Configuration* sub-menus. This option is probably marked as *Disabled*.

page 109

- Using the arrow keys, select the relevant option, then set it to Enabled, or by pressing the *Enter* key () or then selecting the correct value (if the firmware interface indicates something like Enter: Select in its help field), or by using the keys  and  keys (if the interface indicates +/-: Value).
- Once the correct value has been selected, save the change and exit the firmware configuration interface. page 111

23.2.4 Check that hardware virtualization is enabled



The computer then reboots: we can run the test again with the `virt-host-validate` command to make sure that hardware virtualization is now activated.

23.3 Installing a virtualized Windows

- 🔄 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*
- 🕒 *Duration: Around twenty minutes, plus time to install Windows (from thirty minutes to over an hour).*

First and foremost, download an ISO image of the desired Windows version. For example, you can find official ISO off images of recent versions of Windows on the Microsoft website ³.

23.3.1 Create a new virtual machine

To launch the Virtual Machine Manager, open the Activities overview by pressing  ( on a Mac), then type `virt` and click on *Virtual Machine Manager*.

The program starts, and you need to enter your password and authenticate yourself. Click on the *File* menu, then *New Virtual Machine*, and follow the five-step wizard. At the end of each step, click *Next* to move on to the next.

- Step 1: Select *local installation media (ISO image or CD-ROM)*.
- Step 2: To choose an installation media (ISO or CDROM) click on *Browse...*, a window opens. Click on *Browse locally* at the bottom, then select the ISO image you have downloaded. Click on *Open*. In the field *Choose the operating system you are installing*: if the system and its version are not well recognized, uncheck *Automatically detect from installation source/media* to choose it manually, e.g. *Microsoft Windows 10*.
- Step 3: specify the *memory* size and number of *CPUs* dedicated to the virtual machine. Here are the recommended minimums for the latest versions of Windows :

Version	Memory (RAM)	CPU
Windows 7	1024 MiB	1
Windows 8	2048 MiB	1
Windows 10	2048 MiB	1
Windows 11	4096 MiB	2

³. <https://www.microsoft.com/fr-fr/software-download>

- Step 4: Choose the size of the disk image allocated to the virtual machine. Given that we want to host an entire Windows system, it must be substantial: 20 GB is a minimum.
- Step 5: Enter a *Name* for the virtual machine, then select *Customize configuration before installation*.

Finally, click on *Finish*.

If a message *The virtual network is not active* is displayed, click *No*: we won't be using it for this virtual machine anyway.

In the left-hand column of the window that opens, select the *NIC (Network Interface Card)* hardware, which represents the virtual machine's network card, then click on *Remove* at the bottom. In the confirmation window, select *Yes*. The virtual machine is now isolated from the network.

Next, add a channel required for folder sharing between the host and guest systems. To do this, click on the bottom-left *Add Hardware* button. In the window that appears, click on *Channel* in the list on the left. In the *Name* drop-down list, select *org.spice-space.webdav.0*, then click on the *Finish* button.

Click on *Start installation* to start Windows installation.

23.3.2 Installing Windows in the virtual machine

The virtual system boots from the ISO file we've given it and begins installation. We won't go into the details of the process, which depends on our version of Windows, but we must point out that :

- To view the entire installer, choose the menu *Afficher* → *Scale affichage* → *Always*.
- Do not enter personal information when *Name* and *Organization* are requested. Put "user", for example.
- Similarly, if you want to enter a Windows serial number, a link could be made if it has been officially assigned.
- When configuring the network, an error message may be affiché. This is a good sign: we've disabled the virtual machine's network.

Once installation is complete, shut down the virtual Windows by clicking on the *Virtual Machine* → *Shut Down* → *Shut Down* menu. If the machine does not shut down, you can also select *Force Shutdown* from the same menu.

From the virtual machine window, click on the *Afficher* → *Details* menu. In the list on the left, choose *SATA CDROM 1* or *IDE CD-ROM 1* (depending on your ordina- teur), then clear the contents of the *Source directory* field and click *Apply*.

23.3.3 Guest tools for the Virtual Machine Manager


Specific drivers enhance interaction between Virtual Machine Manager and the guest Windows system, thanks to a technology called SPICE: these are the Guest Tools and Folder Sharing Service. These drivers make it possible, for example, to copy and paste or transfer files between the host and virtual guest systems. To achieve this, two small installation programs are used.

From the host system, download the Windows installer for SPICE guest tools⁴.

[page 345] To check the authenticity of the downloaded file, retrieve its signature⁵ and import _ _

the PGP key ⁶ key used to verify it. The key fingerprint ⁷ observed by the people writing these lines, assuming they have an original copy of the guide in their hands, is : [page 343]

```
94 A9 F756 61 F7 7 A61 6864 9 B23 A9D8 C214 29 AC 6 C82
```

Then go to the WebDAV installer web page for SPICE ⁸. Click on the download link for the latest version corresponding to our virtual machine architecture. The name contains *x86* for 32-bit Windows or *-64* for 64-bit Windows. Make sure there is also a signature file ⁹ file for this version. Check the authenticity of the file. If the file corresponding to the downloaded file is a *.sha256*, check authenticity using the checksum. Download the PGP key from this link ¹⁰ with your web browser: in the top-right drop-down menu , click on *Save as...* and save it, giving it a name followed by the *.asc* extension. Import this key to verify the authenticity of the file. The key fingerprint ¹¹ observed by the people writing these lines, assuming you have an original copy of the guide in your hands, is : [page 345]
[page 161]
[page 343]

```
206 D 3 B35 2 F56 6 F3B 0 E65 72 E9 97 D9 123 D E37A 484 F
```

Then follow the instructions for installing spice-guest-tools on a virtualized system. [page 169]

Do the same with spice-webdavd. Installing spice-webdavd may seem surprising: there's no message saying that installation is complete. Don't worry. [page 169]

Now you can copy and paste text between the host machine and the virtual machine. It is also possible to copy and paste files, but only from the host machine to the Windows virtual machine. If this doesn't work, try dragging the document from one window to another (the dragged file arrives on the Windows virtual machine desktop). Another feature is to modify the virtual machine's affage according to the size of the window hosting Windows. To do this, click on the menu *Afficher* → *Scale affichage* and select *Automatically adjust virtual machine to window*.

23.3.4 Backing up freshly installed virtual Windows

Virtual Windows installation is now complete, but there's more to come! Before working on sensitive documents inside the virtual machine, it's important to take a snapshot, i.e. to save the state of this *Windows*, which is considered "clean" because it's freshly installed.

4. <https://www.spice-space.org/download/windows/spice-guest-tools/spice-guest-tools-latest.exe>

5. <https://www.spice-space.org/download/windows/spice-guest-tools/spice-guest-tools-latest.exe.sign>

6. <https://keys.openpgp.org/vks/v1/by-fingerprint/94A9F75661F77A6168649B23A9D8C21429AC6C82>

7. The fingerprint of the imported PGP key can be verified using *Kleopatra* software [page 345].


8. <https://www.spice-space.org/download/windows/spice-webdavd/>


9. In other words, a file with the same name and a *.sig* extension.

10. <https://keyserver.ubuntu.com/pks/lookup?op=get&search=0x206d3b352f566f3b0e6572e997d9123de37a484f>

11. The fingerprint of the imported PGP key can be verified using *Kleopatra* software [page 345].



23.4 Take a snapshot of a virtual machine


 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

 *Duration: Five minutes.*


[page 82]


To follow the method for working on a sensitive document in Windows, you may need to save the state of a virtual machine that you consider to be "clean". To do this, we'll use the *snapshot* management of virtual machines.

To launch the Virtual Machine Manager, open the Activities overview by pressing  ( on a Mac), then type `virt` and click on *Virtual Machine Manager* and enter the password. Select the desired virtual machine and click *Open*. If it is running, switch it off by clicking on *Virtual Machine* → *Switch off* → *Switch off*.

Click on *Afficher* → *Snapshots*. In the list on the left, click on the  button at the bottom. In the window that appears, enter the *Snapshot Name*, avoiding the use of spaces and special characters, e.g. "Windows_own". Add a *Description* if required, then click on *Finish*.

23.5 Restore the state of a virtual machine from a snapshot

 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*



 *Duration: depending on disc size.*

[this page]


The aim of this recipe is to restore the state of a virtual machine from a previously created snapshot. This will enable it to be used for a new project, as recommended when working on a sensitive Windows document.

[page 82]

23.5.1 Afficher snapshots

To do this, open the Activities overview by pressing  ( on a Mac), then type `virt` and click on *Virtual Machine Manager*, then enter the password. Select the desired virtual machine and click on *Open*. In the new window, click on the *Afficher* menu and select *Snapshots*.

23.5.2 Selecting and restoring a snapshot

Select the desired snapshot from which to restore the machine state (e.g. "Windows_clean"). Click on the  button, bottom left. A new window appears, asking whether we are sure we want to run the selected snapshot. Executing this snapshot will mean that all modifications made in the virtual machine since its creation will be lost. If you're sure of your choice, click on *Yes*; if not, click on *No*.

The Virtual Machine Manager will restore the state of the virtual machine as it was when the snapshot was taken.

23.6 Installing new software on a virtualized system

🔄 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

🕒 *Duration: Approximately 20 minutes.*

To install software on a virtualized Windows system, you can use an ISO disk image of the software, as explained here. You can also use a CD or DVD.

It is advisable to start from a "clean Windows", and therefore to restore the state of a virtual machine considered "clean" from a snapshot. At the end of installation, this will create a new snapshot containing the software just installed.

page 171
previous
page.

23.6.1 Download and check software



If you don't already have it, start by finding the software, for example on the Internet. If possible, check the downloaded file. The downloaded program is an installer, a program that installs the software.


page 345

23.6.2 Create an ISO image of installation programs

To transfer an installer from the host machine to the Windows guest, it must be in ISO format. If the installer is already in ISO format, skip to the next paragraph. If not, we'll create an ISO disk image containing the installer using the Brasero software.¹²

this page



To launch Brasero, open the Activities overview by pressing  ( on a Mac), then type `bra` and click on *Brasero*.

Select *Data project* in the left-hand column. Click on the  icon and add the previously downloaded installer file.

In the drop-down list at the bottom of the window, select *Image file*, then click on *Burn...* Choose a file name and click on *Create image*. Once the image has been created, close Brasero.

23.6.3 Import the ISO image into the virtual system

Return to Virtual Machine Manager to share the ISO disk image.

To do this, open the Activities overview by pressing  ( on a Mac), then type `virt` and click on *Virtual Machine Manager*. Finally, enter the required password.

Select the Windows virtual machine on which you wish to install the software and click *Open*. In the new window, afficher the detailed view of the virtual machine by clicking on the menu *Afficher* → *Details*. In the hardware list on the left, select *IDE CD-ROM 1* or *SATA CDROM 1* depending on your computer's features. Then choose *ISO image location* or *Source directory* and click *Navigate*. In the window that appears, choose *Local Browse* and select the ISO image, then click *Open*. Then click on *Apply* at bottom right.

12. It may be necessary to install *Brasero* [page 134].

23.6.4 Install the software on the virtual machine


Do *Afficher* → *Console* to return to Windows. If the window affiche *Guest is stopped*, start the virtual machine with *Virtual Machine* → *Start*. Windows should detect the ISO image as if it were a CD/DVD. If it doesn't, we can go and find it in File Explorer (go to *This PC* → *CD Drive (D:)*). If it doesn't work the first time, repeat the operation.


To carry out the actual installation, double-click on the virtual CD-ROM and double-click on the file to launch the installation, its *Type* is an *Ap- plication*. Depending on the type of software installed, Windows may ask whether to authorize an unknown program (i.e. one not verified by Microsoft). If you trust your basic download, accept it. Accept all other installation program requests by clicking *Next*.


Once installation is complete, the ISO image is no longer required. Return to the virtual machine detail view with *Afficher* → *Details*, select *IDE CD-ROM 1* or *SATA CDROM 1* and delete the contents of the *ISO image location* or *Source directory* field, then *Apply*.

page 168 It is then possible to take a snapshot of a virtual machine to keep a "clean" version of the virtual system with this new software.



23.7 Sharing a USB stick with a virtualized system

 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*



 *Duration: Approximately ten minutes.*

 **Please note:** it is not always desirable for the virtual system to have direct access to the USB stick or external hard disk. Connecting a USB stick to the Windows system will automatically write data to it. To access key data without the virtual system having direct access, see the chapter on Sharing a folder with a virtualized system.

next
page.

To identify the key and find out under which name it is recognized, use Disk Utility. Start by opening the Activities overview by pressing  ( on a Mac), then type *disk* and click on *Disks*. In the Disks window, the left-hand side lists the disks known to the system. Connect the USB key to the computer, and it will appear in the list. Select it and make a note of its model name, which appears in the right-hand window of the software and usually contains the term *USB* or *Flash*.

23.7.1 Connect the key to the virtual system

Launch Virtual Machine Manager by opening the Activities overview and pressing  ( on a Mac), then type *virt* and click on *Virtual Machine Manager*. Finally, enter the required password.

Select the virtual machine to which the USB key will be connected and click *Open*. In the new window, click on the *Virtual Machine* → *Start* menu. For a view of the Windows system, click on the menu *Afficher* → *Console*

When the system has finished booting, click on the *Virtual Machine* → *Redirect to USB device* menu. In the window that opens, select the USB key recognized by its model name noted earlier. The virtual system recognizes it immediately, and you can close the window.


23.7.2 Eject the key and disconnect it from the virtual system

Start by ejecting the key from the Windows system. It is then important to remove the redirection to the USB key, so that the path linking the USB key and Windows is only active when required. To do this, click on *Virtual Machine* → *Redirect to USB device*. In the window that opens, uncheck the box corresponding to the USB key. You can then close the window.



The key is now no longer accessible from Windows, but is still visible from the host system. If it is no longer needed, it can be removed and disconnected from the computer.

23.8 Sharing a CD or DVD with a virtualized system



 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

 *Duration: Approximately ten minutes.*

23.8.1 Enable CD/DVD sharing

To do this, open the Activities overview by pressing  ( on a Mac), then type `virt` and click on *Virtual Machine Manager*. Finally, enter the required password.

Select the Windows virtual machine with which you wish to share a CD or DVD and click *Open*. In the new window, afficher the detailed view of the virtual machine by clicking on the menu *Afficher* → *Details*. In the hardware list on the left, select *IDE CD-ROM 1* or *SATA CDROM 1* depending on your computer's features.


Insert the CD or DVD into the drive and wait a few moments. Select *CD-ROM or DVD* on the right, then click *Apply*. The CD or DVD may have a name. To find it, open the Activities overview by pressing  ( on a Mac), then type `fic` and click on *File*. In the left-hand column, look for the DVD name.


Afficher the virtual machine screen with *Afficher* → *Console* and start it with *Virtual Machine* → *Start*. Windows should then detect the inserted CD. If not, try searching for it in File Explorer. If it doesn't work the first time, repeat the operation.

23.8.2 Eject CD/DVD


When you've finished using the CD in Windows, eject it from Windows, then return to the virtual machine detail view with *Afficher* → *Details*, select *IDE CD-ROM 1* or *SATA CDROM 1* and click *Eject*.

23.9 Sharing a folder with a virtualized system






 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

 *Duration: Approximately 15 minutes.*

Since the Windows *guest* is not allowed to go outside the box to fetch files, it may be necessary to send files from "outside". Let's see how to proceed.

 **Warning:** as you learn to use this sharing system, you may be tempted to configure it to give access to all disks connected to the host system: this **is the worst idea imaginable**, and would single-handedly destroy your entire security policy.

23.9.1 Create a dedicated folder on the host system

Open the Activities overview by pressing  ( on a Mac), then type `fic` and click on *Files*. Then choose the location where you want to put this exchange folder. For example: in the *Personal Folder*, click on the  button, then on the  icon with a small  at bottom right (*New Folder*) and give it an evocative name. ("*Folder readable by Windows*" or "*Folder where Windows can write*", for example). This is the folder in which you'll put the files you want to transfer to Windows.



23.9.2 Install the remote Afficheur

Currently, Virtual Machine Manager does not allow folder sharing to be enabled. It is necessary to use the *Afficheur remote* software. The next step is therefore to install the *Afficheur remote* software.

page 134



23.9.3 Enable folder sharing

To activate folder sharing, first start the Windows virtual machine from the Virtual Machine Manager.

To access the Virtual Machine Manager, open the Activities overview by pressing  ( on a Mac), then type `virt` and click on *Virtual Machine Manager*. Enter the required password.

In the Virtual Machine Manager window, right-click on the desired virtual machine (e.g. *Windows_own*) and click on *Start*.

The virtual machine then starts up, but its screen is not visible. We'll use the remote Afficheur to access it.

Open the Activities overview by pressing  ( on a Mac), then type `affi` and click on *Afficheur distant*. The first time, enter the virtual machine's address in the *Connection address* field. Typically, the address is: `spice://localhost:5900`. If more than one virtual machine is running, the first one started will have the address `spice://localhost:5900`, the second `spice://localhost:5901` and so on. From the second time, you can click on the desired address in *Recent Connections*. Click on the *Connect* button.

A *Allow shortcuts to be disabled* window appears. It asks if you want to neutralize shortcuts. To have the same operation between the Remote Afficheur and the *Virtual Machine Manager* choose *Allow*.



Warning: before checking the *Share folder* box, make sure you want Windows to read all the contents of the folder you've asked to share.

From the remote Afficheur window containing the Windows virtual machine, click on the *File* → *Preferences* menu. In the window that appears, select the folder you wish to share. To do this, select *Other* from the drop-down menu on the right. In the navigation window that opens, select the *Windows-readable folder* you've created, then click *Open*. Check the *Share folder* and *Read-only* boxes.



Always check the *Read-only* box unless you want to output files from virtualized Windows, in which case you should give the shared folder an explicit name such as *Folder where Windows can write*.



Warning: at the time of writing, a bug in the remote Afficheur means that the *Read-only* option is not taken into account. Consequently, even if this box is checked, the virtualized Windows will be able to write to the shared folder. If you want to share files with Windows, it's wiser to put copies rather than originals of these files there, so as not to run the risk of Windows modifying them.



23.9.4 Copy files

In the Windows virtual machine, open File Explorer. After a short while, *Spice Client (Z:)* should be accessible under *This PC*.

Spice Client (Z:) corresponds to the folder we've chosen to share on our host system, and it's possible to read all the files and folders it contains and copy what interests us to another folder in Windows.

23.9.5 Stop sharing

For one reason or another, you may want to stop sharing the folder with Windows.

After booting the virtualized system, open the Activities overview by pressing  ( on a Mac), then type `affi` and click on Afficheur distant. In *Recent Connections*, click on the desired address, `spice://localhost:5900` to access the first virtual machine started. Click on the *Connect* button.

From the remote Afficheur window containing the Windows virtual machine, click on the *File* → *Preferences* menu. In the window that appears, uncheck the *Share folder* box.


The selected folder is now no longer accessible from Windows.


Keeping your system up to date

As we explained earlier, malware finds its way into our computers, among other things, via "security holes".

Corrections for these programming (or design) errors are made available on a regular basis, as and when they are identified. Once these corrections are available, it is particularly important to replace older software versions. This is because corrected problems, which may previously only have been identified by a few specialists, are now publicly known and referenced... and therefore easier to exploit.

24.1 Keeping Tails up to date

 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

 *Duration: Thirty minutes to an hour, plus about thirty minutes download time.*

As a *live* system is an indivisible collection of software, executed from a DVD or USB key, the only practical solution for using the latest versions of this software is to make sure you're using the latest version of the *live* system.

page 113

After connecting the *live* Tails system to the Internet, an *Upgrade is proposed* or *New version is proposed* window appears to notify us when a new version that corrects security vulnerabilities is available.

If you're using a DVD, destroy the one containing the old version and burn a new one. Unless it's rewritable, in which case you'll need to erase it and burn the latest version of Tails.

If you have a USB key and an Internet connection, you can upgrade directly. Click on *Upgrade now* and follow the wizard through the process. If an error occurs, or if you need to use another upgrade method, the wizard will direct you to the appropriate documentation page.

This can be found in the *Tails documentation* on the desktop. In the index that opens, look for the *Downloading, installing and updating* section and click on the *Upgrading automatically* page.

24.2 Keeping an encrypted system up to date

page 119

Once installed, an encrypted system must be kept up to date so that it can continue to be trusted. The following sections focus on the Debian system, but the concepts are broadly applicable to virtually all other systems.

Every two years or so, the Debian project releases a *stable* version. This represents a huge effort to coordinate the compatibility of different software versions, carry out extensive testing and ensure that no major defects remain.

24.3 Daily updates for an encrypted system

- 🔄 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*
- 🕒 *Duration: One minute to launch the update, plus a variable amount of time for downloads and installation, during which you can continue to use your computer.*

The whole point of a *stable* release of Debian is that, afterwards, the software that makes it up is no longer modified in depth: updates include translation improvements, fixes for security-related problems or problems that prevent a program from being used normally, and *so on*.

Generally speaking, these new versions are installed automatically by the system, as long as it has Internet access, and should not disrupt the little habits you've already acquired.

24.3.1 Carry out updates

Once the *graphical desktop environment* has been installed, the system will automatically check the configured repositories for new versions when connected to the Internet.

page 136

When this is the case, a notification will appear indicating that *software updates are available*.

Click on *Afficher* in the notification, which opens *Software*.

A list of updates is displayed. If *Logiciels* hasn't already downloaded them all, a *Download* button appears, which you click to ask it to do so.

Once the updates have been downloaded, the *Restart and Update* button appears. Click on this button, then confirm by clicking on *Restart and install* again. The computer reboots and asks for the hard disk encryption passphrase, before installing the updates. The computer reboots on an up-to-date system and asks for the passphrase.

24.3.2 Remove obsolete packages

Once the computer has been restarted, we still need to ask the system to remove software components that are no longer required: as this operation is not performed automatically by the system, we need to do it regularly, otherwise our disk - and in particular the */boot* partition - will gradually fill up, to the point where it will no longer be possible to perform new updates.

It is not yet possible to perform this operation via the graphical interface, so you need to open a Terminal.

Start by becoming an admin by typing the command :

page 137

```
sudo su
```


The computer should ask for our session password. If we get
 bash: sudo: command not found, so type :


```
su -
```


Our terminal now has administrative power over our system.

The following command, to be typed in this terminal, then removes the obsolete packages:

```
apt autoremove
```

24.4 Upgrade to a new stable version

 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

 *Duration: Half a day to a full day, including a long download period during which you can continue to use your computer, and a long installation period during which you'd better stop using it.*

When a new *stable* version of Debian is released, the project keeps the previous *stable version*, called *oldstable*, up to date for **at least one year**.¹ minimum. This period is extended by a *Long Term Support* team.² For example, the version of Debian used in the 2017 edition of this guide, Debian 9 Stretch, was only supported by the Debian security team until July 2020, and the *Long Term Support* team took over maintenance until June 2022.





It is therefore necessary to take advantage of this period to update your system to the new stable version. This is a more delicate process than daily updates. Not necessarily in the actual implementation, but in the fact that it is then necessary to adapt to changes in the software we usually use.

In any case, before continuing, we strongly recommend that you back up your data. page 151

We have two options:

- Update our system to the new stable version. The advantage of this is that it allows you to keep the software you've installed and the configurations you've made over time... which can also be a disadvantage if you've tinkered too much. If you choose this option, continue to use this tool.
- Install the new version of Debian. The advantage is that you get a clean slate. The disadvantage is that we lose our specific configurations, and we have to download and check the installation program again. If you choose this option, once the backup has been made, you'll find the rest of the process in the tool for making a new Debian installation. page 119

The previous update of the *Digital Self-Defense Guide* used Debian version 9, called Stretch, released in June 2017. At the time of this guide update, Debian is at version 11, called Bullseye, released in August 2021. It's risky to go straight from version 9 to 11 without going through version 10, called Buster, released in July 2019.

To find out which version of Debian you're using, open the activity overview by pressing   (  on a Mac), then type `param` and click on

1. [Debian, 2017, DebianOldStable](https://wiki.debian.org/fr/DebianOldStable) [https://wiki.debian.org/fr/DebianOldStable].

2. [Debian, 2021, Debian Long Term Support](https://wiki.debian.org/fr/LTS) [https://wiki.debian.org/fr/LTS]

Settings. In the left-hand column, scroll to the bottom and click on *About*. The Debian version in use appears under *Operating system name*.

If you're still at version 9 Stretch, we'll propose a two-stage upgrade, from version 9 Stretch to version 10 Buster, then from version 10 Buster to version 11 Bullseye. It is possible to follow only the second step if you are already at version 10 Buster.

24.4.1 From Stretch to Buster

The procedure detailed here concerns upgrading from the version of Debian called Stretch or 9, released in June 2017, to Buster or 10, released in July 2019.

Here we'll document a simplified upgrade procedure that has been tested on Debian Stretch installations with a GNOME graphical desktop environment and software sourced solely from Debian's official repositories.

It requires an Internet connection for the duration of the update.



Warning: this simplified procedure is less likely to work when you've tweaked your system by adding non-official update sources.

If this is the case, please refer to the official Debian off release notes ³in particular the section Upgrades from Debian 9 (Stretch) ⁴ and Problems to be aware of for Buster⁵.

Updating Debian Stretch



page 176

First and foremost, you need an up-to-date Debian Stretch. Without it, the upgrade may not work. In case you haven't been updating on a daily basis, now's the time to catch up. If you're prompted to reboot after numerous updates, do so before proceeding with the next steps.

Make sure you have enough free space on your hard disk

To avoid any unpleasant surprises, you must have at least 4 GB of free space on the hard disk containing the system.

Open the Activities overview by pressing  ( on a Mac), then type `fich` and click on *Fichiers*. In the left-hand bar, click on *Other locations*. To the right of the *Computer* line, the available space is affiche, e.g. *11.7 GB/17.1 GB available* means 11.7 GB available.

Free up disk space if necessary If there isn't enough space on the hard disk, one solution is to delete old updates that have become obsolete. To do this, open the Activities overview by pressing  ( on a Mac), then type `package` and click on *Package Manager*. Since the Package Manager allows you to modify the software installed on your computer, a password is required to open it.

In the *Configuration* menu, choose *Preferences*, then select the *Files* tab and click on the *Delete cached packages* button, then *OK* and close *Synaptic Package Manager*.

Check available disk space again, as explained above. If this doesn't suffice, we'll have to delete some of our own files or remove logi- cials.

3. <https://www.debian.org/releases/buster/amd64/release-notes/index.fr.html>

4. <https://www.debian.org/releases/buster/amd64/release-notes/ch-upgrading.fr.html>

5. <https://www.debian.org/releases/buster/amd64/release-notes/ch-information.fr.html>

Disable non-official repositories

The update is only tested with the packages officially provided by Debian Stretch. We will therefore disable all other Debian repositories, including *backports*.

To do this, open the Activities overview by pressing  ( on a Mac), then type `update` and click on *Software & Updates*.



In the *Other software* tab, uncheck any checkboxes. When making a change, enter the admin password.

Click on *Close*. If any changes have been made, a window affiche *Information on available software is out of date*. Click on *Update*.

Disable screensaver

During the update, the screensaver may freeze, leaving the screen worm-eaten. It is therefore advisable to deactivate it for the duration of the update.


To do this, open the Activities overview by pressing  ( on a Mac), then type `param` and click on *Settings*. In the left-hand column, click on *Privacy*.

Click on *Screen lock*. In the window that appears, deactivate *Automatic screen lock*. Close this window by clicking on , then again on  in the top right-hand corner, to close the *Settings* window.

Open a terminal

As it is not yet possible to perform this operation via the graphical interface, it is necessary to open a Terminal. [page 97]

Start by becoming an admin by typing the command :

```
 sudo su
```

The computer should ask for our session password. If we get `bash: sudo: command not found, so type :`



```
 su -
```

Our terminal now has administrative power over our system.

Update deposits

Let's start by modifying the configured repositories to use those dedicated to the new version. We'll open the file containing the list of repositories used by Debian in the terminal.

```
 gedit / etc/ apt/ sources. list
```

The text editor opens. Choose  → *Find and replace*. In the window that opens, *Search for "stretch" to Replace it with "buster"*. Then click on the *Replace All* button, and close the search window with .

If an installation or update was previously performed using a CD or DVD, it's a good idea to look for lines starting with `"deb cdrom:"` and remove them.

You can then click on *Save* and close the editor.

We've modified the list of repositories, so we now need to download the list of packages available in them, before we can install them; to do this, always in *Terminal*, which we'll keep open, type the command :

```
# apt update
```

If an error occurs about the "AppStream system cache", you can ignore it without worry.

Start the update itself

The update is performed in several stages, each of which is controlled from our Terminal.

Our first command tells the package manager, on the one hand, that we'd like it to ask as few questions as possible about the details of the update; on the other, that we don't want to see the history of changes:

```
# export DEBIAN_PRIORITY=critical APT_LISTCHANGES_FRONTEND=none
```

Our second command performs the first part of the system update: Soon

```
# apt upgrade
```

enough, the terminal affiche `Would you like to continue [Y/n]?` After confirm by pressing `Enter` (`↵`) or `return` (`↵`), a first series of blue windows asking us how to manage certain change- ments. When you're not trying to get out of Debian's choices, pressing `Enter` each time is suffisant.

A window indicating that a configuration file has been modified and asking if we want to replace it with the new version may also appear. *Keep* or *Replace* is a choice that depends on the extent of the changes you've been able to make to it, as well as the new features on offer. So there's no generic answer here. You'll either have to compare the versions, or flip a coin.

After a while, a number of packages have already been updated, and the terminal should return to the command prompt.

The following command will complete the system update:

```
# apt full - upgrade
```

Soon enough, the terminal affiche again `Would you like to continue? [Y/n] ?` After confirming by pressing `Enter` (`↵`) or `return` (`↵`), you can see a second series of blue windows appear, asking us how to manage certain changes. When you're not trying to get out of Debian's choices, pressing `Enter` each time is suffisant.

At this stage of the update, it may happen that the GNOME desktop affiche di- vers error messages. This is not particularly worrying, as many system components are being reinstalled. These problems should resolve themselves once the process is complete.


A few system evolutions later, the terminal once again prompts us for commands.

You can then enter a final command to free up disk space: Then :

```
# apt autoremove
```

```
# apt clean
```

First restart

Now it's time to reboot the system, using the  menu in the top left-hand corner and choosing *Reboot*.

24.4.2 From Buster to Bullseye

The procedure detailed here concerns the upgrade from the Debian version called Buster or 10, released in July 2019, to Bullseye or 11, released in August 2021.

Here we'll document a simplified upgrade procedure that has been tested on Debian Buster installations with a GNOME graphical desktop environment and software sourced solely from Debian's official repositories.

It requires an Internet connection for the duration of the update.



Warning: this simplified procedure is less likely to work when you've tweaked your system by adding non-official update sources.

If this is the case, please refer to the official Debian off release notes ⁶in particular the section Upgrades from Debian 10 (Buster) ⁷ and Problems to be aware of for Bullseye⁸.

Updating your Debian Buster



First and foremost, you need an up-to-date Debian Buster. Without it, the upgrade may not work. In case you haven't been updating on a daily basis, now's the time to catch up. If you are prompted to reboot after numerous updates, do so before proceeding.

page 176

Make sure you have enough free space on your hard disk

To avoid any unpleasant surprises, you must have at least 4 GB of free space on the hard disk containing the system.

Open the Activities overview by pressing  ( on a Mac), then type `fich` and click on *Fichiers*. In the left-hand bar, click on *Other locations*. To the right of the *Computer* line, the available space is affiché, e.g. *11.7 GB/17.1 GB available* means 11.7 GB available.

Free up disk space if necessary If there isn't enough space on the hard disk, one solution is to delete old updates that have become obsolete. To do this, open the Activities overview by pressing  ( on a Mac), then type `package` and click on *Package Manager*. Since the Package Manager allows you to modify the software installed on your computer, a password is required to open it.

In the *Configuration* menu, choose *Preferences*, then select the *Files* tab and click on the *Delete cached packages* button, then *OK* and close *Synaptic Package Manager*.

If there isn't enough space on the hard disk, we'll have to delete some of our own files or remove software.

6. <https://www.debian.org/releases/bullseye/amd64/release-notes/index.fr.html>



7. <https://www.debian.org/releases/bullseye/amd64/release-notes/ch-upgrading.fr.html>

8. <https://www.debian.org/releases/bullseye/amd64/release-notes/ch-information.fr.html>

Disable screensaver

During the update, the screensaver may freeze, leaving the screen worm-eaten. It is therefore advisable to deactivate it for the duration of the update.

To do this, open the Activities overview by pressing  (on a Mac), then type  param and click on *Settings*. In the left-hand column, click on *Privacy*.

Click on *Screen lock*. In the window that appears, deactivate *Automatic screen lock*. Close this window by clicking on , then again on  in the top right-hand corner, to close the *Settings* window.

Open a terminal

It is not yet possible to perform this operation via the graphical interface, so you need to open a Terminal.

page 97 Start by becoming an admin by typing the command :

```
sudo su
```



The computer should ask for our session password. If we get `bash: sudo: command not found`, so type :

```
su -
```



Our terminal now has administrative power over our system.

Update deposits

Let's start by modifying the configured repositories to use those dedicated to the new version.

Please note: this is where the difference with the previous update lies.

In the terminal, type :



```
sed -i 's, buster/ updates , bullseye - security ,g' / etc/ apt/ sources. list
```



```
sed -i 's, buster , bullseye ,g' / etc/ apt/ sources. list
```

You can then click on *Save* and close the editor.

We've modified the list of repositories, so we now need to download the list of packages available in them, before we can install them; to do this, always in *Terminal*, which we'll keep open, type the command :

```
apt update
```



Start the update itself

The update is performed in several stages, each of which is controlled from our Terminal.

Our first command tells the package manager, on the one hand, that we'd like it to ask as few questions as possible about the details of the update; on the other, that we don't want to see the history of changes:

```
export DEBIAN_PRIORITY=critical APT_LISTCHANGES_FRONTEND=none
```



Our second command performs the first part of the system update:



```
apt upgrade
```

Soon enough, the terminal affiche Would you like to continue [Y/n]? After confirm by pressing *Enter* (↵) or *return* (↵), a first series of blue windows asking us how to manage certain change- ments. When you're not trying to get out of Debian's choices, pressing *Enter* each time is suffisant.

A window indicating that a configuration file has been modified and asking if we want to replace it with the new version may also appear. *Keep* or *Replace* is a choice that depends on the extent of the changes you've made to it, as well as the new features it offers. So there's no generic answer here. You'll either have to compare the versions, or flip a coin.

After a while, a number of packages have already been updated, and the terminal should return to the command prompt.

The following command will complete the system update:

```
# apt full - upgrade
```

Soon enough, the terminal affiche again Would you like to continue? [Y/n] ? After confirming by pressing *Enter* (↵) or *return* (↵), you can see a second series of blue windows appear, asking us how to manage certain changes. When you're not trying to get out of Debian's choices, pressing *Enter* each time is suffisant.

At this stage of the update, it may happen that the GNOME desktop affiche di- vers error messages. This is not particularly worrying, as many system components are being reinstalled. These problems should resolve themselves once the process is complete.

A few system evolutions later, the terminal once again prompts us for commands.

You can then enter a final command to free up disk space: Then :

```
# apt autoremove
```

```
# First restart  
apt clean
```

Now it's time to reboot the system, using the  menu in the top left-hand corner and choosing *Reboot*.

Reactivate additional Debian repositories

Now we can breathe a sigh of relief. Most of the work is done. But there are still a few minor adjustments to be made...

If you've deactivated non-official repositories before the upgrade, now's the time to check that you still need them with the new version of Debian. If so, reactivate them. You can also reactivate the screen saver if you previously disabled it.

page 136

Reactivate screen lock

To do this, open the Activities overview by pressing  (⌘) on a Mac), then type *param* and click on *Settings*. In the left-hand column, click on *Privacy*.

Click on *Screen lock*. In the window that appears, activate *Automatic screen lock*. Close this window by clicking on **✕**

Make sure the new system works properly


It may be useful to ensure that the most common actions and commands are functional. If necessary, it may be necessary to diagnose and resolve any problems. It's certainly best to do this as soon as you start using the new system, so that you can leave for another two years with a functional system. The most common problems are often described, along with tips on how to solve them, in various Debian and GNU/Linux documentations.


[page 129]

We should also point out that there are official release notes from the Debian project.⁹

9. <https://www.debian.org/releases/bullseye/amd64/release-notes/index.fr.html>


Clean metadata from a document

 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

 *Duration: a few minutes.*

The aim of the tool we're going to examine is to delete the metadata present in page 30 a document before it is published. This metadata is not the same in all document formats: some are more difficult to clean than others, impossible. However, most of the formats used to exchange finished documents, whether text, images, sound or video, are "cleanable".

The tool to use for this is *MAT2* (for *Metadata Anonymization Toolkit 2*), which makes it easy to clean up a wide range of file formats.

 **Warning:** cleaning metadata does not anonymize the content of files, nor does it remove any markings¹ included in the content itself.

25.1 Install the necessary software

On a system where it is not yet present, you need to install the package (see page 135). `mat2`. Under Tails, *MAT2* is already installed.

25.2 Clean one or more files

In the file manager, right-click on the document whose metadata you wish to remove, then select *Remove metadata*. A new document without metadata is created. It bears the name of the original file, followed by *.cleaned* and the file extension.

Tip! To process several files, you can select a set of files and right-click on *Remove metadata*. This operation may take some time, depending on the number of files and their size.

Some formats are not supported by this tool. In this case, a warning message *Failed to clean some items* appears. A *Show* button displays a list of files that have not been processed. If the format is not supported, it is possible to export the file that cannot be processed in a more common format. For example, to clean up a file in XCF format from the GIMP image manipulation program, it is possible to export it in JPEG or PNG format.

1. See [Wikipedia, 2014, Digital tattoo](https://fr.wikipedia.org/wiki/Tatouage_num%C3%A9rique) [https://fr.wikipedia.org/wiki/Tatouage_num%C3%A9rique] and [Wikipedia, 2014, Steganography](https://fr.wikipedia.org/wiki/St%C3%A9ganographie) [https://fr.wikipedia.org/wiki/St%C3%A9ganographie].

25.2.1 Special case of PDF files

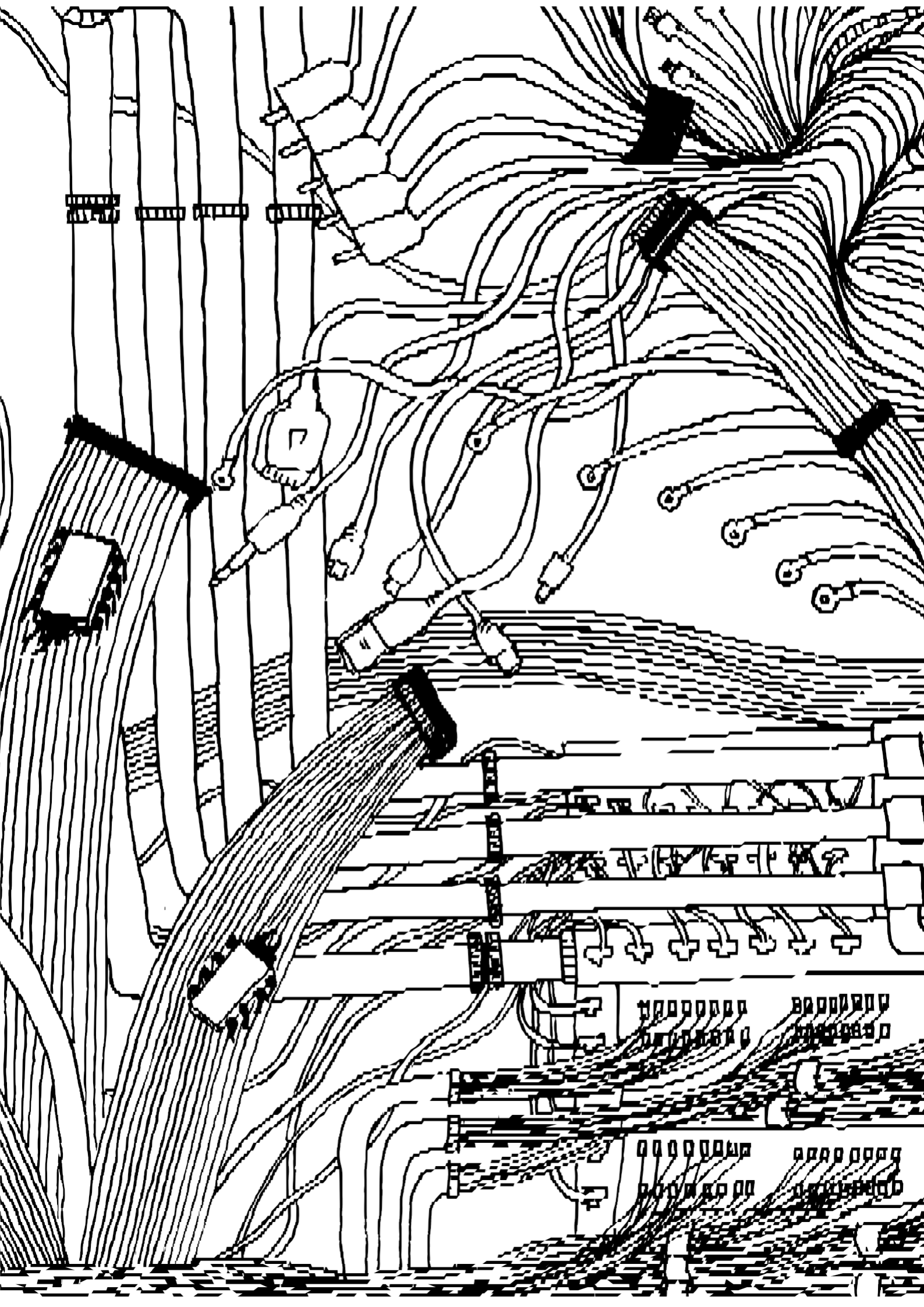
To correctly remove metadata from a PDF file, MAT2 "transforms" it into an image. This means that a PDF file without metadata will lose all its hyperlinks and will be larger than the original file.

25.2.2 Special case of videos

MAT2 removes metadata from a video file, but is unable to remove other traces that could sometimes help identify the source of the video: scratches or fingerprints on the lens, for example, or as we saw above, invisible and undetectable marks (known as *digital watermarks*) that could be added directly to the video images by the capture hardware or software used.

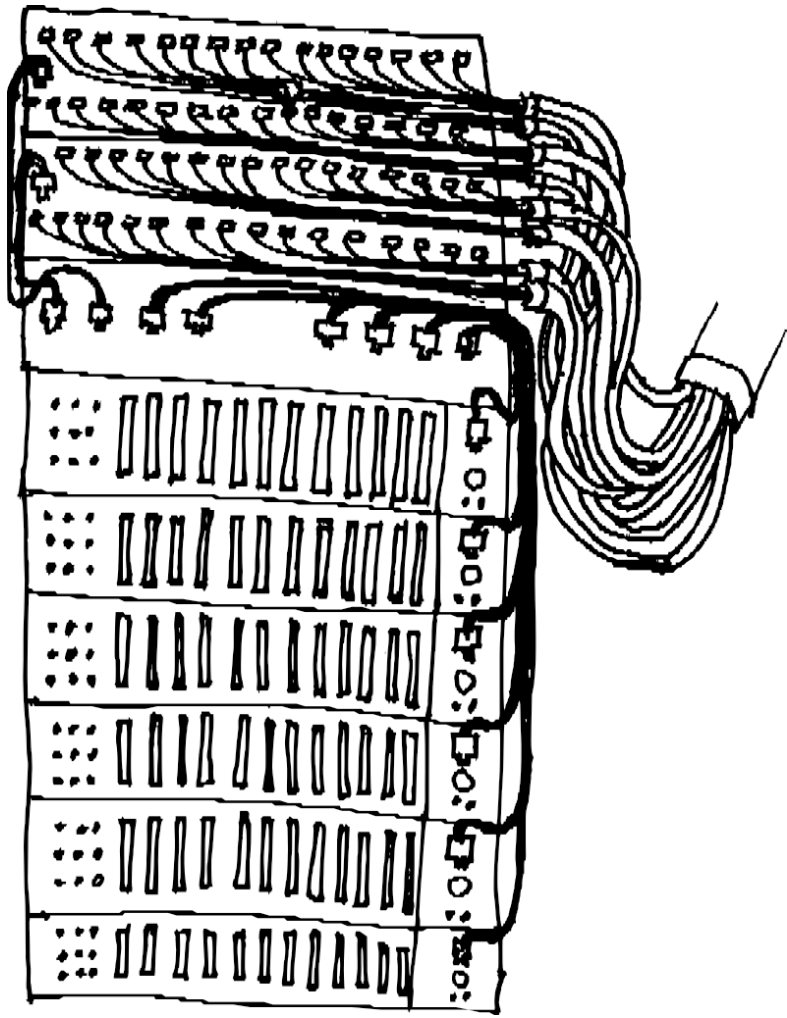
So, to ensure that a video really does not contain any traceable information, MAT2's metadata deletion is not enough: you also need to make the video with material that is not linked to any identity (i.e. that has never been used to publish images with another contextual identity), and use only Tails to edit it.

Nevertheless, in most cases and in the face of most adversaries (and their means) who would like to identify the author of a video, deleting the video's metadata with MAT2 is already a pretty good protective measure.



VOLUME 2

On line



PART FOUR

Understanding

Introduction

In the first volume, we explained that the use of computers leaves traces of our activities and data. Diving into the mysteries of these familiar machines had already proved a little complex. What will it be like now that we're proposing to connect to the Internet? What does it mean to connect our computer to other computers over which we have little or no control? A connected computer is first and foremost a computer, so it's essential to read the first volume in order to get to grips with this *volume 2* on *online* security.

*
* *

Let's start at the beginning. The Internet is a network. Or rather, a set of interconnected networks which, starting with an obscure military application, has expanded over the decades to cover the entire world. A network that has seen a proliferation of applications, users, technologies and control techniques.

Many have gone on endlessly about the "new age" that was opening up, the supposed possibilities of horizontality and transparency in the dissemination of information and resources, or in collective organization, to which this new technology could open up - including in the support it could offer for political struggles. However, as it seems obvious that the powers that be don't like what they can get away with, even partially, the expansion of uses has been accompanied by an expansion of control, surveillance and repression techniques, the consequences of which are becoming increasingly apparent.

In 2011, for the first time, governments organized the disconnection of almost their entire populations from the global network. The leaders of Egypt and Iran, as it happens, felt that to better contain the revolts taking place on their soil, they had every interest in limiting the possibilities of communication via the network as much as possible - which didn't stop them, in the same movement, from seeking to organize surveillance and tracking on the Internet. The Iranian government was thus able to set up a traffic analysis system requiring considerable resources to monitor rebels, whether known or not, map out their relations in order to confound them, and condemn rebels who used the network to organize themselves.

Another example: since the introduction of a Chinese version of Google¹ in 2006, the company has accepted the Chinese government's policy of filtering search results with varying degrees of docility.

Similar methods are also used in so-called democratic countries. For example, at the end of summer 2011, after several days of rioting in London, two young men

1. [Wikipedia, 2017, Google China](https://fr.wikipedia.org/wiki/Google_China) [https://fr.wikipedia.org/wiki/Google_China].

have been sentenced ² to 4 years in prison for calling for rallies in their neighborhoods on Facebook - even though their "calls" were not followed up.

Similarly, Edward Snowden's revelations ³ on the state of electronic surveillance carried out by the NSA ⁴ have given credence to some of the most pessimistic hypotheses.

That's why it's vital to realize that using the Internet, like using computers in general, is anything but harmless. It exposes us to surveillance, and the repression that can follow: the main aim of this second volume is to help everyone understand the risks and limits associated with Internet use. But it's also about giving ourselves the means to make informed choices about how we use the Internet. Choices that can complicate the task of gatekeepers, bypass censorship systems, or even enable us to set up our own tools and infrastructures. A first step towards regaining control of technologies that sometimes seem destined to escape us - an ambition that goes far beyond the scope of this guide.

*
* *

October 2010, Paris

Ana arrives early at work this morning. She works at La Reboute, a mail-order clothing company located on the top floor of a building on Rue Jaurès: "Phew, 18 floors, can't wait to get that elevator fixed!" She sits down at her desk, leans over and presses the computer's power button.

A small window appears on the screen. "Network connection established". Before getting down to work, she wants to check her e-mails. Ana clicks on the web browser icon, opening a window that remains blank for a few milliseconds, before bringing up the Google home page. While mentally enjoying the home page On Google's "Halloween special" page, Ana moves her mouse pointer and clicks on the Login link. Once the page has loaded, she enters her user name and password, then clicks on Gmail. Somewhere in an obscure room crowded with computers, a hard disk crackles. Seconds after opening her web browser, Ana starts browsing her inbox. As she consults an e-mail received from the leboncoin.fr, his gaze is drawn to the link that has just been affichered in the right-hand column: "Well, someone's selling the same model of appliance photo than the one I'm looking for, just around the corner... maybe I should drop in."

— "Ah well you're here?"

The voice at Ana's back startled her slightly. It's Bea, a colleague.

— "I got up a little earlier than usual, so I took the 7:27 a.m. RER instead of the 7:43. I quickly check my emails

2. France Soir, 2011, *Émeutes à Londres : Deux jeunes condamnés à quatre ans de prison* [<http://archive.francesoir.fr/actualite/international/emeutes-londres-deux-jeunes-condamnes-quatre-ans-prison-128302.html>].

3. Wikipedia, 2014, *Edward Snowden* [https://fr.wikipedia.org/wiki/Edward_Snowden].

4. *National Security Agency*, part of the U.S. Department of Defense, responsible for collecting and analyzing foreign data and protecting U.S. data.

before I get started. I'm waiting for confirmation of a ticket booking for the Balearics this winter.

— *Vacations in the sun, I know the type... And will you be long?"*

Bea looks in a hurry.

— *"Uh... no no, I was almost done. Why was that?"*

— *Well, if you don't mind, I'd like to borrow your computer for a couple of minutes... Mine's been down since yesterday, so I'm waiting for the new IT manager to arrive to sort it out.*

As soon as she sits down, Bea nervously clicks on the browser's address bar, and directly enters the address of the blog on which information about the political figures in her district is regularly published. She doesn't like to use Google for her research, so she's memorized it. You never know, it might prevent snitches. Opening a second tab, she also enters the address of no-log, her mailbox, and logs in. Nickel, here it is! The document concerning the Swiss bank accounts of the mayor of her arrondissement, Mme Alavoine! Bea immediately downloads the document and opens it in the text editor. She quickly scans through it, deleting a few details that are best left alone. After entering her username and password to log in to the blog, Bea copies and pastes the contents of the document from your mailbox, and click on Send. "Let's hope it inspires other people!"

Satisfied at finally being able to send her document, Bea immediately stands up and gives Ana back her seat.

— *"Shall we get a coffee?"*

November 2010. La Reboute head office

On arriving at the office, Sarah Ahmed, CEO of La Reboute, starts by going through the mail she's received while drinking her coffee. A summons to the police station. For once, there's something other than bills! No doubt a mistake or a neighbourhood inquiry?

Sarah doesn't think she has anything to be ashamed of, so there's no need to worry. So she goes to the police station on the day she's summoned.

— *"Mrs Ahmed? Hello, we'd like to ask you a few questions about a libel claim..."*

Later the same day. Ana's office

— *"Hello, human resources at La Reboute, Ana speaking.*

— *Hello, Mrs Ahmed speaking. Listen, I've just spent two hours at the police station. I was questioned about bank documents published on the Internet concerning a certain Mme Alavoine, mayor of the 10^e, who I didn't know existed until then. What's more, during my interview, they presented me with a paper authorizing them to search the rue Jaurès offices.*

— *What a story! But what's it got to do with our offices?*

— *Well, that's also why I'm calling you. They affirm that they have all the evidence that these documents were published from your offices. I told them it wasn't me, that I couldn't see anything.*

- what they were talking about. They've made enquiries, contacted I don't know who. But they say that an investigation has been opened, and that it will go all the way. That they will find those responsible. I might as well tell you that I'm not exactly reassured. I do hope that you had nothing to do with it and that it was an unfortunate mistake.*
- *Honestly, I'm the first to be astonished - I don't see what I've got to do with it, or what it's all about.*
 - *I hope so... Anyway, it's up to the police to do their job now. I'll call you back if I hear from them.*
 - *Okay, I'll do the same if they call here.*
 - *Goodbye."*

Ana puts the handset down again, dazed. Scratching her head. What's all this about bank documents? Who could have done this?

Paris Central Police Station, a few weeks later

- *"Commissioner Marta?*
- *The same.*
- *Officière Neus speaking. I'm calling about the Ala-voine case. We got an email from the technical and scientific colleagues working on the seized computers. And we've got something new.*
- *Go ahead, Neus. Go ahead.*
- *Apparently, colleagues ended up finding the document on a certain Ana's workstation. It had been downloaded from the web browser and modified. There was a connection to a Gmail mailbox, as well as another mail address, this time at no-log, shortly before the publication of the incriminating documents.*
- *Ah, very good. Now we know who to summon for questioning! But how can we get proof?*
- *We're going to ask Gmail and no-log for information on these email addresses. From there, we'll undoubtedly have something to go on, or at least enough to ask the right questions!*
- *Good, Neus. Very good, sir. I'll be in touch with the DA. And let me know as soon as there's any news.*
- *Yes, commissioner. Good day."*

So much for context. This little fictitious story may remind you of others, much more real. The idea was simply to show how quick and easy it is to *expose yourself* at the slightest Internet connection, without any form of targeted surveillance.

One of the aims of this second volume is to shed light on the digital traces that could lead back to Ana and Bea. Then, to point out a few ways of protecting ourselves from attacks - targeted or otherwise.

Network basics

The Internet is not a virtual space, an abstract cloud of information where you can find anything and everything. At least, that's not all it is.

The Internet is first and foremost a set of networks¹. Millions of networks, aggregated over several decades and, in a more or less chaotic way, managed by companies, universities, governments, associations and private individuals alike; millions of computers and materials of all types, linked together by a wide variety of technologies, from copper cable to fiber optics to wireless.

But for us, behind our little screen, the Internet is above all what it allows us to do: visit websites, send e-mails, chat with people or download files. New applications are appearing all the time, and only the human imagination seems to limit the possibilities.

Understanding how the Internet works, and how to protect yourself, means unraveling this complexity to understand how these computers communicate with each other, and how the various applications we use work.

26.1 Interconnected computers

Quite early on in the history of computing, it became apparent that computers needed to be able to share resources and information, particularly in the academic and military fields - and over ever greater distances. And so computer networks were born. First, computers were linked together in a restricted area - usually a university, a company or a military site - and then these areas were linked together. In the USA, in the late 1960s, ARPANET (*Advanced Research Projects Agency Network*) was created, a network linking universities across the country. Many of the techniques used today on the Internet were invented to set up and improve the network. The birth of the Internet is linked to that of free software, and it operates according to similar principles of openness and transparency.² However, it was originally developed to meet military needs.

The various computer networks were interconnected, forming the Internet, which has been expanding rapidly since the 1990s.

1. For a five-minute explanation: Rémi explains, 2015, *Internet! How does it work?* [<https://www.youtube.com/watch?v=dCknqjcItU>]. For a detailed explanation in four hours: Benjamin Bayart, 2012, *Qu'est-ce qu'Internet? - Cycle de conférences à Sciences Po* [<https://www.fdn.fr/actions/confs/qu-est-ce-qu-internet/>].

2. According to Benjamin Bayart, "you can't dissociate Internet and free software" because they are appeared on the same dates, had the same players and similar growth and functioning. Benjamin Bayart, 2007, *Internet libre, ou Minitel 2.0? conference at the 8^{es} rencontres mondiales du logiciel libre, Amiens* [<https://www.fdn.fr/actions/confs/internet-libre-ou-minitel-2-0/>].

More and more objects - whose primary function is not to be a computer - are connected to the Internet: surveillance cameras ³ speed cameras ⁴ PMU terminals ⁵ fridges ⁷ medical equipment ⁸ children's toys ⁹ cars ¹⁰ etc. Some people even speak of the *Internet of Shit* ¹¹ (to show the absurdity of many of the objects that are making their way onto the Internet).

26.1.1 A computer network

"A network is a set of nodes [...] connected by links". ¹² In a computer network, the nodes are computers. It's a set of computers linked together by cables, waves, etc. to form a network.

Not all computers used in networks are like the personal computers, fixed or portable, that we generally use. Some are specialized to perform particular functions within the network. For example, the "box" that enables most of us to access the Internet is a small computer; similarly, the servers on which websites are stored are also computers. Other types of specialized computer could be added to this list, some of which are described in the following pages.

26.1.2 Network card

Despite their differences, all computers connected to a network necessarily have one thing in common: in addition to the minimum hardware that makes up a computer, they must have at least one peripheral used to connect to the network. This is called a *network card*. It establishes the link with other computers. Nowadays, several network cards are often integrated into every personal computer (a wired network card and a Wi-Fi card, for example).

Every network card has a hardware address, which identifies it more or less uniquely. In home wired technology, called Ethernet, as in wireless *Wi-Fi* technology, this hardware address is called *the MAC address*. The MAC address supplied with the card is designed to ensure that the probability of two network cards having the same hardware address is very low. ¹³ This poses a problem in terms of anonymity, as we'll see later.

3. Jérôme G., 2012, *Caméras IP : faille-securite-voyeur comblée*, Génération-NT [<https://www.generation-nt.com/actualites/camera-ip-trendnet-faille-securite-voyeur-1539071>].

4. Korben, 2013, *Les radars pédagogiques à la merci des pirates?* [<https://korben.info/les-radars-pedagogiques-a-la-merci-des-pirates/>].

5. Ouest-France with AFP, 2020, *Paris. He pirated the PMU and FDJ scratch cards in bars* [<https://www.ouest-france.fr/societe/faits-divers/paris-il-piratait-les-bornes-de-jeux-de-grattage-du-pmu-et-de-la-fdj-dans-les-bars-6949018>].

6. Fabien Soyez, 2013, *Vie privée : télé connectée, l'espion parfait*, CNET France [<https://www.cnetfrance.fr/news/vie-privee-tele-connectee-l-espion-parfait-39793195.html>].

7. Camille Kaelblen, 2016, *Is your connected fridge the ideal gateway for hackers?*, RTL [<https://www.rtl.fr/culture/futur/votre-frigo-connecte-est-il-la-porte-d-entree-ideale-pour-les-hackers-7785045780>].

8. Gilles Halais, 2012, *Un hacker a trouvé comment pirater à distance les pacemakers*, Franceinfo [https://www.francetvinfo.fr/sciences/un-hacker-a-trouve-comment-pirater-a-distance-les-pacemakers_1631785.html].

9. Sandrine Cassini, 2015, *Les jouets VTech victimes d'un piratage*, Le Monde [https://www.lemonde.fr/economie/article/2015/12/01/les-jouets-vtech-victimes-d-un-cybercriminel_4821275_3234.html].

10. Paul Ackermann, 2015, *Une voiture piratée à distance par des hackers*, HuffPost [https://www.huffingtonpost.fr/2015/07/22/voiture-pirate-distance-hackers_n_7846132.html].

11. Guillaume Ledit, 2017, *On Twitter, "Internet of Shit" ridicules the Internet of shitty... objects*, Usbek & Rica [<https://usbeketrica.com/article/sur-twitter-internet-of-shit-ridiculise-l-internet-des-objets-merdiques>].

12. Wikipedia, 2014, *Computer network* [https://fr.wikipedia.org/wiki/R%C3%A9seau_informatique].

13. A MAC address is a sequence of 12 hexadecimal digits (from 0 to 9, then a for 10, b for 11, and so on up to f for 15) such as 00:3a:1f:57:23:98.

opposite page

15

opposite page

15

page

20

page 215

26.1.3 Different types of links

The most common ways of connecting PCs to a network are either by cable, known as Ethernet, or by radio, known as *Wi-Fi*.



A standard RJ-45 Ethernet connector

But beyond our telephone socket, our communications on the Internet are transported by many other means. There are many different ways of transmitting information: copper cable, fiber optics, radio waves, *and so on*. From modem transmission¹⁴ in the 1990s to fiber optics¹⁵ used for intercontinental connections, via ADSL in the¹⁶ in the 2000s, each has different characteristics, particularly in terms of information throughput (also known as *bandwidth*) and installation and maintenance costs.

These different technologies do not have the same weaknesses when it comes to the confidentiality of the communications entrusted to them or the traces they leave behind: for example, it will be easier to intercept a wide-area radio signal from a distance than light passing through an optical fiber.

26.2 Communication protocols

For machines to talk to each other, they not only need to be connected, they also need to speak a common language. This language is called a *communications protocol*. Most of the "languages" used by machines on the Internet are precisely defined in public documents.¹⁷ This is what enables different networks, computers and software to work together, as long as they respect these standards. This is what is meant by *interoperability*.

Different protocols meet different needs: downloading a file, sending an e-mail, consulting a website, *and so on*.

For simplicity's sake, we will detail these different protocols below, classifying them into three categories: physical, network and application protocols.¹⁸

14. "Modem" is the condensed word for *modulator-demodulator*: it enables digital data to be transmitted on a channel that can carry sound, such as a telephone line.

15. An optical fiber is a wire made of transparent material that transmits data in the form of light pulses. This makes it possible to transmit large volumes of information, even over long distances.

16. ADSL (for *Asymmetric Digital Subscriber Line*) or VDSL (for *Very-high-bit-rate Digital Subscriber Line*) is a technology enabling digital data to be transmitted over a telephone line independently of the telephone service.

17. These public documents are known as *Request For Comments*. The Commentcamarche website explains the RFC concept very well. Jean-François Pillou, 2011, *Les RFC, CommentCaMarche* [<https://web.archive.org/web/20210219111153/https://www.commentcamarche.net/contents/533-les-rfc>].

18. In reality, it's a little more complicated. For more details see: *Wikipedia, 2017, Internet Protocol Suite* [https://fr.wikipedia.org/wiki/Suite_des_protocoles_Internet].

And what better way to get the point across than with an analogy?

Let's compare the journey of our information through the Internet to the routing of a postcard, whose stages, from postal sorting center to mailbox, would correspond to the different computers it passes through.

26.2.1 Physical protocols

To get our mail to its destination, we use a variety of means of transport: planes, boats, trucks and even bicycles.

Each of these means of transport is subject to a number of regulations: highway code, air traffic control, maritime law, *and so on*.

[previous
page.]

Similarly, on the Internet, the various hardware technologies described above imply the use of different conventions. These are known as *physical protocols*.

26.2.2 Network protocols

Knowing how to navigate is not sufficient to get our postcard to the recipient. You also need to know how to read a zip code and have a few notions of geography to reach the recipient, or at least the nearest sorting center.

This is where *network protocols* come into play: their aim is to enable the routing of information from one machine to another, sometimes far away, independently of the physical connections between these machines.

[page 202]

The best-known network protocol is IP. -----

26.2.3 Application protocols

[page 209]

The Internet is often used to access the Web, i.e. a set of pages accessible on servers, which can be consulted using a web browser: <https://guide.boum.org> is an example of a web site. Web applications use a protocol called *HTTP*, the encrypted and authenticated version of which is *HTTPS*. Common parlance often confuses the web with the Internet, with expressions like "going online", for example. But the web is just one of the many uses of the Internet.

In fact, there are a huge number of applications that use the Internet that most Internet users don't even realize they're using. In addition to the web, these include e-mail, instant messaging, file transfer, crypto-currencies *and more*.

This is how you'll come across these different protocols which, although they use the Internet, are *not* web protocols:

- *SMTP*, *POP* and *IMAP* are protocols used in electronic messaging. ¹⁹There are also encrypted and authenticated versions (*SMTPS*, *POPS*, *IMAPS*);
- *Skype*, *Signal*, *IRC* and *XMPP* are all protocols used for instant messaging;
- *BitTorrent* is a peer-to-peer file-sharing protocol.

In fact, anyone with a sufficient knowledge of programming can create a new protocol and therefore a new Internet application themselves.

[page 202
this page]

Each Internet application uses a particular language, called an *application protocol*, and then puts the result into "packets" which are transmitted by the Internet's network protocols. We can compare the application protocol to the language in -----

19. There is a notable difference in the protocols used, which has consequences in terms of confidentiality and anonymity, depending on whether you use a mailbox via your web browser (webmail) or via a mail client. More on this later [page 290].

to write the text of a postcard: both sender and recipient must understand this language. However, the Post Office doesn't need to understand anything, as long as the letter contains a valid address.

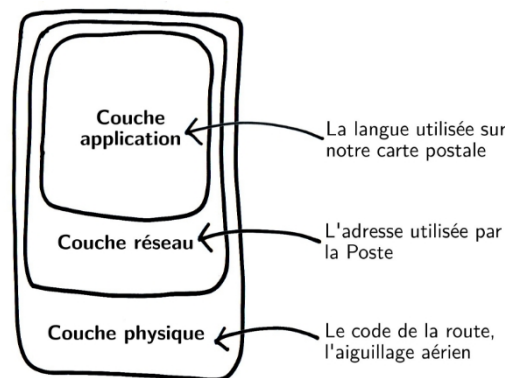
In general, postcards are not placed in envelopes: anyone on the road can read them. Similarly, the source and destination written in the packet header can be read by anyone. There are also many application protocols which are not encrypted: in this case, the contents of the packets can also be read by anyone. page 47 anyone.

Not all application protocols are transparent. While many of them are defined by open and accessible conventions (and thus verifiable by the personnes qui le souhaitent), some applications use proprietary protocols page 39 with little or no documentation. This makes it difficult to analyze any sensitive information contained in the data exchanged. For example, *Skype* works as a veritable black box, which does what you want it to (communicate), but possibly much more besides: in particular, it has been discovered that the content of messages is analyzed and possibly censored ²⁰ and that all web addresses sent *via* e-mail are forwarded to Microsoft ²¹.

26.2.4 Encapsulation

In reality, different protocols are used simultaneously during a communication, each with its own role in routing information.

It's common to represent these different protocols in overlapping layers.



Encapsulated protocols

In fact, when we communicate by mail, our communication is based on writing (in a certain language), then on delivery by the Post Office, which itself relies on different means of transport.

In a similar way, an Internet application will use a pre-established *application protocol*, will be routed using *network protocols*, and will traverse the various infrastructures respecting the *physical protocols* in force.

This is known as encapsulation: application protocols are encapsulated in network protocols, which in turn are encapsulated in physical protocols.

20. Ryan Gallagher, translated by Cécile Dehesdin, 2013, "Throwing eggs", "naughty cinema"... The list of words monitored by Skype in China, Slate.fr [<https://www.slate.fr/monde/69269/tom-skype-surveillance-chine-espionnage-liste-noire>].

21. Jürgen Schmidt, 2013, *Skype's ominous link checking : Facts and speculation*, The H [<http://www.h-online.com/security/features/Skype-s-ominous-link-checking-Facts-and-speculation-1865629.html>] (in English).

26.2.5 More on IP protocol

It's interesting to note that, unlike physical and application protocols, network protocols are relatively universal. Physical protocols evolve with technological advances, whether wired or wireless. Applied protocols evolve with the development of new applications: web, email, chat, *etc.* Between these two levels, to know which way to go and how to route our packets through the millions of Internet networks, since the 1980s everything has gone through the *Internet Protocol* (IP).

Packages

In the IP protocol, the information to be transmitted is broken down and packaged *into packets*, on which the sending and destination addresses are written. This "label" is called the packet *header*, and contains the information needed to route the packets to and from the destination. The packets of information are then transmitted independently of each other, sometimes using different paths, and reassembled once they reach their destination.

In addition to IP, there are two protocols: *TCP (Transmission Control Protocol)* and *UDP (User Datagram Protocol)*. TCP was designed to transmit packets without losing data, taking the time to check everything. UDP ensures the speed of exchanges without checking that packets arrive at their destination; it is used in particular for video and audio-conferencing.

IP address

For this to work, every computer connected to the network must have an address, which is used to send packets to it: the *IP address*. This address must be unique within a network. Indeed, if several computers on the network had the same address, the network wouldn't know which computer to send packets to.

An IP address can be compared to a telephone number: every telephone must have a telephone number if it is to be called. If several phones had the same number, there'd be a problem.

The addresses used since the early days of the Internet have taken the form of four numbers from 0 to 255, separated by a dot: these are known as IPv4 (*Internet Protocol version 4*) addresses. An IPv4 address looks like this `203.0.113.12`.

The IPv4 protocol was defined in the early 1980s and allows a maximum of 4 billion addresses to be allocated. At the time, it was unimaginable that the Internet would ever be accessible to the general public, and it was thought that 4 billion would be sufficient.

In the 1990s, in response to the looming address shortage, the IETF²² began work on IPv6 (*Internet Protocol version 6*). Since 2011 the shortage has been a reality, and it's difficult for new operators to obtain IPv4 addresses. The IPv6 protocol is therefore gradually being deployed by operators (even if there are some recalcitrants). Implementing IPv6 involves considerable political stakes²³ but also new security issues²⁴. In 2022, the two protocols (v4 and v6) will operate in parallel. An IPv6 address looks like this `2001:0db8:85a3:0000:0000:8a2e:0370:7334`.

22. Wikipedia, 2016, *Internet Engineering Task Force* [https://fr.wikipedia.org/wiki/Internet_Engineering_Task_Force].

23. In *this lecture* [<https://ldn-fai.net/intranet-ipv4-ou-internet-ipv6/>], LDN explains the challenges of switching to IPv6.

24. This new standard poses new problems for our online anonymity. Florent Fourcot, 2011, *IPv6 et conséquences sur l'anonymat*, LinuxFr.org [<https://linuxfr.org/users/ffourcot/journaux/ipv6-et-cons%C3%A9quences-sur-lanonymat>]. To be continued...

The IP address is an extremely useful piece of information for anyone seeking to monitor what's happening on a network, as it uniquely identifies a computer on the network at a given time, without being any real proof against a person (as one computer can be used by several people).²⁵ against an individual (as one computer may be used by several people). It can, however, indicate the geographical origin of a connection, provide clues, and initiate or confirm suspicions.

26.2.6 Port

Many applications can be used simultaneously from the same computer: reading e-mail in Thunderbird, looking at the SNCF website, chatting with friends via instant messaging, listening to music online. Each application must only receive packets intended for it and containing messages in a language it understands. Sometimes, however, a computer connected to the network has only one IP address. To this address, we add a number that enables the computer to forward the packet to the right application. We write this number on the packet, in addition to the address: it's the *port* number.

To understand, let's compare our computer to a building: the building has only one address, but houses many apartments, and many different people. The apartment number on an envelope is used to send mail to the right addressee. The same applies to port numbers: they are used to send data to the right application.

Certain port numbers are conventionally assigned to particular applications. So, when our web browser wants to connect to a web server, it knows to dial port 80 (or 443 in the case of an encrypted connection). Similarly, to deliver an e-mail, our computer will generally connect to the server's port 25 (or 465 if it's an encrypted connection).

page 209

On the computer we're using, every application connected to the Internet opens at least one port, whether it's a web browser, instant messaging software, music player, *etc.* So the number of ports opened as part of an Internet connection can be very high, and closing your web browser is often far from sufficient to cut off all connection to the network...



PRECISION

The more ports are open, the more points through which malicious persons or viruses can attempt to infiltrate a computer connected to the network. *Firewalls* usually leave only certain ports open, as defined in their configuration, and reject requests to other ports.

26.3 Local networks

You can network without the Internet. In fact, computer networks appeared long before the Internet. In the 1960s, network protocols such as HP-IB²⁶ which allowed only a limited number of computers to be connected, were already operating *local* networks.

25. Legalis, 2013, *L'adresse IP, preuve insuffisante de l'auteur d'une suppression de données sur Wikipedia* [<https://www.legalis.net/actualite/ladresse-ip-preuve-insuffisante-de-lauteur-dune-suppression-de-donnees-sur-wikipedia/>].

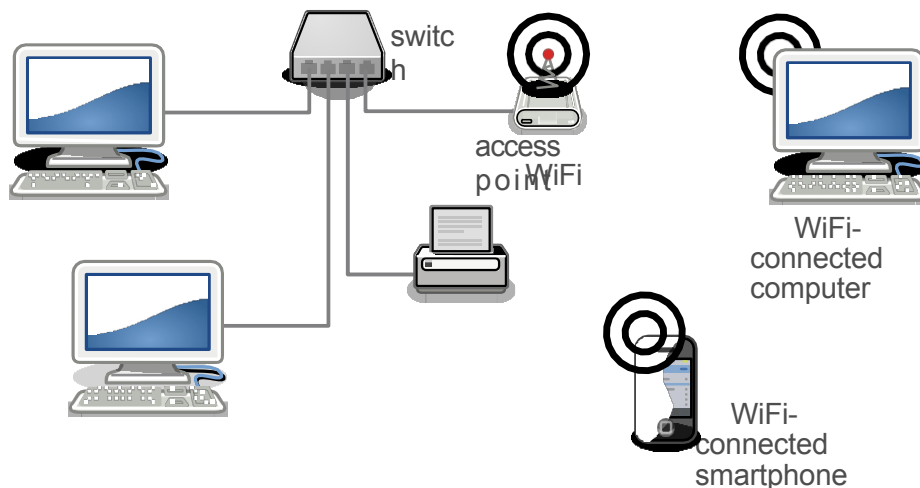
26. Wikipedia, 2014, *HP-IB* [<https://fr.wikipedia.org/wiki/HP-IB>].

26.3.1 The local network, the basic structure of the Internet

When you connect several computers together in the same home, school, university, office, building, *etc.*, you're talking about a *local area network* (LAN). Computers can then communicate with each other, for example, to exchange files, share a printer or play network games.

Local networks can be compared to the internal telephone networks of certain organizations (companies, universities, *etc.*).

These local networks are often made up of different devices that communicate with each other:



Local network diagram

26.3.2 Wi-Fi switch and access point

To link the machines that make up a local network, they are usually each connected to a network "power strip", either by cable or Wi-Fi. A "switch" is often used, and can indeed be compared to a power strip. However, instead of forwarding each incoming packet to all connected computers, a switch reads the address on the packet and sends it only to the right destination socket.

The equivalent of a switch in wired networks is called an "access point" in the wireless world. Each access point has a name, which is broadcast to the surrounding area (this is the list of Wi-Fi networks that our network software affichets).

To continue our comparison, the switch is a bit like the local postman, delivering mail to each recipient in the neighborhood. To do this, the switch processes information from the network cards, identified by their hardware address, plugged into each of its sockets.

Just as physical access to a machine opens up many possibilities for retrieving information, physical access to a network means that, unless there are special defenses in place, you can impersonate one of the other machines on the network. This makes it possible to gather a great deal of information on the communications circulating on the network, by setting up a monster-in-the-middle attack. Physical access to the network can be gained by connecting a cable to a switch, or *via* a Wi-Fi access point.

26.3.3 Addressing

To enable machines connected to the network to communicate with the IP protocol, they must each have an IP address. Software and protocols have been developed to automate the assignment of IP addresses to computers during the

network connection, such as DHCP protocols²⁷ in IPv4 or NDP²⁸ and SLAAC protocols in IPv6²⁹.

In order to function, the system needs to remember the association of a given network card, identified by its hardware address, with a given IP address. The correspondence between IP address and hardware address is only useful within this local network. There is therefore no technical reason for hardware addresses to circulate on the Internet, although this does happen from time to time.³⁰

page 198

26.3.4 NAT and reserved addresses for local networks

Internet standardization bodies realized in the 1990s that the number of IPv4 addresses available was not going to be sufficient to cope with the rapid growth of the network. In response to this problem, certain address ranges were reserved for private networks and are not used on the Internet: these are the *private addresses*³¹.

page 202

Thus, most Internet "boxes" assign the computers that connect to them addresses starting with 192.168³² in IPv4 and fe80: in IPv6. Several local networks can use the same private IP addresses, unlike IP addresses on the Internet, which must be unique worldwide.

Packets carrying these addresses cannot leave the private network unchanged. These private addresses are therefore only used on the local network. So, for example, a machine may have the IPv4 address 192.168.0.12 on the local network, but from the point of view of the other machines with which it communicates via the Internet, it will appear to be using the IPv4 address of the "box" (for example, 203.0.113.48): this will be its *public address*. It's the "box" that takes care of modifying packets accordingly, thanks to *Network Address Translation (NAT)*.

26.4 The Internet: interconnected networks

Internet stands for INTERconnected NETworks.

Each of these networks is called an *Autonomous System (AS)*.

26.4.1 Internet service providers

An *Internet Service Provider (ISP)* is an organization that offers a connection to the Internet, whether via optical fiber, electromagnetic waves³³ telephone line or coaxial cable. In France, the main commercial ISPs for domestic use are Bouygues, Orange, Free and SFR. There are also a number of associative ISPs, such as the members of the Fédération FDN.³⁴

Often, an ISP operates its own network, to which subscribers' "boxes" are connected.

27. Used in IPv4 networks, DHCP stands for *Dynamic Host Configuration Protocol* ().

28. Wikipedia, 2017, *Neighbor Discovery Protocol* [https://fr.wikipedia.org/wiki/Neighbor_Discovery_Protocol].

29. Wikipedia, 2022, *IPv6*, section "IPv6 address allocation" [https://fr.wikipedia.org/wiki/IPv6#Attribution_des_adresses_IPv6].

30. One of the ways in which the physical address circulates on the Internet is through the use of captive portals, which we will discuss later [page 216].

31. At the same time, the IETF was working on version 6 of the IP protocol [page 202], which solves the shortage problem.

32. Private address ranges are defined by convention in a document called "RFC 1918". In addition to addresses starting with 192.168, they include those starting with 10 and from 172.16 to 172.31.

33. Wi-Fi, 4G or other...

34. [The list of FDN Federation members](https://www.ffdn.org/fr/membres) [<https://www.ffdn.org/fr/membres>].

To connect a local network to other networks, you need a *router*. This is a computer whose role is to forward packets between two or more networks.

A "box" used to connect a home to the Internet acts as a router. It has a network card connected to the local network, but also an ADSL modem or a fiber port connected to the ISP's network: this is known as a router-modem. It's not only part of the local network, but also part of the Internet: in IPv4, it's the IP address of the "box" that is visible from the Internet on all the packets it carries for the computers on the local network. Conversely, with IPv6, all machines connected to the network have routed public addresses and are therefore part of the Internet.

The "box" is a small computer which, in the same casing as the modem-router, integrates software for managing the local network (such as DHCP software), as well as an Ethernet and/or Wi-Fi switch for connecting several computers, and sometimes a TV decoder, hard disk, etc. The box can also be used to connect to the Internet.

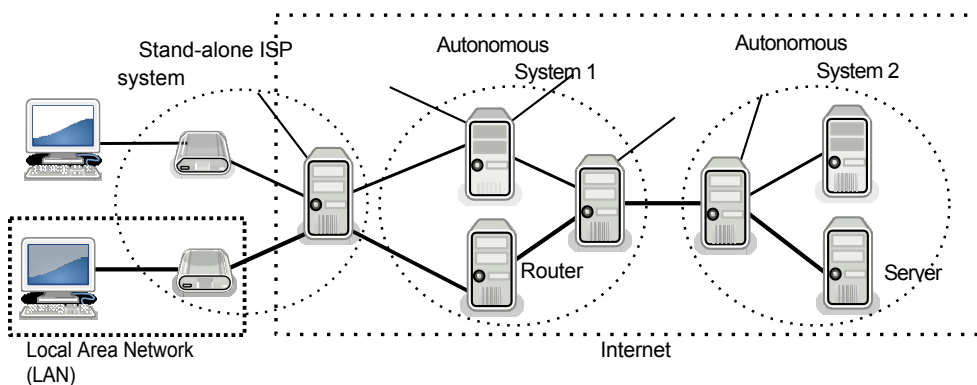
page 204

26.4.2 Autonomous systems

An autonomous system is a coherent network - usually under the control of a single entity or organization - capable of operating independently of other networks.

In 2022, the interconnection of over 72,000 AS worldwide ³⁵ form the Internet.

A stand-alone system can typically be the network of an Internet service provider (e.g. Free, SFR or *tetaneutral.net*). In this case, each "box" used to connect a local home network to the Internet is part of the provider's network, which in turn is interconnected with other stand-alone systems to form the Internet. Organizations that host Internet services (e.g. Gitoyen ³⁶Google or Riseup) and those who manage the "big pipes" - such as the transatlantic cables through which much of the Internet's data flows - also have their own autonomous systems.



The Internet is an interconnection of autonomous networks

The Internet, then, is not a large, homogeneous network managed centrally. Rather, it is made up of a multitude of interconnected networks managed by a wide variety of organizations and companies, each with its own way of operating.

All these networks, infrastructures and computers don't run themselves: they're managed on a daily basis by people called *systems and network administrators*,

35. Nice statistics on the evolution of AS can be found on [the CIDR Report website \[https://www.cidr-report.org/as2.0/\]](https://www.cidr-report.org/as2.0/).

36. Association providing services to [Globenet \[https://www.globenet.org/-Services-.html\]](https://www.globenet.org/-Services-.html), several FDN Federation members, and several [Chatons \[https://chatons.org/\]](https://chatons.org/). More information is available on [its website \[https://gitoyen.org/\]](https://gitoyen.org/).

"admins" or "admins" ³⁷. Admins are in charge of installing, maintaining and updating these machines, so they *necessarily* have access to a lot of information.

In terms of monitoring, the commercial interests and legal obligations of autonomous systems vary widely, depending on the country and the type of organization involved (institutions, companies, associations, *etc.*). No one has complete control over the Internet, and its global nature complicates any attempt at unified legislation. As a result, there is no uniformity of practice.

Network interconnection

Just as we've connected our local network to our ISP's autonomous system, the latter establishes connections to other networks. It is then possible to pass information from one autonomous system to another. It's thanks to these interconnections that we can communicate with the various computers that make up the Internet, regardless of which AS they belong to.



A router

A router is a computer that connects several networks. Operators have routers on all the time, and they look more like large pizza boxes than personal computers. However, their operating principle remains similar to that of other computers, and they are fitted with a few specialized circuits to switch packets very quickly from one network to another.

Autonomous systems agree to exchange traffic with each other, also known as *peering* agreements. In most cases, *peering* is free, and the exchange is balanced. To reach autonomous systems with which it has no *peering* agreement, an operator may use a transit provider. A transit provider is an operator who knows how to reach the whole Internet, and sells connectivity to other operators. ³⁸.



PRECISION

There is a principle that prohibits any discrimination in traffic, whether with regard to the source, destination or content of the information transmitted over the network. This principle is known as *net neutrality*. This principle guarantees that Internet users will not have to deal with any Internet traffic management that would have the effect of limiting their access to applications and services distributed over the network. For example, limiting online video viewing or downloading. Net neutrality ensures that information flows are not blocked, degraded or favored by telecommunications operators, enabling free use of the ³⁹ network. In France, the Quadrature du Net ⁴⁰ and the Fédération FDN ⁴¹ defend and promote net neutrality ⁴².

37. Later, we'll use the term "admins" to refer to system and network administrators.

38. Loïc Komol, 2013, *Le peering : petite cuisine entre géants du Net*, Clubic [<https://www.clubic.com/pro/it-business/article-558086-1-peering-petite-cuisine-geants-web.html>].

39. #DataGueule has made a video [<https://peertube.datagueule.tv/videos/watch/64077068-5d05-4815-9095-af63a33a91c4>] that clearly explains net neutrality and the associated political issues.

40. Net neutrality as seen by La Quadrature du Net [https://www.laquadrature.net/neutralite_du_net].

41. Founding principles of the Fédération FDN [<https://www.ffdn.org/fr/principes-fondateurs>].

42. Net neutrality is defined in French law in article L33-1 of the Code des postes et des communications électroniques [https://www.legifrance.gouv.fr/codes/article_lc/LEGIARTI000043545209/].

Interconnection points...

Network operators used to run cables directly between their routers, which meant a lot of cables, and a lot of expense. Now they use *interconnection points* (*IXs* or *IxPs*, for *Internet eXchange Points*), which are places where many stand-alone systems are linked together. Operators wishing to connect to these points each bring a fiber and install routers. Due to the sheer volume of traffic passing through these points, they are of great strategic importance to governments and other organizations wishing to monitor what passes through the network.⁴³

... interconnected

The major interconnection centers are linked by large bundles of optical fibers. Together, these links form the Internet's *backbones*.⁴⁴

For example, to link Europe to America, several fiber optic bundles run along the bottom of the Atlantic Ocean. These fiber bundles are all points of weakness, and from time to time an accident - such as a ship's anchor cutting a cable - can slow down the Internet on a continental scale.⁴⁵ This may seem strange, given that historically, the idea of the Internet was military-inspired: a decentralized network, multiplying links so as to be resistant to the cutting of any one of them.

26.4.3 Routing

We've seen that computers exchange information by putting it into packets.

[page 202]

Imagine two computers connected to the Internet on different networks and wanting to communicate. For example, Ana's computer in France connects to Bea's computer in Venezuela.

Ana's computer accesses the Internet via her "box", which is part of her ISP's network. Bea's computer, on the other hand, is part of her university's network.

The packet destined for Bea's computer will first arrive on Ana's ISP network. It will be forwarded to her ISP's router C, which acts as a sorting center. The router reads the address of Bea's computer on the packet, and has to decide who to forward the packet to, to get it closer to its destination. How is this choice made?

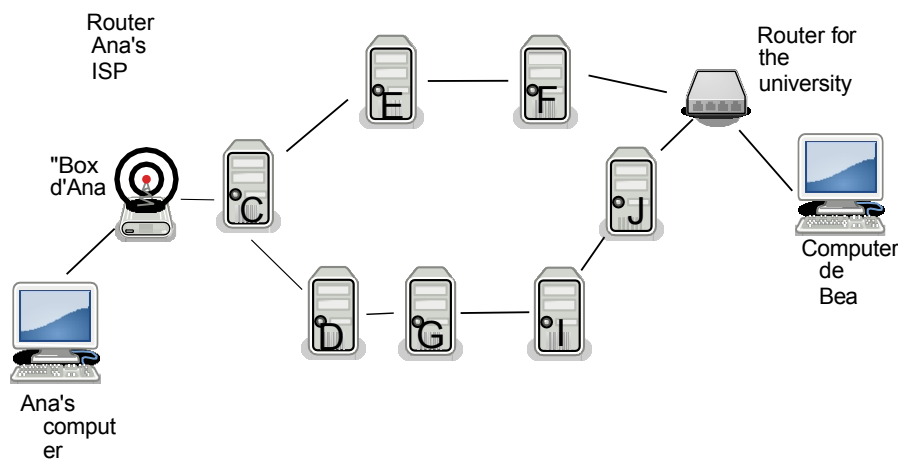
Each router maintains a list of the networks to which it is connected. It sends regular updates of this list to the other routers to which it is connected, its neighbors, who do the same. These lists enable the router to route incoming packets to their destination.

So Ana's ISP router knows that it can reach Bea's university network through four intermediaries by sending the packet to router D. But it can also

43. Guillaume Champeau, 2013, *How Germany also spies on our communications*, Numerama [<https://www.numerama.com/politique/26279-comment-l-allemande-aussi-espionne-nos-communications.html>].

44. TeleGeography, 2017, *Submarine Cable Map* [<https://www.submarinecablemap.com/>] (en English).

45. Pierre Col, 2009, *Internet, boat anchors and underwater earthquakes*, ZDNet [<https://www.zdnet.fr/blogs/infra-net/internet-les-ancres-de-bateaux-et-les-seismes-sous-marins-39602117.htm>], Cécile Dehesdin, 2013, *Des coupures dans des câbles sous-marins ralentissent Internet dans plusieurs pays*, Slate.fr [<https://www.slate.fr/monde/70063/cable-internet-sous-marin-coupe-impac-t-afrique-egypte>].



Routing

It will choose to send the packet to E, which has a more direct path.

The packet thus arrives at E, the router of a transit operator, an organization paid by Ana's ISP to route packets. E will do the same kind of calculation, and send the packet to F. F's network includes computers not only in Europe, but also in America, linked by a transatlantic cable. F belongs to a company, similar to the one running E, which is paid by Bea's university. F finally sends the packet to the university's router, which sends it to Bea's computer. Phew, our packet has arrived at its destination.

This means that every packet of information passing through the Internet passes through several networks. Each time, a router acts as a sorting center, forwarding it to a different router. In the end, each packet passes through many different computers, belonging to many different organizations.

What's more, network topology, i.e. the architecture, layout and hierarchy of individual workstations, changes over time.

When Ana connects to Bea's computer again the next day, the packets her computer sends won't necessarily take the same route as the day before. For example, if router E is switched off due to a power cut, Ana's ISP router will route the packets through D, which previously had a longer route.

The Egyptian government shut down the Internet during the 2011 revolution by taking action at the routing level. The routers of the country's main ISPs stopped telling other routers that they were the ones to route packets to Egyptian computers.⁴⁶ As a result, packets destined for Egypt could no longer find their way through, effectively interrupting access to the network - all without cutting a single cable.

26.5 Customers and servers

Historically, in the 1980s, every computer connected to the Internet provided a part of the Internet. Not only did it "go and see things on the Internet", but it also offered information, data and services to other users connected to the Internet: it *made* the Internet as much as it *accessed* it.

⁴⁶ Stéphane Bortzmeyer, 2011, *Internet outage in Egypt* [<https://www.bortzmeyer.org/egypte-coupure.html>].

The overall picture is very different today. As we've seen, there are computers that are permanently switched on and responsible for connecting bits of the Internet together: routers. Likewise, there's another category of permanently switched-on computers that contain almost all the data and services available on the Internet. These computers are called servers, because they *serve* information and services. They centralize most of the content, be it websites, music, e-mail, *etc., on the Internet*. This creates a vertical hierarchy in the network. Indeed, the more information you have, in the broadest sense, the more power you potentially have.

Servers provide, as opposed to clients who merely access information. This situation corresponds to an Internet where our machines act primarily as clients, centralizing the Internet around content providers.⁴⁷

Let's take the example of one of the services available on the Internet, [the Digital Self-Defense Guide website \[https://guide.boum.org/\]](https://guide.boum.org/): when Ana consults a page on this website, her computer acts as a *client*, connecting to the *server* hosting the Digital Self-Defense Guide.

That said, any computer can be both client and server, either at the same time or in succession. This is particularly true of the peer-to-peer, or *P2P*, model, widely used for file sharing. In this situation, each computer, otherwise known as a *node*, is connected to the network and communicates as both client and server. These two roles are not determined by the type of machine.

26.5.1 Name servers

When Ana asks her web browser to go to the Digital Self-Defense Guide site, her computer must connect to the server hosting the site.

[page 202]

To do this, you need to know the server's IP address. However, an IP address is a series of numbers that are rather difficult to memorize, type or transmit, such as 88.99.208.38 (for an IPv4 address). To solve this problem, there are servers that can be asked questions such as: "What is the IP address of *guide.boum.org*?", just as you would look up the number of a correspondent in the telephone book. This system is called DNS (*Domain Name System*). So Ana's computer begins, *via* its "box", by querying its Internet service provider's DNS server to obtain the IP address of the server hosting the *guide.boum.org* domain name.

Ana's computer receives the server's IP address and can communicate with it.

26.5.2 Web request path

Ana's computer then connects to the guide's server (88.99.208.38), and sends it a request that means: "Send me the home page of the *guide.boum.org* website." The packets carrying the request leave his computer and pass through his "box" to reach his ISP's router. They then cross several networks and routers (not shown in the diagram), before finally reaching the destination server.

[page 206]

[page 217]

26.5.3 Server software

In order to send Ana the requested web page, the server then searches for it in its memory, on its hard disk, or builds it.

47. Benjamin Bayart's talk *Internet libre, ou Minitel 2.0?* [<https://www.fdn.fr/actions/confs/internet-libre-ou-minitel-2-0/>], given at the 8^{es} rencontres mondiales du logiciel libre in Amiens in 2007, explains this shift and the issues involved very well.

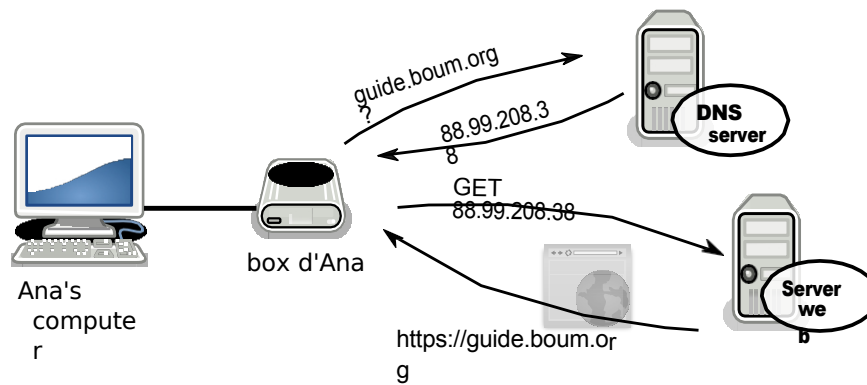


Diagram of a web request

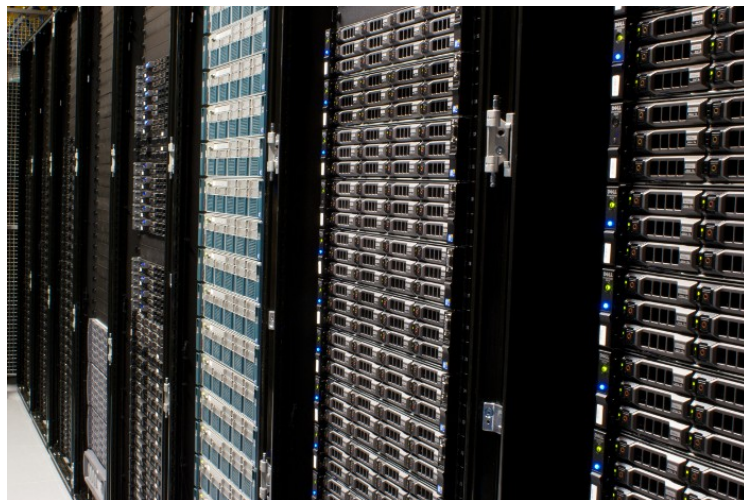
Pages that can be consulted on the web don't necessarily exist in a form that we can see on our computer *before* we request access to them. They are often generated automatically, on demand. These are known as *dynamic websites*, as opposed to *static* websites, whose pages are written in advance.

For example, if you search for "ouistiti moteur virtuose" in a search engine, it doesn't yet have the answer in reserve. The server then executes the source code on page 39 of the site to calculate the page containing the answer before sending it to us.

On the server, there's a piece of software that runs and responds to requests. This server software is specific to each application: it understands the application protocol. In the present example, this software searches for and serves the Ana computer's web page: we call it a *web server*.

26.5.4 Server hosting

Servers - the computers running the server software mentioned above - are usually housed in buildings with good network connections and reliable power supplies: *data centers*.



A server aisle in a data center

These days, it's all the rage to talk about *cloud computing*. This "marketing" concept does not call into question the separation between customers

and servers, quite the contrary. It simply means that data may be moved from one server to another, for legal, technical or economic reasons. And this without the owners necessarily being informed.



There is no cloud, only other people's computers



PRECISION

Google, for example, has at least twenty data centers on three continents⁴⁸ to ensure that its services are operational 24/7, even when certain equipment is unavailable.

Hosting providers of this type run hundreds of physical machines in several data centers around the world, pooling their storage and computing power into an abstract super-machine. They then sell "virtual machines", i.e. shares of this super-machine's computing and storage power. The Amazon Elastic Compute Cloud (or EC2) is one of the best-known services in this field.⁴⁹

A virtual machine can be moved automatically depending on the use of physical machines, the quality of their network connection, *etc.* With such an infrastructure, it's impossible to know in advance which physical machine - and therefore precisely where - a given virtual machine is located.

In practice, this makes it impossible to control our data.⁵⁰ Will they really be erased from physical machines if we "delete" them? We saw in the first volume that deleting data from a computer is a complicated business. This problem becomes even more acute if we don't know which computer we're talking about. What's more, this poses legal problems: data that is legal in one place may end up being illegal because the machine that contains it or serves it on the Internet has changed jurisdiction.

So there's been a shift from an Internet where everyone consulted and distributed data, to a model where data was centralized on physical machines called servers, and then today to the *cloud*, where the same data can be stored, sometimes scattered, on indeterminate servers. It becomes extremely complicated to know where the data is actually stored, and the user has even less control over what happens to it.

48. Google, 2017, *Data center locations* [<https://www.google.com/about/datacenters/inside/locations/index.html>].

49. Wikipedia, 2014, *Amazon Elastic Compute Cloud* [https://fr.wikipedia.org/wiki/Amazon_Elastic_Compute_Cloud].

50. Jos Poortvliet, 2011, *openSUSE and ownCloud* [<https://news.opensuse.org/2011/12/20/opensuse-and-owncloud/>].

Traces all along the line

Normal network operation means that many computers can see what you're doing on them. We're not talking about active surveillance here. It's just that sometimes it's completely necessary. Sometimes, however, this information is collected because it's "more convenient", for example, to diagnose problems.

The operation of any computer leaves a certain number of traces.

This is the theme of the first volume of this guide.

[page 27]

In the case of online use, it's not just the computer in front of your eyes that can keep track of what you're doing on the network, but also each of the computers through which the information transits. Many of these

information circulates *in clear text*, not encrypted.

[page 47]

27.1 On the client computer

The computer used to connect to the network is called the client. This machine knows everything you do with it, and often keeps a record of it.

[page 209]

As explained at length in the first volume of this guide, these traces, page 27 and the ease with which they can be exploited, depend very largely on the computer and operating system used.

27.1.1 Web browser memory

To make them more user-friendly, web browsers record a great deal of information about the pages you visit. Here are a few examples:

- Most web browsers keep a history of web pages consulted.
- They also often offer to record what the surfer enters in the forms found on certain web pages, as well as the passwords to various online accounts.
- In general, they also save recently or frequently consulted pages to speed up loading: this is known as "caching".¹

All this data is stored, enabling the police (among others) to trace our surfing habits. Let's remember our story from the beginning:

[page 194]

- Apparently, colleagues ended up finding the document on a certain Ana's workstation. It was downloaded from

1. To view the cache contents of the Firefox web browser or Tor Browser, type `about:cache` in the address bar.

web browser, and modified. There would have been a connection to a Gmail mailbox, as well as another mail address, this time at no-log, shortly before the publication of the incriminating documents.

27.1.2 Cookies

The word "cookie" comes from the English word "*fortune cookie*", referring to cakes that hide a message on a small piece of paper. A "cookie" is a small piece of text sent by a web site, which the user's browser stores and then sends back to the site each time the user visits. This is what enables webmail applications or commercial sites, for example, to remember that you are authenticated with your address and password during your session, or to memorize the language you wish to use.

Cookies also enable a website to track the people who visit it.

Internet advertising agencies include "tracer" cookies in the ads they affich on sites, enabling them to track the Internet user's movements on all sites that affich ads from the same advertising agency. In this way, they can "collect increasingly precise information about her and consequently offer her increasingly well-targeted advertising."²

What's more, when web pages are consulted, they establish connections to advertising sites, and often to the same sites, which further increases the possibility of tracking by these sites.

Lastly, some cookies have an expiry date, while others are of indefinite duration.

- the sites that pass them on to us will be able to identify our web browser for years to come!

Conventional cookies, however, are limited in terms of data volume, and easy to delete by an informed user. They have therefore been "improved", for example with the "local web storage" feature³ included in the HTML5 standard, which enables several megabytes of data to be stored in the web browser.

Other tracking-enhancing techniques involve storing the same cookie in different places in the web browser, and recreating any deleted cookies on each visit (on the assumption that if each cookie can be deleted, they will not all be deleted at the same time...)⁴.

Accepting the use of cookies therefore has consequences in terms of tracking, and leaves traces on our computer and on servers.

27.1.3 Client-side applications

In the evolution of the web and its browsers, it quickly became clear that to have a minimum of interactivity, it was necessary for part of the website's source code to be executed on the client side, by the web browser, and not on the web server hosting the site.

This has several practical aspects: on the web server side, it means less work and savings on hardware. On the client side, affichage and website functionality are accelerated. It also minimizes network traffic between the browser and the website: there's no need to request a full page of the website every time a small button is clicked, only a small fragment of the page needs to be transmitted.

2. CNIL, *La publicité ciblée en ligne* [<https://www.cnil.fr/fr/publicite-ciblee-en-ligne-quels-enj-pour-la-protection-des-donnees-personnelles>].

3. Wikipedia, 2020, *Local web storage* [https://fr.wikipedia.org/wiki/Stockage_web_local].

4. The *evercookie* JavaScript library [<https://samy.pl/evercookie/>] is an example of this type of technology.

Technologies have been added to web browsers to enable these functions: JavaScript and Java are the main representatives.

But these little extras also come at a cost: as mentioned above, this means that the author of a site is able to execute the code of her choice on the computers of other users.

(which poses a number of security problems, as we saw on page 32 in the first volume of this guide). Web browsers do, of course, have protections in place.⁵ browsers, but they don't cover all the risks, and are no substitute for vigilance on the part of Internet users.

All the more so as these technologies sometimes have functionalities that, while useful, raise questions: for example, WebRTC ⁶a technology designed to integrate real-time communications into web browsers, allows access to the microphone and camera of the computer on which it is used.

As we've seen, putting your trust in software is a complex choice. And page 39 the execution of such programs raises questions about the power given to the authors of web sites or applications to access our computer's resources, and the information it contains.

What's more, before being executed by the web browser, these code snippets pass through the network, often without any authentication. This leaves them free to be modified by well-placed malicious persons, just like the rest of a web page.

To introduce malware, for example. It is also possible to play ^{page 31} with the data these codes have to process, in an attempt to divert their use. This

This kind of manipulation of web pages was detected in the past when a hotel in New York was using a Wi-Fi access point equipped with a network dedicated to this task⁷.

Ultimately, a modern web browser has so many features that potential adversaries have a considerable number of angles of attack.

27.1.4 In software logs

Web browsers aren't the only programs that record traces on the computer used; most software programs have logs. ^{page 29}

For example, instant messaging software often records the history of conversations; peer-to-peer file-sharing software (such as BitTorrent), too, tends to remember what you've downloaded recently; e-mail software keeps track of the e-mails you've downloaded; *and so on*.

- Apparently, colleagues ended up finding the document on a certain Ana's workstation. It was downloaded from the web browser and modified.

In our story, the cops were able to trace Bea's document in the history of Ana's computer's web browser and word processor.

27.2 On the box: network card hardware address

We've seen that the network card used by every computer to connect to the network has a hardware address, or MAC address. This address is used by ^{page 198}

5. This generally involves giving access to website code only to limited functions by running it in a "sandbox" (Wikipedia, 2014, *Sandbox (computer security)* [[https://fr.wikipedia.org/wiki/Sandbox_\(s%C3%A9curit%C3%A9_informatique\)](https://fr.wikipedia.org/wiki/Sandbox_(s%C3%A9curit%C3%A9_informatique))]).

6. In Tor Browser, WebRTC functionality is disabled.

7. Justin Watt, 2012, *Hotel Wifi JavaScript Injection* [<https://justinsomnia.org/2012/04/hotel-wifi-javascript-injection/>].

networks to redirect a data packet to the right network card, when several computers are connected to the same "box" for example.

Normally, this address does not leave the local network. However, we usually connect directly to the "box" of an Internet service provider - if we're using the connection sharing of a telephone, it will act as the "box". Each network card connected to the box gives it its own hardware address.

[page 29] Most "boxes" keep a *log* containing these hardware addresses, at least for as long as they are switched on. It's difficult to know the types and quantity of information contained in this log, as well as the potential existence of backdoors⁸ or security loopholes. Indeed, these "Boxes run on software installed by the ISP, which retains privileged access, if only to update the software.

[page 22] For example, Orange admits to collecting, for 12 months, the physical addresses of computers connected to its "boxes", and the associated IP addresses for "diagnostics management".⁹ For us, the "box" is a veritable black box, to which we don't have the keys, and which can know (and do) a great deal about the local network.



TO FIND OUT MORE...

If you like to tinker, you can replace your ISP's modem router with a modem router running OpenWrt¹⁰ or, more simply, add an OpenWrt router between your ISP's box and your computer. Pre-installed routers are available, and some ISP associations provide their members with routers running only free software.¹¹

What's more, when the local network includes the use of Wi-Fi, the hardware addresses of computers connecting to the "In this way, Google Cars, at the same time as they travelled through thousands of streets to create the Google Street View map, took advantage of the opportunity to record their "box" in Wi-Fi. This is how the Google Cars, as well as covering thousands of streets to create the Google Street View map, took the opportunity to "capture MAC addresses of surrounding computers"¹².

On the other hand, it is possible to temporarily change the hardware address of a network card, so as not to be tracked, for example, by our laptops¹³ on our travels.

We should also mention the cases where, before being able to connect to the Internet, you have to enter a *login* and password in your web browser: this is often the case on public Wi-Fi networks, whether those of a town, an institution or an Internet service provider (*FreeWifi*, *SFR WiFi public* and other *Bouygues Telecom Wi-Fi*). These pages are known as *captive portals*. In this case, in addition to the hardware address of the Wi-Fi card, the organization managing the portal is given the identity of the subscriber corresponding to these identifiers.

8. An example of a backdoor on a manufacturer's routers [<https://korben.info/backdo-les-routeurs-d-link.html>].

9. Orange, 2021, *Diagnostics management* [https://web.archive.org/web/20210510112139/https://assistance.orange.fr/ordinateurs-peripheriques/installer-et-utiliser/la-securite/risques-et-prevention/les-donnees-personnelles/gestion-des-diagnostiques_195036-739979#onglet2].

10. OpenWrt is a free operating system for routers. Here are a few reasons to use it ser [https://openwrt.org/fr/reasons_to_use_openwrt].

11. A list of modems and routers used by FDN Federation members: *FDN Federation, 2017, Modems and routers* [<https://www.ffdn.org/wiki/doku.php?id=modems-routeurs>].

12. Europe 1 with AFP, 2011, *Street View : la Cnil épingle Google* [<https://www.europe1.fr/economie/Street-View-la-Cnil-epingle-Google-309338>].

13. Wikipedia, 2014, *Mac Spoofing* [https://fr.wikipedia.org/wiki/Filtrage_par_adresse_MAC#MAC_Spoofing].

27.3 On routers: packet headers

On the path between a computer and the server you want to connect to, there are numerous routers, which relay packets and send them to the right place.

page 206

To know where to send a packet, these routers read a kind of envelope on which a certain amount of information is written; this "envelope" is called the packet *header*.

The header of a packet contains a great deal of information needed for routing, including the IP address of the destination machine, as well as the public IP of the sender (to whom the reply should be sent). The router can therefore see which computer wants to talk to which computer, just as the postman needs to know the recipient's address in order to forward the mail, as well as the sender's address for a possible return.

page 202

The headers also contain the source and destination port numbers, which can provide information on the application used.

page 203

To do their job, routers *need* to read this information; they *can* also keep track of it in logs.

Although they have no good reason to do so, routers are also able to access the inside of the envelope being transported; for example, the content of the web page consulted by the surfer or that of an e-mail sent: this is known as Deep Packet Inspection (DPI).¹⁴ This is known as *Deep Packet Inspection* (DPI).

French Internet service provider Orange, for example, includes a clause in its subscriber contracts concerning the use of traffic "data".¹⁵

27.4 On the server

Like routers, the server hosting the visited site has access to the headers of IP packets, and thus to all the information we've just been talking about. In particular, it looks at the IP address of the "box" used by the connecting computer, to know who to send the reply to.

page 202

In addition to the IP headers, which correspond to the network layer of the communication, the server will read the application protocol headers, which correspond to the application layer.

page 200

page 200

But the server also reads the contents of the packets themselves: in fact, it's the server that has to open the envelope and read the letter in order to reply. The server software then interprets the letter received, which is written using the application protocol, to provide the appropriate response.

However, many application protocols also carry information that identifies the connecting computer - as we'll see in detail here.

Like client computers, servers have system logs - we'll talk more about these in the next section.

page 225

27.4.1 HTTP headers

When a web browser requests a web page, it includes in the request the name of the software, its version number, the operating system used and the language in which it is configured.

14. Wikipedia, 2021, *Deep Packet Inspection* [https://fr.wikipedia.org/wiki/Deep_packet_inspection].

15. Martin Untersinger, 2016, *Fin de l'Internet illimité : ça se précise chez Orange, qui dément* [<https://www.nouvelobs.com/rue89/rue89-internet/20121011.RUE3086/fin-de-l-internet-illimite-ca-se-precise-chez-orange-qui-dement.html>].

Here is a request sent by the Firefox web browser:

```
GET / index. php
HTTP/2 Host:
User-Agent: Mozilla /5.0 ( X11 ; Linux x86_64 ; rv:91.5) Gecko
/20100101 Firefox
Accept: text/ html, application/ xhtml+xml, application/ xml;q=0.9,
image/webp, */*;
Accept - Language: fr- FR, en;q=0.5
Accept - Encoding: gzip, deflate,
br Referer: https:// duckduckgo.
com/
Cookie: donation - identifier:
dd634367a6b4485ba288197bd92745b4
```

First we see a command containing the name of the requested page (/index.php), the corresponding domain name (example.org), followed by a header containing, among other things, the name and version of the web browser (Mozilla/5.0 (X11; Linux x86_64; rv:91.5) Gecko/20100101 Firefox/91.5) as well as the operating system used (Linux x86_64), the languages supported (fr-FR for French from France, en for English), the page on which the link was located that the surfer followed to arrive at the requested page (https://duckduckgo.com/), and the session cookie (donation-identifier: dd634367a6b4485ba288197bd92745b4).

[page 214]



TO FIND OUT MORE...

In the Firefox web browser, we can afficher in a few clicks the headers of our requests :

- click on top right ;
- select *Additional Tools* then *Web Development Tools*, then select the *Network* tab.

A new panel opens. When a page is loaded, a line affiche for each request. Select one - the first, for example, which corresponds to the loading of the page itself - to afficher its headers in the right-hand panel, in a tab named *Headers*.

This information is used by the web server, which adapts its response accordingly: this is how, for example, a site available in several languages is displayed in our language without our having to specify it.

But this information, like all that passes through the server, is also accessible to the people who maintain the server: its admins... and their hierarchy. In general, servers also keep this information in logs, for varying lengths of time, notably for statistical purposes and to facilitate diagnostics in the event of a breakdown. They add to the headers the originating IP address as well as the date and time. Here's a log line recorded for our request (the original IP address is at the beginning: 203.0.113.42):

```
203.0.113.42 - - [22/ Jan /2022:00:00:00 +0100] " GET / index. php
HTTP /2" 200 2131 " https:// duckduckgo. com/" " Mozilla /5.0
(X11 Linux x86_64 ; rv:91.5) Gecko /20100101 Firefox /91.5)
Gecko /20100101 Firefox
/91.5"
```

27.4.2 Mail headers

Every e-mail includes a header; despite its name, this has nothing in common with the header of a web page. This header contains information about the data contained in the e-mail: another example of metadata, the "don-

[page

on the data". It's rarely shown in its entirety by our e-mail software, but it's there nonetheless. It often includes a great deal of information about the sender - much more than just her e-mail address.

In the following example, we can read the public IP address, i.e. the one that will be visible on the Internet, of the computer used to send the e-mail (203.0.113.98), which tells us where the sender was at the time, the IP address of her computer within her local network (192.168.0.10), the e-mail software used (Thunderbird/91.5.0), or even her operating system (Mac OS X 11).

page 205

```
Return - Path: <bea@fai.net >
Delivered - To: ana@exemple.org
Received: from smtp. fai. net ( smtp. fai. net [198.51.100.67])
        by mail. exemple. org ( Postfix) with ESMTTP id 0123456789
        for <ana@exemple.org >; Sat , 22 Jan 2022 20:00:00 +0100 (
        CET)
Received: from [192.168.0.10] ( paris. abo. fai. net
        [203.0.113.98]) by smtp. fai. net ( Postfix) with ESMTTP
        id ABCDEF1234 ;
        Sat , 22 Jan 2022 19:59:49 +0100 ( CET)
Message - ID: <CB0ABB91 .17 B7F@fai.net
> Date: Sat , 22 Jan 2022 19:59:45 +0100
From: Bea <bea@fai.net >
User - Agent: Mozilla /5.0 ( Macintosh; Intel Mac OS X 11;
        rv:91.5) Gecko /20100101 Thunderbird /91.5.0
MIME - Version: 1.0
To: Ana <ana@exemple.org >
Subject: See you Tuesday
Content - Type: text/ plain; charset=iso -8859 -1
Content - Length: 22536
Lines: 543
```

These headers sometimes also contain the subscriber's ID at her mail service provider or the name of her machine.¹⁶

Like these few common examples, virtually all en- see not only content information, but also metadata in their proto page 30 cole.

27.5 The traces we leave behind

It's not just the traces left by the way networks operate: there are also those we leave ourselves, more or less voluntarily, for example by entering information on websites or simply connecting to services.

Attempting to control the traces we leave on the networks therefore also means thinking about the uses we make of the services offered on the Internet, and the data we entrust to them - topics we'll be dealing with in more detail in the sections to come.

¹⁶. Most of the time, this is found in the Received line of the first machine or in the Message-ID. But some other software or messaging services add other, more specific lines.

Monitoring and control of communications

Beyond the traces left by the very operation of networks in general and the Internet in particular, it is possible to "listen in" to our activities on the Internet at several levels.

More and more often, the organizations that run parts of the Internet (cables, servers, *etc.*) are even legally obliged to retain a certain amount of data on what's happening on their machines, under *data retention* laws.

page 224

28.1 Who wants the data back?

A variety of people and organizations can be prying eyes on Internet exchanges. Parents who are a little too curious, websites looking for customers to target, multinationals like Microsoft, the police in Saint-Tropez, or the US *National Security Agency*...

As in the case of bugs on personal computers, the various Page 31 entities involved do not necessarily work together, nor do they form a coherent whole. If the curious are too varied to claim to draw up an exhaustive list of the interests at stake, we can nevertheless describe some of the most common motivations.

28.1.1 Companies looking for profiles to resell

"You decide to book a plane ticket to New York on the Internet. Two days later, while reading your online newspaper, an advertisement suggests an interesting offer for a car rental in New York. This is no mere coincidence: it's a targeted advertising mechanism, as is currently being developed more and more on the Internet."¹

Advertising is one of the main sources of revenue for companies that provide "free" services on the Internet: mailboxes, search engines, social media, *etc.* But from the advertiser's point of view, the quality and therefore the price of online advertising space depends on the interest of Internet users. However, from the advertisers' point of view, the quality and therefore the price of online advertising space depends on the interest that Internet users will show in the ads.

That's why personal data is worth its weight in gold. Interests, gender, age, *etc.* This is the kind of information that enables us to present ads to which Internet users are most likely to respond. This is how Google cross-references the results of its

1. CNIL, 2009, *Targeted online advertising* [https://web.archive.org/web/20170830003226/https://www.cnil.fr/sites/default/files/typo/document/Publicite_Ciblee_rapport_VD.pdf].

personal activities² on all its services, such as the search engine, YouTube videos watched or photos in Google Photos to afficher targeted ads on its other applications, such as Gmail³.

page 214

What's more, each site visited is another "center of interest". When you add up all this information, a whole profile emerges⁴. A small piece of software enables you to see which cookies are downloaded to your computer with each page you visit. If you start by visiting allocine.fr, four advertising agencies record your visit. If they then go on to the Le Monde site, four advertising agencies will be aware of this, two of which were already on the AlloCiné site. They therefore know that the user has visited these two sites, and can cross-reference these two centers of interest. By subsequently visiting two other sites (Gmail and Dailymotion), a total of twenty-one advertising agencies became aware of the surfer's visit. Each of these visits included XiTi and Google-Analytics. As a result, the world's largest search engine has been informed of all the sites visited, and can now implement targeted advertising.

Social media are particularly well placed to obtain personal data directly from users. On Facebook, for example, a company can "target an advert at 13 to 15 year olds living in Birmingham, England, who have 'drinking' as their center of interest". What's more, Facebook indicates that the chosen target comprises approximately one hundred people⁵. In this way, Facebook exploits the data it collects from its members to provide advertising that can be highly targeted.⁶

Targeted advertising is, in fact, "one of the reasons why Internet players have diversified their services and activities, in order to gather ever more information about user behavior on the Internet." "For example, Google provides search services. It has bought advertising companies like DoubleClick. It has [...] launched a Google Suggest service, integrated into its Chrome browser, which sends Google all the web pages visited by Internet users, even when the latter have not accessed them *via* the search engine, *etc.*"⁷

To give you an idea of the stakes involved, Google bought Doubleclick for \$3.1 billion.⁸

This accumulation and processing of data also enables Google to sort and adapt results to the supposed interests of the Internet user. So, for an identical search, two people with different profiles won't get the same result, which has the effect of reinforcing each person's interests.

2. Julien Lausson, 2017, *Why Google won't stop targeted advertising and scanning your emails on Gmail*, Numerama [<https://www.numerama.com/tech/270293-pourquoi-google-ne-va-pas-arreter-la-publicite-ciblee-et-le-scan-de-vos-mails-sur-gmail.html>]

3. "When you open Gmail, you see ads selected according to their usefulness and relevance. The process of selecting and afficher personalized ads in Gmail is fully automated. These ads are presented to you based on your online activity while logged into Google. We do not analyze or read your Gmail messages to choose which ads are shown to you." Google, 2021, *How ads work in Gmail* [<https://support.google.com/mail/answer/6603?hl=fr>].

4. Data Gueule, 2014, *Big data: data, data, give me! - #DATAGUEULE 15* [<http://peertube.datagueule.tv/w/etMw3qxMsdZHcvhFzekvie>].

5. A similar interface is publicly available and helps answer some disturbing queries: Tom Scott, 2014, *Actual Facebook Graph Searches* [<https://actualfacebookgraphsearches.tumblr.com/>].

6. CNIL, 2009, *Targeted online advertising* [https://web.archive.org/web/20170830003226/https://www.cnil.fr/sites/default/files/typo/document/Publicite_Ciblee_rapport_VD.pdf], p. 13.

7. CNIL, 2009, *Targeted online advertising* [https://web.archive.org/web/20170830003226/https://www.cnil.fr/sites/default/files/typo/document/Publicite_Ciblee_rapport_VD.pdf], p. 4.

8. Le Monde, 2007, *Google buys DoubleClick for \$3.1 billion* [https://www.lemonde.fr/technologies/article/2007/04/14/google-rachete-doubleclick-pour-3-1-milliards-de-dollars_96316_651865.html].

interests and convictions. This is what some people call "the individualization of the Internet"⁹.

In addition to being thematically targeted, advertising is also geographically targeted: thanks to the GPS integrated into mobile terminals such as smartphones, but also thanks to the IP address and Wi-Fi networks "visible" within range of the laptop or phone.¹⁰ This makes it possible, for example, to display advertisements for stores located close to the subscriber.

Economic interests drive Internet service providers to gather profiles of Internet users, as precise as possible, in order to then sell, directly or indirectly, targeted advertising space.

Once this information has been collected, companies will be able to respond to requests from the cops. All the big content providers have offices dedicated to responding to requests, and so have forms, procedures, *etc.*, written out for the cops, explaining the best way to go about requesting information.¹¹

28.1.2 Companies and governments seeking to protect their interests

Other companies take an interest in what's happening on the Internet to protect their interests. This ranges from the audiovisual industry's fight against illegal downloading, to technology watch: companies observe and analyze hundreds of sources (news sites, patent registration databases, expert blogs, *etc.*) in real time and on an automated basis, in order to keep abreast of the latest technological advances and remain as competitive as possible.

Companies are far from the only ones scrutinizing the Internet. Governments, from the justice system to secret services and police forces, are certainly the most curious.

More and more countries are introducing laws to make it possible to identify the authors of any information circulating on the Internet.¹²

But it goes even further. Intelligence agencies and other secret services are no longer content with spying on a few groups or individuals they consider to be targets. On the fringes of legality, the NSA, the US intelligence agency, collects "all kinds of data on people - we think it would involve millions of people".¹³ Among its objectives: "to examine 'virtually everything an individual does on the Internet'".¹⁴ and to establish a *social graph*, i.e. "the network of connections and relationships between people".¹⁵ "In general, they analyze networks located two degrees of separation from the target." Otherwise

9. Xavier de la Porte, 2011, *Le risque de l'individualisation de l'Internet*, InternetActu.net, Fondation Internet nouvelle génération [https://web.archive.org/web/20210413221428/https://www.internetactu.net/2011/06/13/le-risque-de-l'individualisation-de-l'internet/].

10. Audenard, 2013, *Bornes wifi et smartphones dans les magasins*, blogs/sécurité, Orange Business [https://www.orange-business.com/fr/blogs/securite/mobilite/souriez-vous-etes-pistes-merci-aux-bornes-wifi-des-magasins].

11. Several versions of the guide published by Facebook have been leaked [https://publicintelligence.net/facebook-law-enforcement-subpoena-guides/] in recent years. Several other similar guides (not all of which are accurate) can be found on cryptome.org [https://cryptome.org/isp-spy/online-spying.htm].

12. Begeek, 2013, *Facebook publishes its first international report of government requests* [https://www.begeek.fr/facebook-publie-premier-rapport-international-demandes-gouvernementales-102351].

13. Bruce Schneier, quoted in Guillaud, 2013, *Lutter contre la surveillance : armer les contre-pouvoirs*, Internet Actu [https://web.archive.org/web/20220126013621/https://www.internetactu.net/2013/06/13/lutter-contre-la-surveillance-arter-les-contre-pouvoirs/].

14. Maxime Vaudano, 2013, *Plongée dans la "pieuvre" de la cybersurveillance de la NSA*, Le Monde.fr [https://www.lemonde.fr/technologies/visuel/2013/08/27/plongee-dans-la-pieuvre-de-la-cybersurveillance-de-la-nsa_3467057_651865.html].

15. Pisani, 2007, *Facebook/5: the recipe* [https://www.francispisani.net/facebook5-la-recette/].

says, the NSA also spies on those who communicate with those who communicate with those who are being spied on".¹⁶.

French intelligence services now have an arsenal of laws at their disposal, enabling them to carry out analyses on all Internet traffic or on targeted individuals in complete legality, in France¹⁷ or abroad¹⁸.

28.2 Logs and data retention

Most organizations that provide services over the Internet (connection, site hosting, *etc.*) keep more or less trace of what passes through their machines, in the form of connection logs: who did what, when. We call these *logs*.

Historically, these logs have served a technical purpose: they are used by server maintainers to diagnose and resolve problems. However, they can also be very useful for gathering data on the users of these servers.

page

29

28.2.1 Data retention laws

In most Western countries, Internet service providers are now legally obliged to keep their logs for a certain period of time, in order to be able to respond to requisitions.

The laws governing data retention define more or less clearly the information that must be kept in these logs. The notion of Internet service provider can thus be understood in a fairly broad sense¹⁹ A cybercafé is an Internet service provider that *also* supplies a machine to access the network.

Over and above their legal obligations, many Internet service providers store varying amounts of information about the Internet users who use their services, particularly for targeted advertising. GAFAMs such as Google, Amazon and Facebook are particularly well known for this. As this The "ad-supported service delivery model" has virtually become the norm.²⁰ It's safe to assume that many others are doing the same, more discreetly.

In the UK, an Internet Service Provider (ISP) caused controversy when it emerged that it was keeping track of all the web pages visited by its subscribers in order to test a profiling technology designed to "offer" "behavioral advertising".^{21 22}

The server hosting the content used (web page, mailbox, *etc.*) and the Internet service provider are particularly well placed to have the information needed to identify the originator of a connection request. In France, they are particularly targeted by data retention laws.

16. Manach, 2013, *Pourquoi la NSA espionne aussi votre papa (#oupas)*, Bug Brother [https://bugbrother.blog.lemonde.fr/2013/06/30/pourquoi-la-nsa-espionne-aussi-votre-papa-oupas/].

17. Légifrance, *Code de la sécurité intérieure*, articles L851-2 et L851-3 [https://www.legifrance.gouv.fr/codes/section_lc/LEGITEXT000025503132/LEGISCTA000030935576].

18. Légifrance, *Code de la sécurité intérieure*, article L854-1 [https://www.legifrance.gouv.fr/cod/es/article_lc/LEGIARTI000037200982/].

19. CNIL, 2010, *Conservation des données de trafic : hot-spots wi-fi, cybercafés, employeurs, quelles obligations ?* [https://www.cnil.fr/fr/conservation-des-donnees-de-traffic-hot-spots-wi-fi-cybercafes-employers-what-obligations].

20. CNIL, 2009, *Targeted online advertising* [https://web.archive.org/web/20170830003226/https://www.cnil.fr/sites/default/files/typo/document/Publicite_Ciblee_rapport_VD.pdf], p. 4.

21. CNIL, 2009, *Targeted online advertising* [https://web.archive.org/web/20170830003226/https://www.cnil.fr/sites/default/files/typo/document/Publicite_Ciblee_rapport_VD.pdf], p. 17.

22. Arnaud Devillard, 2009, *Affaire Phorm : Bruxelles demande des comptes au Royaume-Uni* [https://www.01net.com/actualites/affaire-phorm-bruxelles-demande-des-comptes-au-royaume-uni-501173.html].

28.2.2 Logs kept by hosting providers

We've seen that the server hosting a service (such as a website, mailbox or instant messaging room) has access to a large amount of data.

page 217

In France, article 6 of the Law for Confidence in the Digital Economy (LCEN)²³ (LCEN), which obliges hosts of public content to retain "data likely to enable the identification" of "any person who has contributed to the creation of content posted online"; for example, writing on a social media site, a blog or a participative media site, or posting on a public mailing list.²⁴

For content that constitutes private correspondence, article L34-1 of the French Post and Electronic Communications Code²⁵ (CPCE) which imposes the same obligation for writing an e-mail or sending an instant message, for example. In concrete terms, this means keeping any identifiers or pseudonyms supplied by the author for one year, but above all the IP address at the source of the connection each time the content is modified.²⁶ A request to the Internet Service Provider (ISP) supplying this IP address can then generally be traced back to the owner of the connection used.

page 202

In addition, *the law on military programming*²⁷ promulgated at the end of December 2013, makes it possible to request this same information, in real time, for reasons as varied as: terrorist attacks, cyber-attacks, attacks on scientific and technical potential, organized crime, etc.

It is this obligation to retain data that enables the police, in our introductory story, to obtain information from the organizations hosting the incriminated e-mail addresses:

page 194

- We're going to ask Gmail and no-log for information on these email addresses. Then we'll probably have something to go on, or at least something to ask the right questions!

Hosting providers can be more or less cooperative in verifying the legality of subpoenas sent to them by the cops, and in responding to them.

23. Légifrance, 2022, Article 6 de la loi n° 2004575 du 21 juin 2004 pour la confiance dans l'économie numérique

[https://www.legifrance.gouv.fr/loda/article_lc/LEGIARTI000045292730].

24. Légifrance, 2021, Decree no. 2021-1362 of October 20, 2021 on data retention enabling the identification of any person who has contributed to the creation of content posted online [<https://www.legifrance.gouv.fr/loda/id/JORFTEXT000044228912>].

25. Légifrance, 2013, Article L34-1 - code des postes et des communications électroniques [https://www.legifrance.gouv.fr/codes/article_lc/LEGIARTI000028345210/].

26. "Natural or legal persons who provide, even free of charge, for disposition of the public by services of communication to the public on line, the storage of signals, writings, images, sounds or messages of any nature provided by recipients of these services" (LCEN, *op. cit.*, article 6 paragraph 1.2 [https://www.legifrance.gouv.fr/loda/article_lc/LEGIARTI000045292730]), i.e. hosting companies, are obliged to keep for one year and for each content creation, modification or deletion operation: " a) The identifier of the connection at the origin of the communication; b) The types of protocols used to connect to the service and transfer content" (Article 5 of Decree no. 2021-1362 of October 20, 2021, *op. cit.* [https://www.legifrance.gouv.fr/loda/article_lc/LEGIARTI000044230063]).

But also: " a) The identifier assigned by the information system to the content, object of the operation; b) The nature of the operation; c) The date and time of the operation; d) The identifier used by the author of the operation when provided by the author" (article 6 of decree no. 2021-1362 of October 20, 2021, *op. cit.* [https://www.legifrance.gouv.fr/loda/article_lc/LEGIARTI000044230065]), in view of article 1 of decree no. 2021-1363 of October 20, 2021, ordering, in view of the serious and current threat to national security, the retention for a period of one year of certain categories of connection data [https://www.legifrance.gouv.fr/loda/article_lc/LEGIARTI000044231713]).

27. Légifrance, 2014, Law no. 2013-1168 of December 18, 2013 on military programming. for the years 2014 to 2019 and various provisions concerning national defense and security [<https://www.legifrance.gouv.fr/affichTexte.do?cidTexte=JORFTEXT000028338825&dateTexte&categorieLien=id>].

some respond to a simple email from the cops, while others will wait for a signed letter from a judge²⁸ or even fail to respond to requests²⁹.

Not only can people with access to the server collaborate with the cops voluntarily, but adversaries can also, as in the case of a personal computer, break in and spy on what's going on there by using loopholes, without going through the requisition stage. They will then have access to all data stored on the server, including logs.

But the server doesn't always know the real identity of Internet users who connect to it: generally, all it can give is an IP address.

That's where the ISP comes in.

28.2.3 Logs kept by Internet service providers

page 205

We've seen that the Internet is accessed via an Internet Service Provider (ISP). This ISP is generally a company that provides a "box" connected to the Internet. But it can also be an association or a public institution (a university, for example, when you use their computer rooms). ISPs are also subject to data retention laws.

Within the European Union, a directive obliges Internet service providers to keep track of who has logged on, when and from where.³⁰ In practice, this means recording which IP address has been assigned to which subscriber for which period of time.³¹ Institutions that provide access to the Internet, such as libraries and universities, do the same: generally, you log in with a user name and password. This makes it possible to find out who was using which workstation at what time. The European directive stipulates that this data must be kept for between 6 months and 2 years. In France, the legal period is one year.³²

Counter-intuitively, this obligation applies to all places offering Internet access to the public, whether for a fee or free of charge, even if the users are not identified. Bar managers who ignored this provision paid the price and found themselves in police custody for offering Wi-Fi to their customers without keeping the connection data.³³

28. Globenet, 2014, *No-log, logs and the law* [<https://www.globenet.org/No-log-les-logs-et-la-loi.html>].

29. "It should be noted that the servers hosting the Indymedia network sites, domiciled in the USA at Seattle, systematically refuse to disclose to the authorities the connection *logs* of computers consulting these sites or posting a contribution, thus rendering the authors of the contributions unidentifiable" (judicial investigation file cited by Anonymes, 2010, *Analyse d'un dossier antiterroriste* [https://infokiosques.net/spip.php?page=lire&id_article=789]).

30. European Parliament and Council, 2006, *Directive 2006/24/EC of the European Parliament and of the Council. Council of 15 March 2006 on the retention of data generated or processed in connection with the provision of publicly available electronic communications services or of public networks and amending Directive 2002/58/EC* [<https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32006L0024:EN:HTML>], known as "Data Retention".

31. "Persons whose business is to provide access to public communication services. en ligne" (LCEN, *op. cit.*), i.e. ISPs, are required to keep for one year: " a) The connection identifier; b) The identifier assigned by these persons to the subscriber; c) The IP address assigned to the source of the connection and the associated port" (article 5 of decree no. 2021-1362 of October 20, 2021, *op. cit.* [https://www.legifrance.gouv.fr/loda/article_lc/LEGIARTI000044230063]).

But also: " a) The date and time of the start and end of the connection; b) The characteristics of the connection of the subscriber's line" (article 6 of decree no. 2021-1362 of October 20, 2021, *op. cit.* [https://www.legifrance.gouv.fr/loda/article_lc/LEGIARTI000044230065]), in light of article 1 of decree no. 2021-1363 of October 20, 2021, *op. cit.* [https://www.legifrance.gouv.fr/loda/article_lc/LEGIARTI000044231713]).

32. Légifrance, 2021, Décret n° 2021-1362 du 20 octobre 2021 relatif à la conservation des données permettant d'identifier toute personne ayant contribué à la création d'un contenu mis en ligne [<https://www.legifrance.gouv.fr/jorf/id/JORFTEXT000044228912>].

33. Sputnik France, 2020, *Les gérants de bars en garde-à-vue pour avoir-offert-du-wifi-a-leurs-clients-a-grenoble/* [https://fr.sputniknews.com/faits_divers/202009291044498557-des-gerants-de-bars-en-garde-a-vue-pour-avoir-offert-du-wifi-a-leurs-clients-a-grenoble/].

In addition, French ISPs and hosting companies are required to retain "information relating to the civil identity of the user" for five years following the end of the user's contract.³⁴ They must also retain "other information provided by the user when subscribing to a contract or creating an account".³⁵ and "payment information [...] for each payment transaction".³⁶ for a period of one year after the end of the validity of the contract or the closure of the account.

The aim of data retention laws is therefore to make it easy for the authorities to associate a name with any action taken on the Internet.

Cops investigating an article published on a blog, for example, can ask the server hosting the blog for the IP address of the person who posted the article, along with the corresponding date and time. Once they have this information, they can ask the ISP responsible for the IP address to whom it was assigned at the time of the incident.

- *What a story! But what's it got to do with our offices?*
- *Well, that's also why I'm calling you. They affirm that they have all the evidence that these documents were published from your offices. I told them it wasn't me, that I didn't know what they were talking about.*

This is exactly what we're talking about when, in our story at the beginning, the police claim, with supporting evidence, that the bank statements were mailed from the Rue Jaurès offices. They have first obtained the IP address of the connection responsible for publishing the incriminating documents from the site's hosts. This first step makes it possible to determine from which "box" the connection originates. A request to the Internet Service Provider (ISP) reveals the subscriber's name - a bonus address - *via* its contract, associated with the IP address.

page 194

28.2.4 VPN, a story of trust

VPN (*Virtual Private Network*) is a system initially created to share a private network between several sites.³⁷ It creates a direct link between our computer and the server of the chosen VPN provider. A VPN enables you to change the IP addresses of your Internet connection: for the routers and servers you connect to, the connection no longer comes from the ISP's "box", but from the VPN server. This can help bypass certain types of censorship.

34. "1° The surname and first name, date and place of birth or company name, as well as the surname and first name, date and place of birth of the person acting on its behalf when the account is opened in the name of a legal entity; 2° The associated postal address(es); 3° The e-mail address(es) of the user and of the associated account(s), if any; 4° The telephone number(s)." (Article 2 of Decree no. 2021-1362 of October 20, 2021 on the conservation of personal data) from data, *op. cit.* [https://www.legifrance.gouv.fr/loda/article_lc/LEGIARTI000044230081]).

35. "1° The identifier used; 2° The pseudonym(s) used; 3° Data intended to per- enable the user to check or change his or her password, if necessary by means of a dual user identification system, in their last updated version." (Article 3 of Decree no. 2021-1362 of October 20, 2021 on data retention, *op. cit.* [https://www.legifrance.gouv.fr/loda/article_lc/LEGIARTI000044230083]).

36. "1° Type of payment used; 2° Payment reference; 3° Amount; 4° Date, the time and place in the event of a physical transaction". (Article 4 of Decree no. 2021-1362 of October 20, 2021 on data retention, *op. cit.* [https://www.legifrance.gouv.fr/loda/article_lc/LEGIARTI000044230085]).

37. Some companies use shared document storage on their local network. The VPN enables encrypted, authenticated connection to the company's local network for access to shared storage.

Some VPN services relay the traffic of many people with just a few IP addresses. This makes it possible to blend in with the mass of people using the VPN service, and complicates identification.

Data can be encrypted for the ISP, but remains visible to the VPN provider. VPN admins always have access to both the source and destination of communications. Using a VPN simply shifts the problem from trusting the ISP to trusting the VPN.

Although it may be considered in some threat models, the use of VPNs is not developed in this guide. If you want to blend into a mass of Internet users and be easily identifiable, without depending on trust in a single intermediary, it's safer to use Tor, as suggested below.

28.2.5 Requisition

In France, when the cops want to access the logs provided for in data retention laws, they're supposed to go through a *judicial requisition*: an official request that obliges the people administering a server to provide them with the requested information... or disobey. These requisitions are supposed to specify the information requested and be legally founded. But they are not always, and Internet service providers sometimes provide information that the law does not oblige them to provide.

Here is an extract from a requisition received by a French e-mail host. The e-mail address of the account in question has been anonymized by replacing the identifier with the *adresse*. The spelling has not been changed.



JUDICIAL REQUISITION

Lieutenant de Police On duty at B.R.D.P

Let us pray and, if necessary, request :

Monsieur le président de l'association GLOBENET 21ter, rue Voltaire
75011 Paris

for the purpose of :

About the *adresse@no-log.org* e-mail address

- Provide us with **the full identity** (surname, first name, date of birth, parentage) and **contact details** (postal, telephone, electronic and bank) of the **owner of the account**.
- Give us the last THIRTY connection data (IP address, date, time and time zone) used to **consult, read or send messages** with the said address (Pop, Imap or Webmail).
- Indicate whether a **redirection is active** on this mailbox, and provide us with the destination e-mail(s), if applicable.
- Give us the **phone number of** the no-log.org account "*adresse*" and the **last 30 login details**.
- Send us the **last THIRTY connection data** (IP address, date, time and time zone) to the **administration pages** of the no-log account "*adresse*".

What's more, it's a fact that cops sometimes ask for such information in a simple e-mail, and it's likely that many Internet services providers respond directly to such officious requests, which implies

that *anyone* can obtain such information by posing as the police.

Requisitions are commonplace. The big ISPs now have dedicated legal departments to deal with them, and a fee schedule encrypts each type of request.³⁸ Since October 2013, in France, a government-approved fee schedule has even homogenized these different services.³⁹ For example, the cost of identifying a subscriber based on her IP address was €4 (rates in force in October 2013). For more than 20 requests, this rate is reduced to 18 centimes.

In the first half of 2020, for example, Google received an average of 1,349 requests per month from France for information on its female users, concerning a total of 10,864 accounts - figures that have been rising steadily since 2009. After analyzing the legal admissibility of the requests, the company responded to 60% of them.⁴⁰ The other half of the requests did not fall within the scope of what the company considered itself legally obliged to provide.

In addition to connection logs, since the 2016 law⁴¹ against organized crime, the authorities can take cognizance of the content of stored correspondence⁴² upon simple request.

28.3 Mass listening

In addition to the logs and requisitions provided for in data retention laws, Internet communications are systematically monitored by various state services.

A former employee of U.S. telecommunications operator AT&T has testified⁴³ that the NSA (the US electronic intelligence agency) was monitoring all Internet and telephone communications passing through a major AT&T telecommunications facility in San Francisco. This was done using a supercomputer specially designed for real-time mass surveillance of communications.⁴⁴ He also stated that such installations probably existed in similar infrastructures in other U.S. cities, as confirmed by the revelations of a former NSA and CIA employee.⁴⁵ Similar installations are said to have been set up by the British secret services on more than 200 undersea optical fibers.⁴⁶

38. Christopher Soghoian, 2010, *Your ISP and the Government: Best Friends Forever* [<https://www.defcon.org/html/defcon-18/dc-18-speakers.html#Soghoian>].

39. Légifrance, 2013, *arrêté du 21 août 2013 pris en application des articles R. 213-1 et R. 213-2 of the French Code of Criminal Procedure setting the fees applicable to requisitions issued by electronic communications operators* [<https://www.legifrance.gouv.fr/affichTexte.do?cidTexte=JORFTE XT000028051025>].

40. Google, 2021, *France - Google Information Transparency* [<https://www.google.com/transparencyreport/userdatarequests/EN/>].

41. Law no. 2016-731 (*op. cit.*)

42. Légifrance, 2019, *Article n° 706-95-1 du code de procédure pénale* [https://www.legifrance.gouv.fr/codes/article_1c/LEGIARTI000038311668]; This article thus makes it possible to circumvent the constraints of a search and not alert the person concerned to this invasion of his or her privacy.

43. Mark Klein, 2004, *AT&T's Implementation of NSA Spying on American Citizens* [<https://www-tc.pbs.org/wgbh/pages/frontline/homefront/etc/kleindoc.pdf>].

44. Reflets.info, 2011, *#OpSyria : BlueCoat master craftsman of Syrian censorship* [<https://web.archive.org/web/20160823002531/https://reflets.info/opsyria-bluecoat-maitre-artisan-de-la-censu-re-syrienne/>].

45. Craig Timberg and Barton Gellman, 2013, *NSA paying U.S. companies for access to communications networks* [https://www.washingtonpost.com/world/national-security/nsa-paying-us-companies-for-access-to-communications-networks/2013/08/29/5641a4b6-10c2-11e3-bdf6-e4fc677d94a1_story.html] (in English).

46. L'expansion.com, 2013, *"Operation Tempora": how the British are outperforming the Americans to spy on the Internet* [https://www.lexpress.fr/economie/high-tech/operation-tempora-comment-les-britanniques-depassent-les-americains-pour-espionner-internet_1434134.html].

French security services are now authorized to install such traffic analysis tools in ISPs' networks in order to "detect connections likely to reveal a terrorist threat".⁴⁷

Since the Finance Act 2020⁴⁸ the French tax and customs authorities have been authorized to use certain personal data automatically. They can in fact collect freely accessible information from social media used to "put several parties in touch with a view to selling a good, providing a service or exchanging or sharing content, a good or a service", i.e. platforms such as Le Bon Coin or BlaBlaCar.

The NSA has also gained direct access to the servers of several Internet "giants" (Microsoft, Yahoo, Google, Facebook, PalTalk, Youtube, Skype, AOL and Apple).⁴⁹ which enables it to access the data they host or that passes through their servers.⁵⁰ The DGSE, the French equivalent of the NSA, has direct access to Orange's networks.⁵¹

Similarly, satellite communications are listened in on by the Echelon network, a "global system for intercepting private and public communications" developed by Anglo-American countries.⁵² developed by Anglo-Saxon countries⁵³. Information on this subject remains unclear, but France also appears to have a telecommunications monitoring network on its territory.⁵⁴

The NSA is also monitoring and cross-checking email exchanges to map the relationships between all US residents.⁵⁵ While such practices are not necessarily documented elsewhere in the world, they are just as possible.

What's more, for any organization with the means to be a significant network node, whether officially or not, the use of *Deep Packet Inspection* (or *DPI*) is becoming widespread. *DPI* surveillance is particularly intrusive: it is no longer limited to the information contained in the headers of IP packets, but touches the very content of communications. If these are not encrypted, it is possible to retrieve, for example, the complete content of e-mails, or the entirety of our web searches.

The use of this technique in Libya and Syria, for example, enabled the entire population of the country to be placed under digital surveillance, and then targeted attacks to be carried out. With the help and support of the French government, Amesys, a French-based company⁵⁶ from

47. Légifrance, *Code de la sécurité intérieure*, article L851-3 [https://www.legifrance.gouv.fr/cod/es/article_lc/LEGIARTI000043887520/].

48. Légifrance, 2019, *LOI n° 2019-1479 du 28 décembre 2019 de finances pour 2020* [https://www.legifrance.gouv.fr/jorf/article_jo/JORFARTI000039684091/].

49. NSA, 2013, *Dates When PRISM Collection Began For Each Provider* [https://commons.wikimedia.org/wiki/File:Prism_slide_5.jpg].

50. Le Monde, 2013, *Le FBI aurait accès aux serveurs de Google, Facebook, Microsoft, Yahoo! et d'autres géants d'Internet* [https://www.lemonde.fr/ameriques/article/2013/06/07/le-fbi-a-acces-aux-serveurs-des-geants-d-internet_3425810_3222.html].

51. Jacques Follorou, 2015, *Espionnage : comment Orange et les services secrets coopèrent*, Le Monde [https://www.lemonde.fr/international/article/2014/03/20/dgse-orange-des-liaisons-inces-tueuses_4386264_3210.html].

52. Wikipedia, 2021, *Echelon* [<https://fr.wikipedia.org/wiki/Echelon>].

53. Gerhard Schmid, 2001, *Report on the existence of a global interception system for private and economic communications (ECHELON interception system)* [https://www.europarl.europa.eu/doceo/document/A-5-2001-0264_EN.html].

54. Wikipedia, 2021, *Frenchelon* [<https://fr.wikipedia.org/wiki/Frenchelon>].

55. Gorman, Siobhan, 2008, *NSA's Domestic Spying Grows As Agency Sweeps Up Data: Terror Fight Blurs Line Over Domain; Tracking Email* [<https://www.wsj.com/articles/SB120511973377523845>].

56. kitetoo, 2011, *Amesys : le gouvernement (schizophrène) français a validé l'exportation vers la Libye de matériel d'écoute massive des individus*, Reflets.info [<https://web.archive.org/web/20181121190456/https://reflets.info/articles/amesys-le-gouvernement-francais-a-valide-l-exportation-vers-la-libye-de-materiel-de-surveillance>].

page 217

page 217

next page.

at the time, installed such systems in Libya⁵⁷ Morocco, Qatar⁵⁸ and France⁵⁹.

28.4 Targeted attacks

When an Internet user or a resource available *via the* Internet - such as a website or mailbox - arouses the curiosity of adversaries, they can set up targeted attacks. These targeted attacks can take place at various levels: the directories that enable the resource to be found, the servers that host it, the clients that access it, *and so on*. We look at these different possibilities in this section.

In France, Internet service providers are required by law to block access to websites that have been placed on a "blocked list" following a court ruling⁶⁰ or considered by the *office central de lutte contre la criminalité liée aux technologies de l'information et de la communication* to contain child pornography, to provoke "directly acts of terrorism" or to "glorify" such acts.⁶¹ In addition, an ordinance obliges them to do the same for websites infringing copyright or related rights.⁶²

In October 2011, the Tribunal de Grande Instance de Paris ordered seven French ISPs to block "by IP or DNS" the website copwatchnord-idf.org⁶³ the site was accused of insulting and defamatory comments, and of collecting personal data on police officers. In February 2012, the court ordered the blocking of one of the 35 mirror sites⁶⁴ that the Ministry of the Interior was seeking to block.⁶⁵

On the other hand, the court did not order the blocking of the 34 other mirrors referenced by the Ministry of the Interior, as the latter "did not indicate whether or not it had attempted to identify their publishers and hosts", nor that of any mirror sites that might appear.

More recently, in 2019, the Tribunal de Grande Instance de Paris ordered Bouygues Télécom, Free, Orange and SFR to prevent access to the Sci-Hub and LibGen websites on the grounds of infringement of copyright or neighboring rights.⁶⁶ These sites provide free access to scientific articles that would otherwise be kept behind a paywall by their academic publishers. Blocking

57. Fabrice Epelboin, 2011, *Kadhafi espionait sa population avec l'aide de la France* [<https://web.archive.org/web/20150629233215/https://reflets.info/kadhafi-espionait-sa-population-avec-l%E2%80%99aide-de-la-france/>].

58. Reflets.info, 2011, *Qatar: Amesys' Finger held high* [<https://web.archive.org/web/20200923032548/https://reflets.info/articles/qatar-le-finger-tendu-bien-haut-d-amesys>].

59. Jean Marc Manach, 2011, *Amesys also monitors France* [<https://web.archive.org/web/20171011205936/http://owni.fr/2011/10/18/amesys-surveillance-france-takieddine-libye-eagle-dga-dgse-bull/>].

60. LCEN, *op. cit.* article 6-3 [https://www.legifrance.gouv.fr/loda/article_lc/LEGIARTI000043969099], created by Journal Officiel de la République Française, 2021, *loi n° 2021-1109 du 24 août 2021 confortant le respect des principes de la République* [<https://www.legifrance.gouv.fr/jorf/id/JORFARTI000043964844>].

61. Légifrance, 2015, *décret n° 2015-125 du 5 février 2015 relatif au blocage des sites provoquant to acts of terrorism and sites disseminating pornographic images and representations of minors* [<https://www.legifrance.gouv.fr/loda/id/JORFTEXT000030195477>].

62. Légifrance, 2020, article L336-2 du code de la propriété intellectuelle modifié par *Ordonnance n° 2019-738 du 17 juillet 2019* [<https://www.legifrance.gouv.fr/codes/id/LEGIARTI000033688218>].

63. Tribunal de grande instance de Paris, 2011, *Summary judgment of October 14, 2011* [https://data.over-blog-kiwi.com/1/13/34/21/20140707/ob_2fbf9e_jugement-tgi-paris-14-octobre-2011-gu.a.pdf].

64. A mirror site is an exact copy of another website.

65. Legalis, 2012, *ordonnance de référé rendue le 10 février 2012* [https://www.legalis.net/jurisp_rudences/tribunal-de-grande-instance-de-paris-ordonnance-de-refere-10-fevrier-2012/].

66. Numerama, 2019, *Sci-Hub and LibGen fight for the free dissemination of scientific knowledge: France orders their blocking* [<https://www.numerama.com/sciences/477218-sci-hub-et-libgen-lutent-pour-la-diffusion-gratuite-du-savoir-scientifique-la-france-ordonne-leur-blocage.html>].

covering 57 domains was imposed for one year. It is noteworthy that ISPs did not wish to oppose this censorship measure. In 2021, a new ruling extended the list to 278 domains.⁶⁷

28.4.1 Block access to the resource provider

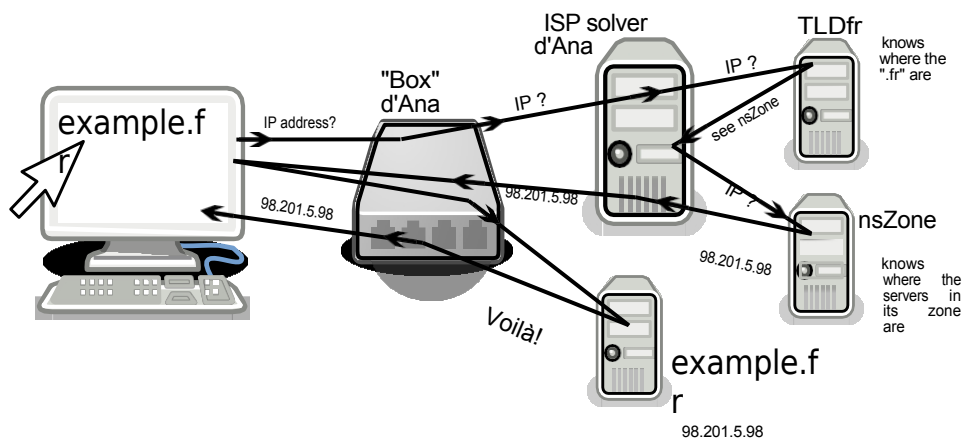
Let's take a look at the different ways of blocking access to a resource on the Internet.

Attack on domain names

It's possible to divert traffic that was supposed to go to a certain server by modifying the directory used to switch from its domain name to its IP address, i.e. DNS.

page 210

This can be done at different levels.



The key stages of a DNS query

Organizations managing the domain name directory For reasons of efficiency and robustness, the directory system (DNS) is managed by various organizations, in a hierarchical, distributed information system. The global DNS database is thus distributed among several name servers, each of which maintains only a part of the database.

Some organizations or companies are responsible for the DNS of so-called *Top Level Domains* (TLDs), which correspond to the characters after the last dot in the domain name, such as .com, .fr, .org, etc. All domains ending in .fr are under the responsibility of the AFNIC name server. All domains ending in .fr come under the name server of AFNIC, an association created for this purpose in 1997. Domains ending in .com, on the other hand, are managed by Verisign, a US corporation listed on the stock exchange.

The list of organizations and companies responsible for managing TLDs can be found on the IANA website⁶⁸ (Internet Assigned Numbers Authority), which manages the DNS root servers, the one that has authority over all the others.

While TLD managers have a purely technical role (keeping an up-to-date list of the domains in their care), those to whom they delegate are generally commercial companies (called *registrars*) who sell domain names.

A map of the nerve centers where censorship can intervene is now taking shape.

67. The Sound Of Science, 2021, [Exclusive] Why major French ISPs are blocking Sci-hub and Libgen again [https://web.archive.org/web/20221226013526/https://www.soundofscience.fr/2724].

68. IANA, 2014, *Root Zone Database* [https://www.iana.org/domains/root/db].



TO FIND OUT MORE...

So, renting a domain name is a separate operation from having an IP: for example, to set up your own website, you'll need to buy a domain name on the one hand, and find hosting for the site on the other, with an IP address attached to it. And then set up the link between the two. Some companies offer all these services at the same time, but it's neither systematic nor compulsory.

Domain name seizure The most spectacular domain name seizure to date was certainly that registered in connection with the closure of the file-hosting site megaupload.com by the U.S. Department of Justice. To make the site's services inaccessible, the FBI asked Verisign, the company that manages .com domain names, to modify its mapping tables so that the address no longer pointed to Megaupload's servers, but to an FBI server indicating that the site had been seized.⁶⁹

However, one of the first known domain name suspension censorship occurred, in 2007, at a registrar : GoDaddy (the world's largest). In the context of a dispute between one of its customers, seclists.org, and another site, myspace.com, GoDaddy sided with the latter and modified its database, making the site unreachable overnight and without notifying anyone⁷⁰ (except for those who know its IP address by heart).

Lying DNS Finally, while modifying global directories is within the reach of only a few states and companies, many can simply falsify their own version of the directory: this is known as "lying DNS". For example, each ISP generally has its own domain name servers (DNS), which are used by default by its subscribers.

When a domain name server provides something other than what has been registered with the registrars, this is also known as "lying DNS".⁷¹ a violation of net neutrality.

page 207

This is the level at which the administrative blocking of sites operates in France: ISPs must modify their directories to redirect addresses listed by the *office central de lutte contre la criminalité liée aux technologies de l'information et de la communication* (central office for combating crime linked to information and communication technologies).

to a Ministry of the Interior page⁷².

People using the Orange ISP were able to experience this blocking in spite of themselves on October 17, 2016. Following a "human error" "when updating blocked sites"⁷³ for an hour, Orange's resolver gave a "false" response to the address *fr.wikipedia.org*, pointing not to Wikipedia's servers, but to a page that read "You have been redirected to this page of the Ministry of the Interior website because you tried to connect to a page whose

69. After this shutdown, thousands of Internet users found themselves deprived of their content in the blink of an eye (and not just their pirated files, given the online petitions and all the people saying that their professional lives were ruined because they no longer had access to all their documents).

70. Fyodor, 2007, *Seclists.org shut down by Myspace and GoDaddy* [<https://seclists.org/nmap-an-ounce/2007/01>].

71. Stephane Bortzmeyer expands on the concept [<https://www.bortzmeyer.org/dns-menteur.html>].

72. Légifrance, 2015, *décret n° 2015-125 du 5 février 2015 relatif au blocage des sites provoquant to acts of terrorism or glorifying them, and sites disseminating pornographic images and representations of minors* [<https://www.legifrance.gouv.fr/affichTexte.do?cidTexte=JORFTEXT000030195477>].

73. Marc Rees, 2016, *Blocage de Google, OVH et Wikipedia : " on ne cherche pas à vous cacher la vérité " assure Orange*, Nextinpact [<https://www.nextinpact.com/article/24123/101785-blocage-google-ovh-et-wikipedia-on-ne-ne-cherche-pas-a-vous-cacher-verite-assure-orange>].

the content incites acts of terrorism or publicly condones acts of terrorism".⁷⁴.

Dereferencing

Finally, a simple but efficient way of preventing access to a website is to remove it from search engines and directories: this is known as dereferencing. The site still exists, but it no longer appears on search engines (such as Google).

In France, dereferencing is one of the techniques used to block sites administratively: the *office central de lutte contre la criminalité liée aux technologies de l'information et de la communication* sends search engines or directories a list of addresses it considers to contain child pornography, to "directly pro- voce[ant] acts of terrorism" or to "glorify" such acts.⁷⁵ They then have 48 hours to ensure that these addresses no longer appear in their results. 4,138 requests for dereferencing were made in 2020, 4 of which were cancelled after review.⁷⁶

28.4.2 Phishing

In the same vein, phishing⁷⁷ (also known as "*phishing*") consists in tricking the Internet user into connecting to a site that is not what they think it is, but looks very similar. For example, a site that looks exactly like a bank's, in order to obtain passwords to a bank account management interface. To do this, opponents buy a domain name that looks like the right one at first glance. All that's left to do is to entice the target to log on to the site, usually by scaring him or her, e.g. "We've detected an attack on your account" or "We've detected an attack on your account".

"You've exceeded your quota", followed by a proposal to regularize the situation. by clicking on the booby-trapped link.

To ensure that the affiché domain name also resembles that of the copied site, there are plenty of techniques: the adversary can, for example, use special characters that have the appearance of Latin alphabet characters. For example, by substituting a Cyrillic "e" for a Latin "e" in *example.org*, you get an address that looks (almost) identical to the original, but which represents a different address for the computer; you can also find hyphens (*ma-banque.fr* instead of *mabanque.fr*); sometimes it's an identical name, with a *top-level domain* (TLD): .com, .net, .org, .fr, etc.) (*site.com* instead of *site.org*). sub-domains (*paypal.phishing.com* links to the phishing site, not to *paypal.com*), etc.

Web browsers are designed to warn users of the danger and ask for confirmation before accessing the suspect site.⁷⁸ However, this

74. Yannux, 2016, Screenshot of Ministry of the Interior page, twitter.com [https://pbs.twimg.com/media/Cu9JoGNWAAAQAO9.jpg].

75. Légifrance, 2015, *décret n° 2015-253 du 4 mars 2015 relatif au déréférencement des sites provoking or condoning acts of terrorism and sites displaying pornographic images and representations of minors* [https://www.legifrance.gouv.fr/loda/id/JORFTEXT000030313562].

76. Alexandre Linden, 2021, *report on the 2020 activities of the qualified person appointed under article 6-1 de la loi n° 2004-575 du 21 juin 2004 créé par la loi n° 2014-1353 du 13 novembre 2014 renforçant les dispositions relatives à la lutte contre le terrorisme*, CNIL [https://www.cnil.fr/sites/default/files/atoms/files/rapport_linden_2020.pdf], p. 9.

77. See Wikipedia, 2014, *Phishing* [https://fr.wikipedia.org/wiki/Hameçonnage], which explains some (partial) countermeasures to this attack.

78. Mozilla, 2022, *How does phishing and malware protection work?* [https://support.mozilla.org/fr/kb/comment-fonctionne-protection-contre-hame%C3%A7onnage-and-malware]

solution requires the web browser to contact a centralized database, listing sites considered to be malicious. This can pose problems of discretion: the server hosting this list will necessarily be aware of the phishing or malware sites being visited.

28.4.3 Attack the server

Another type of attack consists of attacking the computer hosting the resource of interest. This can be done either physically or remotely.

Entering servers

It's simply a matter of an adversary who has the means - the police or the law, for example - going to the location of the computer they're interested in. The adversary can then seize the machine, or copy the data it contains. She can then study all traces left on it by people who have connected to it... page 27 at least if its hard disk is not encrypted.

At least fourteen servers were seized by the courts in Europe between 1995 and 2007.⁷⁹

In 2007, a Greenpeace Belgium server was taken away by the Belgian police following a complaint of "criminal conspiracy" by a Belgian electricity company⁸⁰ against which the environmental organization had called for a demonstration.

More recently, in the spring of 2017, a number of servers belonging to the Tor anonymization network were seized⁸¹ in connection with, or at least with the pretext of, an investigation into a cyberattack that was transiting through this network⁸².

Server hacking

Like any computer, a server can be *hacked*: this involves the attacker "breaking into" the computer. Design or programming errors, which can be used to hijack a program's operation and gain access to the computer on which it is running, are regularly discovered in the programs commonly used on servers. Software configuration errors on the part of server admins are also possible.

In 2014, for example, exploiting vulnerabilities in the publishing software used on the website of Gamma International, the company behind the FinFisher spyware, enabled a hacker to gain access to their server.⁸³ This gave him access to 40 gigabytes of documents, including a list of their customers, documents on the operation and efficacy of their spy software, as well as portions of their website. its source code⁸⁴.

The vulnerabilities that make this kind of hacking possible are not uncommon, and any server can be affected. Once inside the server, hackers can potentially gain remote access to all the data stored on it.

Even without breaking into the server, software vulnerabilities can be discovered and exploited to exfiltrate information that can be accessed by anyone.

79. Globenet, 2007, *Server seizures in Europe: a history* [https://www.globenet.org/Les-seizures-of-servers-in-Europe.html?start_aff=6].

80. Gérard De Selys, 2008, *Greenpeace, association de malfaiteurs*, *Articulations* n°33, CESEP [https://web.archive.org/web/20230129143208/https://www.cesep.be/PDF/ARTICULATIONS/ARTICULATIONS_33.pdf], p. 7.

81. Guénaël Pépin, 2017, *WannaCrypt: Tor nodes seized by French authorities* [<https://www.nextinpact.com/article/26455/104302-wannacrypt-nuds-tor-saisis-par-autorites-francaises>].

82. Wikipedia, 2021, *WannaCry* [<https://fr.wikipedia.org/wiki/WannaCry>].

83. Phineas Fisher, 2014, *Hack Back - DIY Guide for those without the patience to wait for whistleblowers* [<https://gist.github.com/vlamer/2c2ec2ca80a84ab21a32#file-gistfile1-txt-L171>] (in English).

84. Wikipedia, Phineas Fisher [https://en.wikipedia.org/wiki/Phineas_Fisher].

who shouldn't have access. This is what allowed the famous leak of personal data from 500 million Facebook accounts⁸⁵ from June 2020.

Denial of service attack

Without seizing or even hacking the server, it is possible to prevent it from working by saturating it: the adversary ensures that a large number of robots are constantly trying to connect to the site to be attacked. Beyond a certain number of requests, the server software becomes overwhelmed and can no longer respond, rendering the site inaccessible. This is known as a *denial-of-service attack*.⁸⁶ The bots used for this type of attack are often malware installed on personal computers without the owner's knowledge.

32

28.4.4 On the way

Finally, an adversary controlling part of the network - such as an ISP - can eavesdrop or hijack packets in a number of ways.

Filtering

As mentioned above, an adversary controlling one of the routers through which traffic passes between an Internet user and a resource can read the content of the packets in greater or lesser depth and possibly modify it, all the more easily if it is not encrypted.

Today, virtually all ISPs use this type of inspection, *DPI*, at the very least for statistical purposes. What's more, more and more of them are using it, more or less discreetly, more or less deliberately, to put certain packets ahead of others, depending on their destination or the application to which they correspond. For example, to slow down video-on-demand, which generates a lot of traffic (and therefore costs them a lot of money), and give priority to Internet telephony.⁸⁷ SFR, for example, uses this type of tool⁸⁸ to modify the web pages visited by its 3G subscribers.⁸⁹

The massive deployment of equipment enabling this in-depth examination of packets makes surveillance at the gateways to ISP networks much easier.

By analyzing this type of data, governments can identify the position of an individual, their relations and members of a group, such as "political opponents".⁹⁰ Such systems have been sold by Western companies to Tunisia, Egypt, Libya, Bahrain and Syria⁹¹ and are also in use in a number of Western countries. Based on mass surveillance, these systems enable Internet users to be targeted, and content to be filtered and censored.

The use of this technique, in Spain for example, has enabled certain ISPs to monitor the traffic of its users and prevent them from accessing the company's website.⁹² of its users to prevent them from accessing the

85. Elise Viniacourt, 2021, *Facebook : les données de 533 millions d'utilisateurs en fuite sur-le-web*, Liberation.fr [https://www.liberation.fr/economie/economie-numerique/facebook-les-donnees-de-533-millions-dutilisateurs-en-fuite-sur-le-web-20210406_FNRIQR4PXB5BK6ALSEREIOPOY/].

86. Wikipedia, 2021, *Denial of service attack* [https://fr.wikipedia.org/wiki/Attaque_par_d%C3%A9ni_de_service].

87. Wikipedia, 2021, *Deep packet inspection* [https://fr.wikipedia.org/wiki/Deep_packet_inspection].

88. bluetouff, 2013, *SFR changes the HTML source of pages you visit on 3G* [https://web.archive.org/web/20150629235630/https://reflets.info/sfr-modifie-le-source-html-des-pages-que-vous-visitez-en-3g/].

89. Wikipedia, 2021, *3G* [https://fr.wikipedia.org/wiki/3G].

90. Elaman, 2011, *Communications monitoring solutions* [https://wikileaks.org/spyfiles/docs/elaman/188_communications-monitoring-solutions.html].

91. Jean Marc Manach, 2011, *Internet massively monitored* [https://web.archive.org/web/20190411142441/http://owni.fr/2011/12/01/spy-files-interceptions-ecoutes-wikileaks-qosmos-amesys-libye-syrie/].

92. Sans Censure, 2020, *Summary of the technical report and current status of the Wome-site nOnWeb* [https://sindominio.net/sincensura/fr/post/censura/].

the NGO [Women on Web](https://www.womenonweb.org/fr/) [https://www.womenonweb.org/fr/], which provides abortion information and assistance worldwide.

Listening

Just like good old telephone tapping, it is now possible to record all or part of the data passing through a network link: this is known as "IP tapping". This makes it possible, for example, to eavesdrop on all traffic exchanged by a server, or that passing through a domestic ADSL connection.

In France, such interceptions are authorized as part of a judicial investigation, but also for the "prevention of terrorism" to gather "information or documents [...] relating to a person [...] likely to be linked to a threat" but also relating to "persons belonging to the entourage of the person concerned".⁹³

If no special precautions are taken, an IP interception reveals to an adversary much of our Internet activity: web pages visited, emails and their content, instant messaging conversations... everything that leaves our computer "unencrypted". The encryption of communications makes the analysis of the content resulting from such eavesdropping much more difficult: the adversary still has access to the data exchanged, but she cannot understand and exploit it directly. She can then try to break the encryption used... or attempt to circumvent the way it is implemented. We'll talk more about these encryption-related issues later. In any case, the adversary will always have access to a certain amount of valuable information, such as the IP addresses of the various interlocutors involved in a communication.

page 249

page 202

Network traffic analysis

When traffic is encrypted, more subtle attacks are still possible. An adversary who can eavesdrop on network traffic, even without access to data content, has other clues at his disposal, such as the amount of information transmitted at any given time.

So, if Ana sends 2 MB of encrypted data to a publishing website, and a new 2 MB document appears on this site a few moments later, this adversary will be able to deduce that it was probably Ana who sent this document.

By studying the quantity of information transmitted per unit of time, opponents can also draw a "shape": we'll call it the *traffic pattern*.⁹⁴ The content of an encrypted web page will thus not have the same pattern as an encrypted instant messaging conversation.

What's more, if the same traffic pattern is observed at two points on the network, adversaries can assume that it's the same communication.

To take a specific example: let's consider adversaries who are eavesdropping on Ana's ADSL connection, and who observe encrypted traffic that they can't decrypt, but who suspect that Ana is chatting with Bea via encrypted instant messaging. Let's assume they also have the means to tap Bea's connection. If they observe a similar pattern between the traffic leaving Ana's house and that entering Bea's a few (milli-)seconds later, they'll be backed up in their hypothesis - without, however, having formal proof.

93. Légifrance, 2021, *Code de la sécurité intérieure*, article L851-2 [https://www.legifrance.gouv. en/codes/article_lc/LEGIARTI000043887533/].

94. Yin Zhang, Vern Paxson, 2000, *Detecting Stepping Stones*, Proceedings of the 9th USENIX Security Symposium [https://www.usenix.org/legacy/events/sec2000/full_papers/zhangstepping/zhangstepping.pdf].

This type of attack can be used to confirm a pre-existing hypothesis, but not to develop one based on the information gathered alone, unless the adversaries have the means to eavesdrop on the *entire* network where the traffic between Bea and Ana is located, and have colossal computing power at their disposal. The existence of such global adversaries is technically possible, but not very realistic. On the other hand, agencies like the NSA are capable of carrying out this type of attack, at least on the scale of their country: the NSA has computing power that can be suffisante, and leaks indicate that it would listen in on 75% of US Internet traffic. ⁹⁵.

28.4.5 Customer hacking

The Internet user's computer can also be a target. Just as attackers can break into a server, they can also break into a personal computer. Programming errors or other flaws in the operating system or installed applications sometimes enable adversaries to carry out such piracy - legal or illegal - from the Internet, without having physical access to the machine. What's more, intrusion can be facilitated by bad practices on the part of users, such as opening fraudulent attachments or installing programs found at random on the web.

A renowned German hacker group, the Chaos Computer Club, has uncovered a bug used by the German police to spy on and control a computer remotely. ⁹⁶. Such bugs can be installed remotely and are permitted under French law.

But "remote spying" is not just reserved for police practices. In the United States, a high school has embarked on a massive spying operation. Under the guise of "recovering stolen or lost laptops", the school had installed a "function" enabling the webcams of the several thousand computers distributed to students to be turned on at the school's discretion. The case came to light at the end of 2009: one of the students was accused of "inappropriate behavior", in this case drug use. The official accusing the student produced, as evidence, a photo which turned out to have been taken without the student's knowledge, by his computer's webcam while he was at home in his bedroom. ⁹⁷!

28.5 In conclusion

Identifying the Internet user by his IP address, reading the origin and destination of packets via their headers, recording various information at different stages of the journey, even accessing the actual content of exchanges... all this is more or less straightforward depending on the entity involved.

Pirates, advertisers, the police in Saint-Tropez and the NSA do not have the same technical and legal possibilities for accessing the traces described in this chapter.

Let's just conclude by observing that the way in which the Internet was conceived and is most commonly used is virtually transparent to even the most attentive adversaries... unless they use a whole series of parries adapted to make these indiscretions more difficile; these parries will be discussed below.

95. [latribune.fr, 2012, Barely 25% of US web traffic escapes NSA surveillance \[https://www.latribune.fr/actualites/economie/international/20130821trib000781040/a-peine-25-du-traffic-web-americain-echappe-a-la-surveillance-du-nsa.html\]](https://www.latribune.fr/actualites/economie/international/20130821trib000781040/a-peine-25-du-traffic-web-americain-echappe-a-la-surveillance-du-nsa.html).

96. Mark Rees, 2011, *CCC dissects a holey government Trojan horse*, PCInpact [<https://www.nextinpact.com/archive/66279-loppsi-ccc-cheval-de-troie-faille-malware.htm>].

97. Me, myself and the Internet, 2011, *Mais qui surveillera les surveillants?* [<https://web.archive.org/web/20180107033100/https://memyselfandinternet.wordpress.com/2011/02/14/%C2%AB-mais-qui-surveillera-les-surveillants-%C2%BB/>].

Web 2.0

These days, the term web 2.0 is almost commonplace. However, it seems difficult to grasp its true meaning, due to its misuse or, on the contrary, its sometimes overly technical definitions.¹

It's first and foremost a marketing term, defining an evolution of the web at a time when mass Internet access is turning it into a juicy market. Many companies, whether in the media, communications or retail sectors, can no longer afford to ignore it. They have had to adapt their business models to this new market.

The arrival of these new players on a web that until then had consisted mainly of uni-versities and enthusiasts has transformed the way websites are designed, and consequently the way web users use them.

Beyond these marketing formulations, we're going to take a closer look at how these evolutions manifest themselves to Internet users, and the changes in the way the network operates that they imply.

29.1 Rich Internet applications"...

One of these evolutions concerns the interactivity of websites. They are no longer simply static pages like those in a book or magazine. Using pre-existing Web 2.0 technologies such as JavaScript, websites increasingly resemble applications such as those found on our personal computers: dynamic websites responding to the user's requests.

[page 210]

What's more, most of the software normally installed on a personal computer has been transposed to a web version, and is now accessible via a web browser.

We're even seeing the emergence of operating systems, like Chrome OS, designed page 22 entirely along these lines. This movement, this shift from software installed on the computer to the web, is in particular a response to concerns about incompatibility of software, licenses and upgrades.

There's no need to install anything: just connect to the Internet and, *via* a web browser, you'll have access to most traditional applications: word processing, spreadsheets, e-mail, a collaborative diary, a file-sharing system, a music player, *and so on.*

Google Drive lets you write documents and do your accounting online, for example. But the service also lets you share it with friends, colleagues and *so on.*

1. The opening presentation at O'Reilly and Battelle's Web 2.0 conference, cited by [Wikipedia, 2014, Web 2.0](https://fr.wikipedia.org/wiki/Web_2.0) [https://fr.wikipedia.org/wiki/Web_2.0] is a fine example of an overly technical definition.

Some people even see the possibility of accessing these online tools from "any computer, in any country, at any time" as a way of reconciling work with possible medical, weather or even pandemic problems. ² a way of reconciling work with possible medical, weather or even pandemic problems...

No need to go to the office, "a computer connected to the Internet suffit to immediately reconstitute the work environment".

29.2 ... and volunteer web surfers

When these companies entered the web market, they had to rethink their business model. As the Internet audience grew, it was no longer possible to finance a website on advertising alone, while paying an army of female editors to provide ever-increasing amounts of content.

Service providers used a technique that had already been present on the web for a long time: relying on the participation of Internet users. From now on, it is the latter who are responsible for writing the content that feeds the sites. Service providers simply host the data and provide the interface for accessing it, but also and above all add advertising around it... and collect the cash.

For many years, the YouTube video-sharing platform has enabled its users to upload and view the videos of their choice free of charge, with no visible quid pro quo. Today, thanks to its success and monopoly, most people who want to view and share videos are dependent on this platform, which allows YouTube to gradually impose advertising. At first, advertising was displayed on a banner next to the image, then on a transparent banner over the image, and now it's simply videos embedded at the beginning or in the middle of the one you want to watch.

Another advantage of this solution for service providers is that Internet users more or less consciously provide a whole range of data, which can then be monetized. ³ which can then be monetized, notably by building consumer profiles and tailoring advertising affichées to the audience.

[page 221]

It's no longer the case, for example, that surfers use the Internet solely to download films or read their favorite periodicals. Increasingly, for example, by filling in their Facebook page, Internet users are producing content and offering it, so to speak, to the hosts or other companies that provide these services. On their own initiative, Internet users will put online a list of the music they listen to, photos of their vacations in the Meuse, or their contemporary history lessons to share with their classmates.

Of course, by providing content, you're also providing information about yourself, information that the prying eyes of advertisers and other adversaries are bound to use.

[page 221]

[page 223]

29.3 Data centralization

The use of Internet storage space generally goes hand in hand with the centralization of Internet users' data. The most widely used online storage spaces are in fact in the hands of the web giants.

2. Lionel Damm and Jean-Luc Synave, 2009, *Entrepreneur 2.0, la boîte à outils de la compétitivité... à petit frais* [<https://web.archive.org/web/20220125225053/https://www.confederationconstructio.n.be/Portals/28/UserFiles/Files/WP2guideentrepreneurweb20.pdf>].

3. Fanny Georges, Antoine Seilles, Jean Sallantin, 2010, *Des illusions de l'anonymat - Les stratégies de préservation des données personnelles à l'épreuve du Web 2.0*, Terminal numéro 105, Technologies et usages de l'anonymat à l'heure d'Internet [<https://www.revue-terminal.org/article/s/105/introDossierAnonymat105.pdf>].

Using online applications means, among other things, that documents are no longer stored on a personal computer, hard drive or USB stick. Instead, they are stored on remote servers such as those operated by Google⁴ servers, in data processing centers far from the Internet user, both geographically and technically. In other words, Internet users lose control over their data.

A simple lack of Internet connection makes it impossible to access your documents, unless you've made a backup. This shift in storage also makes it impossible to securely erase the page 42 documents stored on it. [-----]

This tendency to migrate data and applications from the personal computer to the Internet also creates a "connection addiction". When all your music, address book and city maps exist only on the Internet, it becomes difficult to imagine using a computer *offline*. Yet any connection to the Internet opens doors. And the more exposed a computer is, the more difficult it is to guarantee its security - from the anonymity of the user to the confidentiality of the data entrusted to it. [page 221]

Nor is there any guarantee that our online data is secure. Even if an organization gives us every guarantee of security today (and yet, what proof do we have of this?), it is still not safe tomorrow from the discovery of a flaw, or a program configuration error that would give anyone access to this data, as was the case with the encrypted online data storage service Dropbox.⁵ [page 235]

The companies to which we entrust our data can also, on their own initiative, delete content⁶ delete our account⁷ or even shut down their services through no fault of their own - or simply go bankrupt. And when states get involved, a court decision can close down a service, as in the case of Megaupload, or a simple report from an authority in another state can now force an online service provider to remove content qualified as terrorist in less than an hour.⁸ [page 233]

29.4 Program control

Most of the time, these online applications are developed in a more closed way than the free applications you can install on your computer. When Google or Facebook decide to modify the interface or change the way things work service, to "tidy up", the web user has no say in the matter. [-----]

What's more, the interactivity of these web applications means that part of their program must be executed on the client computer (ours), using technologies such as JavaScript or Java. These technologies are now activated by default in our web browsers, for all sites. It's nice, practical and modern. But [page 214]

4. The paragraph *Elements that you create or provide to us* in the **Privacy Policy** [<https://policies.google.com/privacy?hl=fr#infocollect>] for services provided by Google clearly demonstrates the lack of concrete power an Internet user has over the content she has stored online. "What's yours, stays yours", but Google is free to do what it likes with it, as long as you leave your content on its servers.

5. Vincent Hermann, 2011, *Dropbox admits to possessing duplicate data access keys* [<http://www.nextinpact.com/archive/64460-dropbox-conditions-utilisation-chiffrement-securite.htm>].

6. Marie Claire, 2018, *Breast cancer: Facebook censors mastectomy publications (again)* [<https://www.marieclaire.fr/cancer-du-sein-facebook-censure-publications-mastectomie,1249169.asp>].

7. Owni, 2011, *After 7 years of use, he has his Google account deleted, including emails, calendars, docs, etc.* [<https://web.archive.org/web/20200224160152/http://owni.fr/2011/08/29/google-deletion-account-personal-data-life-privee-god/>].

8. Article 17 of *Regulation (EU) 2021/784 of the European Parliament and of the Council of 29 April 2021 on combating the dissemination of terrorist content online* [<https://eur-lex.europa.eu/legal-content/FR/TXT/HTML/?uri=CELEX:32021R0784>].

these technologies pose a number of problems for the security of our computers, and therefore for the confidentiality of our data⁹... However, it is possible¹⁰ their use on a site-by-site basis, depending on the level of trust you place in them.

29.5 From centralization to decentralized self-hosting

Faced with ever-increasing centralization of data and applications, can we enjoy the benefits of a participative, interactive network without losing control over our data? The challenge seems daunting. But work is underway to develop Internet applications that would operate decentrally on each surfer's computer, rather than being centralized on a few servers. Projects such as peer-to-peer social media, Mastodon¹¹ Nextcloud¹² the YunoHost¹³ distribution, or the BriqueInter.net¹⁴ are working in this direction.

Until they are as easy to use as the solutions offered by the Web 2.0 giants, it's already possible to host most of the services you want to offer or use yourself.

9. We have no control over the JavaScript or Java programs sent by the web application. It is therefore entirely possible that bugs or other malicious functionality [page 32] may be included among these programs and then executed by our *na vigateur*.

10. Depending on the web browser you use, there are *extensions* such as *noscript* [<https://noscript.net>], which allow you to manage these parameters.

11. Mastodon [<https://joinmastodon.org/>].

12. Nextcloud [<https://nextcloud.com/fr/>].

13. YunoHost project page [https://yunohost.org/#/index_fr].

14. La BriqueInter.net [<https://labriqueinter.net/>].

Contextual identities

One of the presuppositions of this *guide* is the desire that our actions, gestures and thoughts should not be automatically, if at all, linked to our civil identity.

However, it may be necessary or simply preferable to know who we're talking to: to start a discussion on a forum or send e-mails, for example. In such cases, having an *identity* - i.e. being identifiable by our correspondent - simplifies communication.

30.1 Definitions

First, two definitions:

- *anonymity* means not revealing your name;
- *Pseudonymity* means choosing and using a name different from your civil identity.

Because of the way it works, it's very difficult to be *anonymous* or remain a *pseudonyme* on the Internet.

30.1.1 Nicknames

A *pseudonym* is an identity that is not the one assigned to a person by civil status. We can choose to be called "Falaise", "Amazone enragée", "Zigoui-goui", or even "Jeanne Dupont". By keeping the same pseudonym for different exchanges, our interlocutors will have a good chance of thinking that the various messages written by this *pseudonym* come from the same person: they'll then be able to reply to us, but won't be able to come and beat us up if they disagree.

When choosing a pseudonym, however, you need to be aware that it can in itself be a clue to the person using it, at least for people who already know the pseudonym.

30.1.2 Contextual identity

Bea immediately downloads the document and opens it in the text editor. She flips through it quickly, deleting a few details that are best left alone. After entering her login and password for logging on to the blog, Bea copies and pastes the contents of the document from her mailbox, and clicks Send. "Let's hope it inspires others!"

Continuing the thread of our introductory story, the contextual identity would correspond to "one or more people publishing information about the mayor", and the physical person to Bea.

Whether we're talking to people with whom we share a passion for climbing, or about our professional project with a Pôle emploi advisor or our banker, the tenor of what we say, the way we talk about it, is not the same. On the one hand, we'll be rather exalted and adventurous, on the other, more sober and serious. So we can speak of contextual identity.

It's the same when using a computer: when you post a message on a dating forum, announce a big party on your Facebook account or reply to an email from Dad, you're using different contextual identities. These can, of course, be mixed together to form a single identity made up of the three contextual identities mentioned above: the single woman, the party girl and the daughter of.

A contextual identity is therefore a fragment of a global "identity" that is supposed to correspond to a physical person, or a group. Just as a photograph is a snapshot of a person or group, from a certain angle, at a certain age, *etc.*, so a contextual identity *is a fragment of a global "identity"*.

[page 213]

It's not easy to be absolutely anonymous on the Internet: as we've seen, many traces are recorded when using the network. This phenomenon is all the more true with social media, for which the generation of a unique, traceable identity is a core business.¹ It's impossible to leave no trace, but it may be possible to leave traces that lead nowhere.

Similar difficulties are encountered when choosing pseudonymity: the more you use a *pseudonym*, the more traces you leave behind. Small clues that, when cross-checked, can reveal the civil identity that corresponds to a pseudonym.

30.2 From contextual identity to civil identity

There are various ways, more or less offensive, of undermining a pseudonym or revealing the link between a contextual identity and the physical person(s) using it.

30.2.1 Cross-checking

[previous
page.]

Starting from the example of three contextual identities, it is legitimate to ask what juggling these different identities implies in terms of anonymity. Assuming that you use a pseudonym rather than your civil identity, it may be more appropriate to have one identity, i.e. a *pseudonym*, in each context: one for dating sites, another for social media, and one for family relationships, *etc.*, in order to avoid overlap. If the information emanating from these identities is not compartmentalized, i.e. if the same pseudonym is used, their cross-referencing can reduce the number of people to whom they may correspond. This makes it easier to link a digital presence to a physical person, and thus to put a name to the corresponding contextual identity.

Consider, for example, a person who uses the pseudonym *bruise76* on a blog where she says she's a vegetarian and likes action movies. There are only so many people who fit these criteria. Add to this the fact that this same pseudonym is used to organize a fiesta in such-and-such a town *via* social media and to communicate by e-mail with Ms. Unetelle. There probably aren't many vegetarians who like action movies, organize a party in the same town and communicate by e-mail with Ms. Unetelle.

1. Ippolita, 2012, *I don't like Facebook* [<http://inventin.lautre.net/livres/Ippolita-J-aime-pas-Facebook.pdf>].

The more numerous and varied the uses of a pseudonym by the same person, the more restricted the number of people who can match that pseudonym. Thus, by cross-checking the uses of the same pseudonym, it is possible to weaken or even break the pseudonym.

Here's an example of the weakness of pseudonymity: AOL published the results of 3 months of queries submitted to its search engine. Queries from the same person were associated with the same pseudonym. By cross-checking, it was possible to break the pseudonymity associated with the queries.²

Similarly, the governor of the state of Massachusetts was also the victim of this cross-referencing when his medical records, supposedly anonymized, were identified among those of all the state's female citizens. The researcher who carried out this demonstration of data de-anonymization even went so far as to send him his medical file by mail.³

30.2.2 Time correlation

A little more technical this time, temporal correlation also makes it possible to break or weaken anonymity or pseudonymity a little more. If, within a short space of time, there is a connection to the `ama-zone@exemple.org` mailbox as well as `jeanne.dupont@courriel.fr`, the probability that these two mail addresses are in the hands of the same person increases, and all the more so if this observation is repeated. Various countermeasures, to meet different needs, will be explained below.

30.2.3 Stylometry

Statistical analysis can be applied to the shape of any type of data, including text. By analyzing⁴ characteristics of a text, such as the frequency of word-posts⁵ the length of words, sentences and paragraphs, and the frequency of punctuation marks, we can correlate anonymous texts with other texts, and extract clues about their authors.

This type of analysis was used, for example, during the trial of Theodore Kaczynski⁶ to prove that he was the author of the manifesto "Industrial Society and its Future".⁷

The authors of a recent study⁸ sought to "simulate an attempt to identify the author of an anonymously published blog. If the author is sufficient to avoid revealing her IP address or any other explicit identifier, her adversary (e.g. a government censor) may turn to analyzing her writing style". Their findings show that stylometry can greatly reduce the number of possible authors of a given

2. Nate Anderson, 2006, *AOL releases search data on 500,000 users* [<https://arstechnica.com/uncategorized/2006/08/7433/>].

3. Paul Ohn, 2009, *Broken Promises of Privacy: Responding to the Surprising Failure of Anonymization* [<http://www.uclalawreview.org/pdf/57-6-3.pdf>].

4. For example, thanks to software such as *The Signature Stylometric System* [<https://www.philocomp.net/texts/signature.htm>] or *Java Graphical Authorship Attribution Program* [https://evlilabs.com/?page_id=42].

5. Toolwords are words whose syntactic role is more important than their meaning. They are typically *linking words* [<https://fr.wikipedia.org/wiki/Mot-outil>].

6. Kathy Bailey, 2008, *Forensic Linguistics in Criminal Cases, Language in Social Contexts* [https://archive.org/download/bailey-forensic-linguistics-paper/Bailey_-_Forensic_Linguistics_Paper.doc] (in English).

7. Theodore Kaczynski, 1998, *Industrial society and its future* [<https://www.fichier-pdf.fr/2012/12/20/kaczynski/kaczynski.pdf>].

8. Hristo Paskov, Neil Gong, John Bethencourt, Emil Stefanov, Richard Shin, Dawn Song, 2012, *On the Feasibility of Internet-Scale Author Identification* [<https://www.cs.princeton.edu/~arvindn/publications/author-identification-draft.pdf>].

anonymous text - precision obviously increasing with the number of samples "signed", i.e. with a known author, supplied to the analysis software.

More often than not, this enables them to reduce the size of the set of possible female authors from 100 to 200 out of 100,000 initially. "[...] added to another source of information, this can be sufficient in making the difference between anonymity and identification of an author". At the time of writing, it is even possible in 20% of cases to directly identify the anonymous author.

The particularity of this work is that it goes beyond the framework of small samples (around a hundred possibilities) to which previous studies were confined, to focus on the identification of the author among a very large number of possibilities; in other words, it demonstrates that stylometry can be used to confirm the origin of a text on the basis of a very large number of samples.

However, writing in an attempt to disguise one's style, without any particular expertise, seems to make stylometric analyses inefficient. Imitating someone else's style even fools them in more than half the cases.⁹

Other researchers are developing software that suggests the changes needed to anonymize a text.¹⁰

30.3 Compartmentalization

As we've just seen, there are a number of ways in which a civil identity can be matched with a contextual identity. Using the same name for different activities is undoubtedly the practice most likely to confuse us.

So it's important to think carefully about how you use your pseudonyms. It is often dangerous to mix several contextual identities under the same pseudonym. The best way to prevent this is to keep them clearly separate from the outset, so as to limit any subsequent problems. After all, a practice or identity that can be used at a given moment can suddenly turn into a source of problems due to external conditions that cannot necessarily be anticipated or controlled.

However, these practices are not always easy to implement. For, in addition to the techniques described above, the separation between these different contextual identities depends on many other parameters. In particular, the relationships you establish with other people, whether these relationships are digital or not. It's not necessarily easy to have a different contextual identity for absolutely every facet of one's personality or every activity, or to avoid some of them overlapping. These identities evolve with the activities we assign them and over time. The longer they are used, the more their separation tends to diminish. It is therefore often difficult to balance and measure the efforts required to set up multiple contextual identities against the expected benefits. All the more so as it is generally complicated to backtrack in this area.

Some tools, such as social media, even make them virtually impractical by imposing absolute transparency.

[page 244]

9. M. Brennan, R. Greenstadt, 2009, *Practical attacks against authorship recognition techniques*, in *Proceedings of the Twenty-First Innovative Applications of Artificial Intelligence Conference* [<https://www.aaai.org/ocs/index.php/IAAI/IAAI09/paper/viewFile/257/1017>].

10. Andrew W.E. McDonald, Sadia Afroz, Aylin Caliskan, Ariel Stoleran, Rachel Greenstadt, 2012, *Use Fewer Instances of the Letter "i": Toward Writing Style Anonymization*, The 12th Privacy Enhancing Technologies Symposium [<https://www1.icssi.berkeley.edu/~sadia/papers/anonymouth.pdf>].

30.4 Social media: centralized functions and a unique identity

Indeed, social media tend to centralize functions that were previously performed by different tools, from message exchange to news publishing to newsgroups. They tend to replace email, instant messaging, blogs and forums.

At the same time, new functions are developing, such as a certain relational digital life where the existence of a communication takes precedence over its content, pushed to its paroxysm with "pokes", those messages without content.¹¹ The web

2.0 encourages expression on subjects previously considered intimate¹².

In the end, there's not much that's new, apart from the centralization of numerous functions and varied practices into a single tool. In fact, this is the These platforms' "all-in-one" design and ease of use have made them a success. But this centralization raises questions about the consequences of using these tools on our intimacy.

The social pressure to use social media is very strong in some places: when groups use them for the majority of their communications, from interpersonal messages to invitations to the publication of information, not to participate in social media is to be marginalized. The success of these sites is based on the "network effect": the more people who use them, the more important it is to be present.

But at the same time, these social media also allow us to escape these group pressures and more easily assume or experiment with certain parts of our personality that are not necessarily tolerated by these groups.

Centralizing all activities on a single platform makes it extremely difficult to use different pseudonyms for different contextual identities. Indeed, by putting all the information in one place, the risk of different contextual identities overlapping is maximized. Many social media require a single identity, corresponding to the civil identity of a physical person. This is a key difference from a model where an individual can have several blogs with different tones and content, each under a different pseudonym. What's more, just as with dating sites, where the more honest you are, the better the results, here the more content you provide, the more you use this platform, the better the interactions.

This is all the more true given that using one's civilian identity is part of the rules of networks such as Facebook, which sets up various mechanisms to track pseudonyms¹³. These companies push the *business model* of targeted publicity and profile sales to the limit: they "implement various technical processes to capture users' identities, from identities based on their declarations, to active identities¹⁴ and calculated identity based on analysis

11. Fanny Georges, 2008, *Les composantes de l'identité dans le web 2.0, une étude sémiotique et statistique*, Communication au 76^{ème} congrès de l'ACFAS : Web participatif : mutation de la communication?, Québec, Canada [<https://hal.archives-ouvertes.fr/hal-00332770/>].

12. Alain Rallet and Fabrice Rochelandet, 2010, *The regulation of personal data on the web* Réseaux issue 167, Données personnelles et vie privée [<https://www.cairn.info/revue-reseaux-2011-3-page-17.htm>].

13. Nikopik, 2012, *Facebook and ranting* [<https://geeko.lesoir.be/2012/09/24/facebook-demande-a-ses-membres-de-denoncer-les-pseudonymes/>].

14. Active identity" refers to messages that automatically appear on the that detail a person's activity on the platform. These messages don't reflect what the person is saying on the site, but *what they're doing there*. For example, "Ana has changed her profile photo" or "Ana is now friends with Betty". Fanny Georges, Antoine Seilles, Jean Sallantin, 2010, *Des illusions de l'anonymat - Les stratégies de préservation des données personnelles à l'épreuve du Web 2.0*, Terminal numéro 105, Technologies et usages de l'anonymat à l'heure d'Internet [<https://journals.openedition.org/terminal/1876>].

of their behavior (sites visited, number of messages, *etc.*). It seems that total anonymity is becoming impossible in a virtual universe where users are first and foremost consumers who need to be observed."¹⁵

In July 2011, Max Schrems succeeded in obtaining all the data Facebook held on him, by invoking a European directive. The file he received comprises 1,222 pages¹⁶ which include not only all the information available on his profile, but also all events to which he has been invited (including declined invitations), all messages sent or received (including deleted messages), all photos uploaded to Facebook accompanied by metadata including geolocation, all "pokes" sent or received, all "friends" (including deleted "friends"), Facebook connection logs (including IP address and geolocation), all "machines" (identified by a cookie) used by a profile, as well as other profiles using the same "pokes".

"or the location of its last known connection to Facebook. (longitude, latitude, altitude).

Finally, despite the Facebook founder's declaration that the age of privacy is over¹⁷ a number of strategies remain to be developed and reworked, in order to play with the various margins that are still relevant. And this with a view to getting a grip on these fundamental questions: "What do we want to show?", "What do we want to make visible?" and "What do we want to hide, and at what cost?"

15. Chantal Enguehard, Robert Panico, 2010, *Approches sociologiques*, Terminal numéro 105, Technologies et usages de l'anonymat à l'heure d'Internet [<https://journals.openedition.org/terminal/1868>].

16. Damien Leloup, 2012, *Max Schrems : "The important thing is that Facebook respects the law"*, Le Monde [https://www.lemonde.fr/technologies/article/2011/11/23/max-schrems-l-important-c-est-que-facebook-respecte-la-loi_1607705_651865.html].

17. Bobbie Johnson, 2010, *Privacy no longer a social norm, says Facebook founder* [<https://www.theguardian.com/technology/2010/jan/11/facebook-privacy>].

Hiding the content of communications: cryptography asymmetrical

In the first volume of this guide, we saw that the most serious avenue for to protect data from prying eyes is encryption.

47 unreadable by anyone who does not have the *secret key*.

page

31.1 Limits of symmetric encryption

With symmetrical encryption, the same secret key is used for both encryption and decryption.

Symmetrical encryption is ideal for encrypting USB sticks and other storage media.

50 storage media.

page

When you want to encrypt a communication, it's trickier: the person who has to decrypt the data is not the same as the one who encrypted it.

If the secret key were the same for all the people you communicate with, then each of them would be able to decipher messages not intended for them. So we need as many secret keys as there are people we're communicating with, and we need to find a way of exchanging these secret keys confidentially.

31.2 A solution: asymmetric cryptography

In the 1970s, cryptographers came up with a solution to the problems posed by cryptography. symmetrical encryption by creating asymmetrical encryption.

page 47

With asymmetric encryption, each person communicating has a pair of keys: a *public* key so that encrypted messages can be written to them, and a *private key* so that they can decrypt and read them.

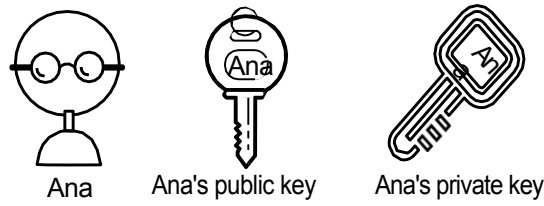
For each exchange, imagine that the communications travel in a box fitted with a special lock.

The public key locks the box when the message is sent. However, it cannot be used to unlock the box.

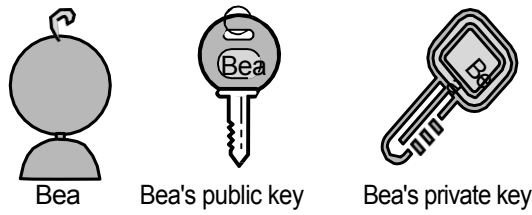
The other key, the private key, is only used to unlock the box and thus access its contents.

The public key can be distributed to anyone. It can even be put online, since it only serves to lock the box. The private key, on the other hand, is never shared.

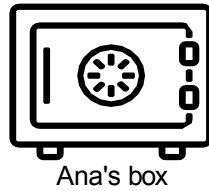
In our example, encryption takes place on Bea's computer and decryption on Ana's computer.



Ana has a pair of keys.



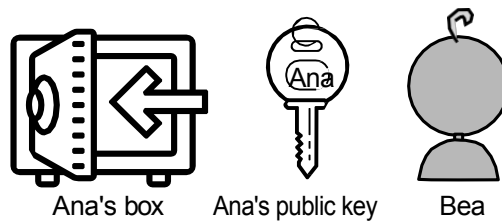
Bea also has a pair of keys.



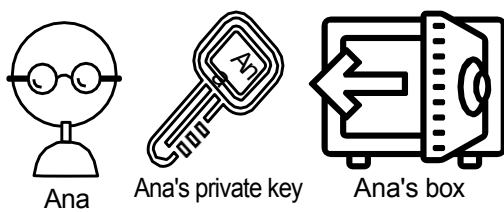
Messages intended for Ana will be placed in a box that will be locked with her public key.



Bea gets Ana's public key.



Bea drops a message in Ana's box, then locks it with Ana's public key.



Ana uses her private key to unlock the box and retrieve the message. Only her private key can unlock the box.

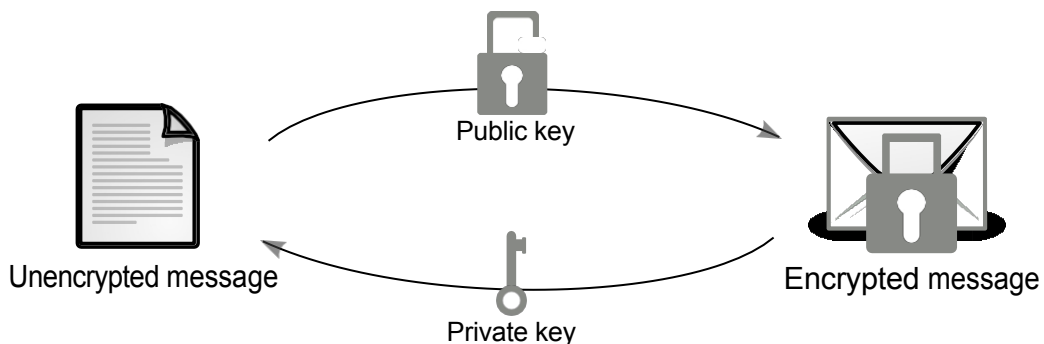
31.3 End-to-end encryption

When only those communicating can read the messages exchanged, this is known as *end-to-end encryption*. In principle, this prevents eavesdropping, including by telecoms providers, Internet service providers and even the communications service provider. With end-to-end encryption, no one is able to intercept the cryptographic keys needed to decrypt the conversation.

End-to-end encryption systems are designed to resist any attempt at surveillance or falsification, as no third party can decipher the data communicated or stored. In particular, services offering end-to-end encryption are unable to deliver a decrypted version of their users' messages to the authorities.¹

31.3.1 A matter of prime numbers...

In reality, the public and private keys are numbers. What one key can encrypt, the other can decrypt:



The public key is used for encryption and the private key for decryption

But how is it possible for the public key to encrypt a message without allowing it to be decrypted? Asymmetric cryptography is in fact based on mathematical problems that are extremely difficult to solve. The RSA encryption algorithm, for example, is based on "integer factorization". In other words, the decomposition of a whole number into prime numbers.

Given the number 12, it's simple to decompose it into $2 \times 2 \times 3$. Similarly, 111 is equal to 3×37 . But how do you decompose the following number, which has 232 digits?

1230186684530117755130494958384962720772853569595334792197322452151726400
5072636575187452021997864693899564749427740638459251925573263034537315482
6850791702612214291346167042921431160222124047927473779408066535141959745
9856902143413

The result is the product of two prime numbers, each with 116 digits.

This problem of factoring integers has been studied by mathematicians for over 2000 years, yet no practical solution has yet been found: the best known solution is to try with all possible primes.

With today's computers, this calculation would take much longer than a human lifetime.² The most difficult numbers to factor are the products of two

1. This section is taken from Wikipedia, 2022, *End-to-end encryption* [https://fr.wikipedia.org/wiki/Chiffrement_de_bout_en_bout].

2. Factoring this 768-bit number in 2010 required 10^{20} operations. The researchers who carried it out estimate that the calculation would have taken around 2,000 years on a single core of an AMD

large prime numbers. So we'll choose numbers sufficiently large that even with extremely powerful computers, factoring can't be done in realistic time.

Trusting this method means betting that your opponent has relatively limited computing power. Key size, measured in bits, is of paramount importance. If we consider that a 2048-bit asymmetric key is secure until 2030³ a 512-bit key can be broken in just a few months on today's high-end personal computers.⁴ Bear in mind that what can be broken by one computer in ten years could be broken in one year by ten computers identical to the first.

What's more, if one day someone solves this mathematical problem, it will be possible to decrypt the encrypted exchanges that have been recorded without too much difficulty.

- This type of data collection and storage is part of the activities of the NSA, the US intelligence agency.⁵ Many military and commercial secrets would then be revealed to those who have access to these recordings. In other words, we can imagine quite a mess between competing companies and enemy intelligence agencies...

In the meantime, attacks on asymmetric cryptosystems currently target the way they are implemented in particular software, or an error in the source code, rather than the mathematical principle of the system.

page 39 31.4 Digital signature

The key pairs used for asymmetric cryptography can also be used to prove the authenticity of a message. How does this work? Let's take the example of Bea sending a message to Ana. This time, Bea wants to digitally sign her message so that Ana can be sure she is the author.

In the first volume of this guide, we talked about checksums, or fingerprints: a number used to verify the integrity of a message. This fingerprint is also used to sign digital data. First, Bea's computer calculates a *fingerprint* of the message she will send to Ana.

page 53 Next, this fingerprint is encrypted with Bea's private key: this is the *digital signature*. That's right: the fingerprint is encrypted with Bea's private key, which only she has, and not with Ana's public key. This signature is used to authenticate the sender, not the recipient. As we've just seen, public and private keys are in fact two numbers chosen in such a way that one can decrypt what the other has encrypted. So there's nothing to stop you encrypting something with the private key. The public key is then used to decrypt it.

Bea then sends the message with her signature to Ana.

To verify the signature, Ana's computer will also calculate the message's fingerprint and decrypt the signature in parallel.

Since it is encrypted with Bea's private key, Bea's public key suffices to decrypt this signature. If the received message's fingerprint matches the signature

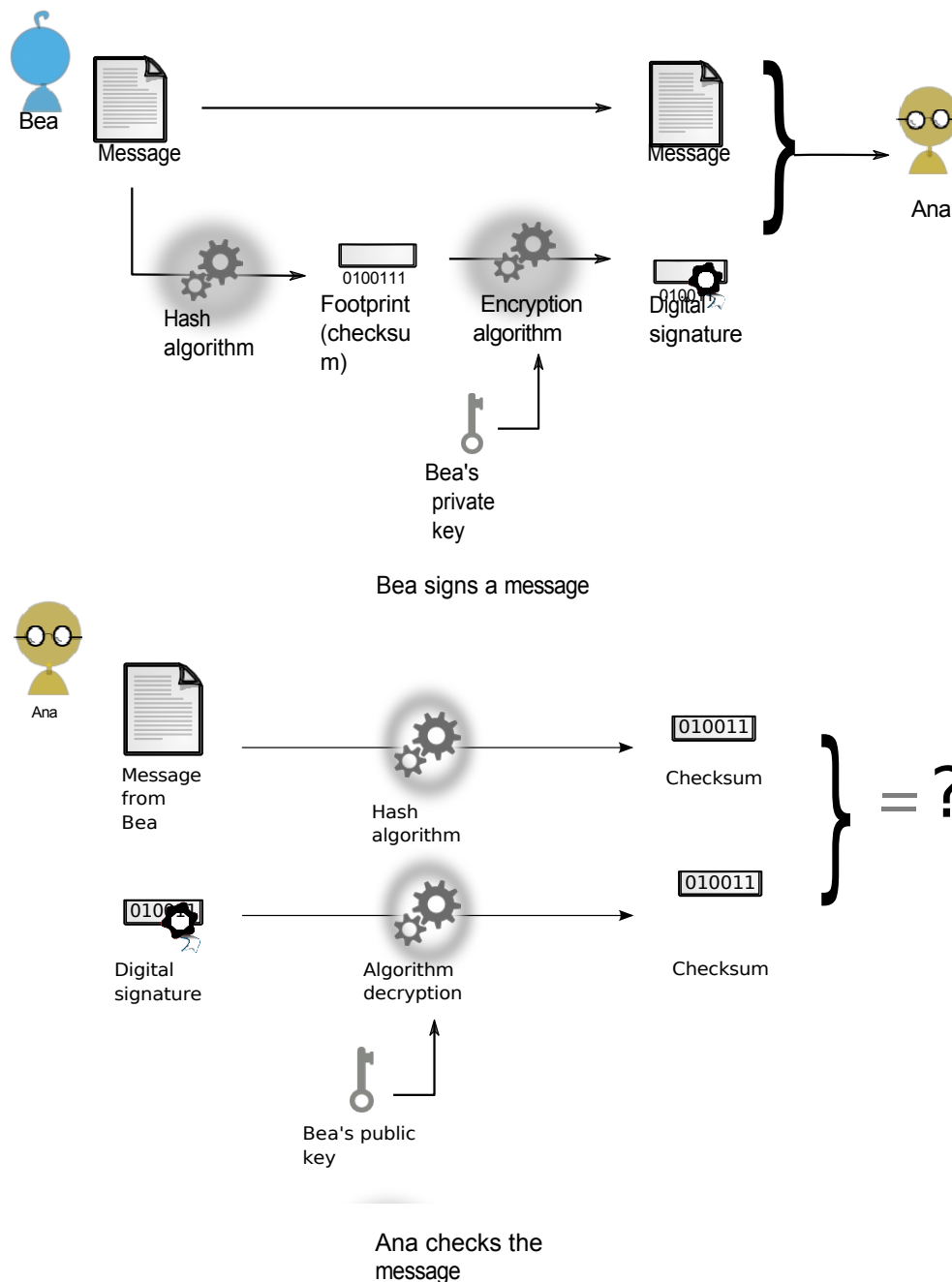
Opteron at 2.2 GHz, which corresponds to several hundred years on a current processor (Kleinjung et al., 2010, *Factorization of a 768-bit RSA modulus* [<https://eprint.iacr.org/2010/006.pdf>]

- in English).

3. Agence nationale de la sécurité des systèmes d'information, 2014, *Mécanismes cryptographiques - Règles et recommandations concernant le choix et le dimensionnement des mécanismes crypto graphiques* [https://www.ssi.gouv.fr/uploads/2014/11/RGS_v-2-0_B1.pdf].

4. S. A. Danilov, I. A. Popovyan, 2010, *Factorization of RSA-180* [<https://eprint.iacr.org/2010/270.pdf>] (in English).

5. Nicole Perloth, Jeff Larson and Scott Shane, 2013, *N.S.A. Able to Foil Basic Safeguards of Privacy on Web*, The New York Times [<https://archive.org/details/n.-s.-a.-able-to-foil-basic-safe-guards-of-privacy-on-web>].



decrypted (this being nothing more than the message fingerprint calculated by Bea's computer), Ana is sure of the authenticity of the message she has received. Bea keeps her private key in a safe place. She is therefore the only one who has been able to encrypt the fingerprint that Ana has decrypted with Bea's public key.

The disadvantage of this certainty is that Bea, possessing the private key, will find it more difficult to deny being the author of the message.

31.5 Verify public key authenticity

Asymmetrical cryptography enables messages to be encrypted and signed without the need to first exchange a shared secret.

However, it doesn't solve an important question: how can I be sure that I really do have my recipient's *real* public key, and that it isn't someone who has usurped his or her public key to be able to intercept my messages, while giving me a false impression of security?

31.5.1 The attack of the monster in the middle

Let's take the example of Ana, who wishes to receive an encrypted message from Bea, in the presence of an adversary, Carol, who may have access to the messages exchanged:

- Ana starts by sending Bea her public key. Carole can read it.
- Bea encrypts her message with the public key she has received, then sends it to Ana.
- Carol, who doesn't have Ana's private key, but only his public key, can't decrypt the message.
- Ana can decrypt the message using the private key she keeps in a safe place.

However, if Carole is able to change the exchanges between Ana and Bea, things get complicated:

- When Ana sends her public key to Bea, Carole intercepts it and sends back to Bea, in place of Ana's, a public key for which she holds the corresponding private key.
- Bea encrypts her message with the public key she received, then sends it to Ana. But the key she received belonged to Carole: she substituted it for Ana's.
- Carole intercepts the message again. But this time, it's encrypted with her public key, of which she has the private key. She can therefore decrypt the message to read it and modify it if necessary. She then encrypts the message again with Ana's real public key, before sending it to Ana.
- Ana can then decrypt the message with her private key, without realizing anything.

For example, Bea is convinced that she is using Ana's key, when in fact she is using Carole's key. In the same way, Carole can usurp Bea's public key and forge the signature on the message sent by Bea to Ana. Ana will receive an encrypted message duly signed... by Carole.

This attack is known as the *monster-in-the-middle attack (MitM)*.⁶ In our example, Carole was the *monster in the middle*, able to read and modify the encrypted communication by pretending to be the other party to the communication.

An adversary can position herself as a *monster in the middle* by various means.

The Internet Service Provider (ISP), for example, is particularly well placed, as all traffic will inevitably pass through it. Similarly, a *large* network node through which a significant amount of traffic passes will be well placed to carry out this attack.⁷ Finally, an adversary with access to the local network you're using can always route network traffic through his computer, using more specific techniques.⁸

To guard against this attack, Bea must have a way of verifying that the public key she is using is actually Ana's key. While the public key is not confidential information, it is important to ensure its *authenticity* before using it.

Sometimes, the easiest way for Bea is to meet Ana to verify that the public key she has is really hers. It doesn't matter if Carole is present

6. Common usage is to speak of *a man-in-the-middle attack* [https://fr.wikipedia.org/wiki/Attaque_del%27homme_du_milieu]. The hacktivist community questions the inclusivity of this concept, using alternative expressions: *man-in-the-middle*, *person-in-the-middle*, *machine-in-the-middle*, *monster-in-the-middle* [<https://sindominio.net/sincensura/fr/post/informe/#inspection-ap-profondie-des-paquets>], etc.

7. Pixellibre.net, 2011, *#OpSyria: the evidence speaks for itself*. [<https://pixellibre.net/2011/10/opsyria-bluecoat-censure-leaks-censorship-syrie/>], or more precisely, in English Jakub Dalek and Adam Senft, 2011, *Behind Blue Coat, Investigations of commercial filtering in Syria and*

Burma, The Citizen Lab [<https://citizenlab.ca/2011/11/behind-blue-coat/>].

8. Wikipedia, 2014, *ARP poisoning* [https://fr.wikipedia.org/wiki/ARP_poisoning].

at the time of this meeting: only a *public* key verification will take place, and no secrets will be exchanged (except that Bea and Ana wish to communicate, but given her position, Carole may know this in other ways). Once this verification is complete, end-to-end encryption can be set up between Ana and Bea.

Between the two, a message whose content will be encrypted will circulate; only the header of the communication, whether an HTTP request or an email, will circulate *in clear text*.

page 217

However, it's often the case that Bea can't meet Ana - especially if she doesn't know her: if she meets someone who introduces herself as Ana, Bea can't be sure it's really Ana. This is usually the case when you want to encrypt your connections to a website: you don't know the people behind it.

31.5.2 Hierarchical Public Key Infrastructure

The first commonly used solution is to have trusted authorities certify public keys by digitally signing them: these are known as *certificates*. Ana asks the authority to certify her public key. The authority verifies Ana's identity, for example by asking for her identity card, and then digitally signs her key. Before using Ana's key, Bea (or her computer) checks that it has been signed by an authority she considers trustworthy. This is known as *hierarchical public key infrastructure* (or *hierarchical PKI*).

page 252

The TLS protocol

This is the principle commonly used to authenticate websites or mail servers with which the computer establishes an encrypted connection. The most common reasons for establishing an encrypted connection to a web site are to protect passwords - when logging into an e-mail account, for example - or to protect bank details - when shopping online. The protocol used for this type of encryption is called TLS (formerly SSL).⁹

page 208

This standard encapsulates the usual application protocol in an encryption layer. For example, the HTTP web protocol, when encapsulated in TLS and thus encrypted, is called HTTPS. The same applies to POPS, IMAPS and SMTPS e-mail protocols.

page 199

page 201

The TLS protocol can be explained as a very cordial greeting between the originating computer and the destination server. They will encrypt the communication by exchanging encryption keys.

The problem of certification authorities

Certification Authorities (CAs) will ensure that these encryption keys are the correct ones, and will produce an *electronic certificate* for this purpose. However, such a solution only shifts the problem: you have to trust the certification authority. In general, these are commercial companies, and more rarely public authorities.

Microsoft, Apple and Mozilla each include government certification authorities among the certification authorities recognized by their web browsers.¹⁰ Mozilla Firefox includes¹¹ government certification authorities

9. SSL for Secure Sockets Layer is the predecessor of TLS for Transport Layer Security.

10. Christopher Soghoian, Sid Stamm, 2011, *Certified Lies: Detecting and Defeating Government Interception Attacks Against SSL*, Financial Cryptography and Data Security [<https://s3.amazonaws.com/files.cloudprivacy.net/ssl-mitm.pdf>].

11. Common CA Database, 2017, *CA Certificates In Firefox* [<https://ccadb-public.secure.force.com/mozilla/CACertificatesInFirefoxReport>].

(Chinese, Catalan, Spanish, Dutch, Turkish), certification companies (Entrust, GoDaddy, Verisign), and telecommunications companies (Amazon, Deutsche Telekom, Google).

Firefox also includes the authority of the Internet Security Research Group¹²). This group has set up Let's Encrypt¹³ a free, open and automated certificate authority launched in 2016 that simplifies access to valid electronic certificates for small servers.

[page 254] But governments, which can often position themselves as *monsters in the middle*, have the power to designate any certificate as valid for a web site by signing it with their certification authority: web browsers that include it would see nothing of it.

[page 235] In the case of companies, their primary aim is not to certify identities, but to make money by selling identity certification as a service. But verifying an identity is expensive. What proof do we have that they're doing it properly? That the private keys they use to sign are stored in a safe place? Once again, it's a question of trust. We can only hope that, if only to maintain their activity, these certification authorities do their job properly...

Except that... examples show that they sometimes do it very badly. In 2008, for example, researchers succeeded in creating forged "valid" certificates, as six certification authorities were still using cryptographic algorithms that had been known to be broken since 2004.¹⁴ Certificates created in this way are "true-false" certificates: the web browser recognizes them as real, because despite their fraudulent origin, everything suggests that they have been issued by a recognized authority.

In 2011, nine fake certificates signed by Comodo, a certification authority, were created. At least one of these certificates was reportedly used on the web¹⁵. The company took more than a week to publicly acknowledge this compromise - and many companies probably don't do this in such situations, to avoid the bad publicity¹⁶ and the financial losses that go with it.

[page 254] Furthermore, it seems that if ordered to do so by the police or the courts in their country, some certification authorities give the cops fake certificates, issued in the name of entities they wish to monitor.¹⁷ That said, these fake certificates have to be set up in the right place on the Internet and combined with attacks by the *monster in the middle*, in order to be exploited to the full. Finally, as our connections generally pass through several countries, this attack can be deployed by a country other than the one from which we are connecting.

[page 208] In a sales brochure, Packet Forensics, a U.S. company that sells network surveillance equipment, writes that "to use our product in this scenario, government users have the option of importing a copy of a legitimate key that they can obtain (potentially through a court requisition)".¹⁸ The CEO of Packet Forensics reportedly confirmed orally to the author of

12. <https://www.abetterinternet.org/about/>

13. <https://letsencrypt.org/fr/about/>

14. Alexander Sotirov *et al.*, 2008, *MD5 considered harmful today - Creating a rogue CA certificate* [<https://www.win.tue.nl/hashclash/rogue-ca/>].

15. Comodo, 2011, *Comodo Fraud Incident* [<https://www.comodo.com/Comodo-Fraud-Incident-2011-03-23.html>].

16. Jacob Appelbaum, 2011, *Detecting Certificate Authority compromises and web browser collusion* [<https://blog.torproject.org/detecting-certificate-authority-compromises-and-web-browser-collusion>].

17. Christopher Soghoian, Sid Stamm, 2011, *Certified Lies: Detecting and Defeating Government Interception Attacks Against SSL*, Financial Cryptography and Data Security [<https://s3.amazonaws.com/files.cloudprivacy.net/ssl-mitm.pdf>].

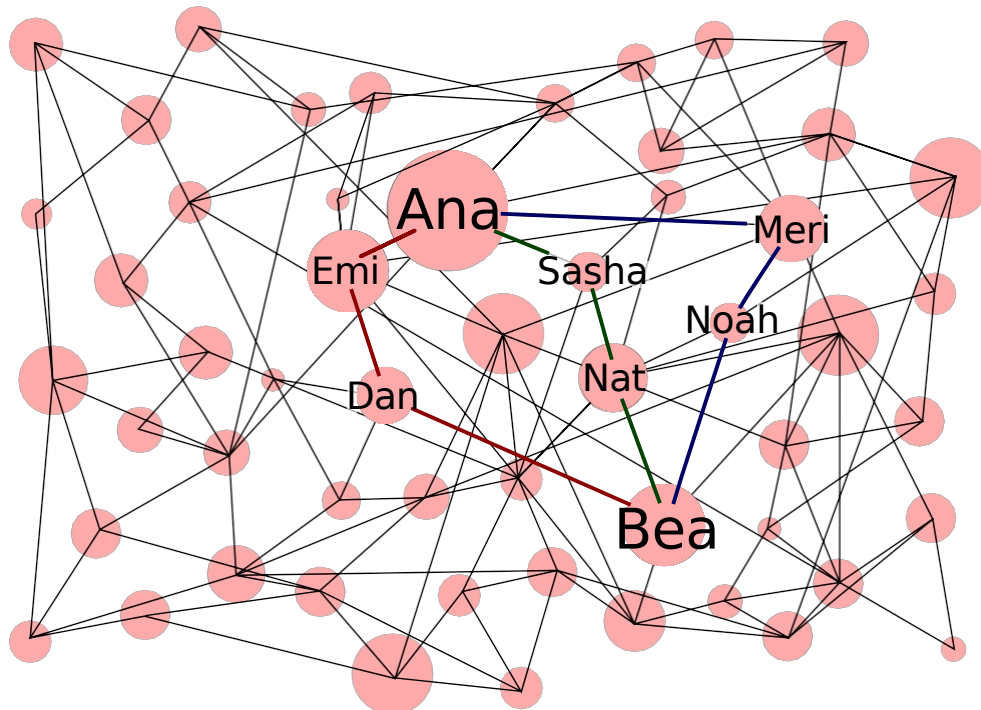
18. "To use our product in this scenario, government users have the ability to import a copy of any legitimate key they obtain (potentially by court order)". Quote from the paper by Christopher Soghoian and Sid Stamm quoted above.

the study of government clients collaborating with certification authorities to obtain fake certificates for use in surveillance operations ¹⁹.

31.5.3 Web of trust

Another solution to the question of the authenticity of public keys is the *web of trust*.

Rather than trusting a few centralized authorities, it's all about establishing a bond of trust from person to person. Bea doesn't know Ana, but she knows Dan, who knows Emi, who knows Ana. So there's a *path of trust* between Bea and Ana. If there were only this path of trust, it would imply that Bea places a high level of trust in Emi, whom she doesn't know directly. But Bea also knows Nat, who knows Sasha, who also knows Ana. She also knows Noah, who knows Meri, who knows Ana. So there are three paths of trust between Ana and Bea, who doesn't need to have total trust in each of the parties involved in the certification.



A web of trust linking Ana and Bea

These webs of trust are commonly used to authenticate software and personal communications, such as e-mails, using the OpenPGP standard. Unfortunately, they are not commonly used to authenticate websites, although this is technically possible. ²⁰

Webs of trust therefore make it possible to guard against attacks by the monster in the middle, without having to rely on centralized authorities. However, participating in a web of trust requires revealing links between people, networks of friends or activists. Do we really want to publish the list of our friends or comrades?

page 254

19. This quote can be found in a draft version, dated April 2010, of the paper by Christopher Soghoian and Sid Stamm quoted above; this version is available on [cryptome.org](https://cryptome.org/ssl-mitm.pdf) [https://cryptome.org/ssl-mitm.pdf] (in English).

20. For example, the [Monkeysphere](https://manpages.debian.org/bullseye/monkeysphere/monkeysphere.1.en.html) project [https://manpages.debian.org/bullseye/monkeysphere/monkeysphere.1.en.html] extends the use of OpenPGP webs of trust to website authentication.

31.6 Persistent confidentiality

[page 249] As we've seen, anyone in possession of a secret key can use it to decrypt a text that has been encrypted using the public key associated with it. This is a very useful property, but in some cases it can prove embarrassing.

Let's say an ill-intentioned person records an encrypted online conversation between two people. Of course, they won't be able to read the content of the conversation immediately. However, he or she may then have the idea of breaking into these people's homes or computers and getting hold of their private keys. In this case, it will be able to read, *a posteriori*, all the past conversations it has stored.

This was the case a few years ago, when the admins of the *autistici.org* server realized during a trial that the police had got hold of the secret keys installed on their server, because they were producing e-mail exchanges for the record that they should not normally have been able to read.²¹

To prevent a secret from compromising many other secrets that depend on it (such as the content of encrypted instant messaging conversations, e-mail exchanges, *etc.*), some software packages include so-called persistent confidentiality functions²² (or *Perfect Forward Secrecy*) functions.

They ensure that even if one day a long-term secret, typically a private key, is discovered by an adversary, exchanges will be protected from a posteriori analysis.

In fact, instead of directly using the public key to encrypt communications, this type of encryption uses a secret exchange protocol designed to work even over an insecure communication channel, by negotiating a temporary key at each communication session. In this case, the secret key of a key pair only serves to ensure that the right person is being contacted, by signing the secret exchange.

] This temporary secret is then used to symmetrically encrypt communications.

[page 55] Once the communication has ended, the software involved suffit to forget this temporary secret. Even if someone were to get their hands on the secret keys of both parties, the confidentiality of the communication would not be compromised: the participants in the exchange themselves no longer have access to them.

To protect the privacy of Internet users, the TLS protocol implements persistent confidentiality.

31.7 Summary and limitations

Asymmetrical cryptography is therefore a good complement to symmetrical cryptography whenever we need to protect not just our data, but also the content of our communications: e-mail exchanges, web browsing, instant messaging conversations *and so on*. It's not as complicated to use as you might think, and making encryption a routine means that particularly sensitive information is not lost in the shuffle.

] To conclude this little tour of cryptographic techniques, it's worth remembering that encryption, however difficult to break, has limits, which we touched on in the first volume of this guide. These limits relate in particular to the trust we place in

21. Austitci, 2005, *CRACKDOWN, violato autistici.org - some legal notes* [https://www.autistic i.org/ai/crackdown/legal_en.html].

22. Wikipedia, 2014, *Persistent privacy* [https://fr.wikipedia.org/wiki/Confidentialit%C3%A9_persistente].

in the computer and software to which encryption and decryption (and therefore *plaintext*) is entrusted. They also affect the legal obligations to provide authorities the means to decrypt communications when requested. Finally, page 50 deals with the evolution of cryptography: what is secure today may not be secure tomorrow.

Finally, while encryption hides the content of the communication, the parties involved (who is communicating with whom) remain apparent.

Tor or onion routing

We have seen that it is possible to hide the content of communications through encryption. However, it is always possible for adversaries to determine the source and destination of communications. Tor solves this problem.

32.1 The problem: hiding origin and destination

Paper letters affix the recipient's address and the sender's address. Similarly, on the Internet, every packet contains a source IP address (sender) and a destination IP address (receiver). The servers you connect to can therefore tell where the packet is coming from. Even when the data is encrypted, the addresses remain visible. [page 258]

The route between the sender and the recipient involves multiple routers. Each of these routers inspects the destination IP address and forwards the packet to the nearest neighboring router. In this way, they can see that the sender is communicating with the recipient, just as the postmen can see where a parcel has come from and where it's going. [page 208]

In particular, the Internet service provider is able to make an exhaustive profile of its subscribers' Internet use. Similarly, all the routers on the Internet that see our packets passing through can profile our behavior.¹

32.2 One solution: Tor

Tor stands for *The Onion Router*. It is a free page 39 software. []

In general, Tor tries to solve three privacy problems² :

Firstly, Tor prevents websites and other services from learning the location of Internet users, which they can use to build up databases of their habits and interests. With Tor, Internet connections don't reveal personal data by default.

Secondly, Tor prevents people from spying on traffic locally (like ISPs or an adversary with access to home Wi-Fi) and seeing what information is being accessed on which servers. It also prevents them from restricting what can be viewed and published.

1. Most of this section is adapted from the Tor website: *What protection does Tor provide?* . [<https://support.torproject.org/fr/about/protections/>].

2. This section is taken from the Tor Project website: *What protection does Tor provide? Tor ?* [<https://support.torproject.org/fr/#protections>].

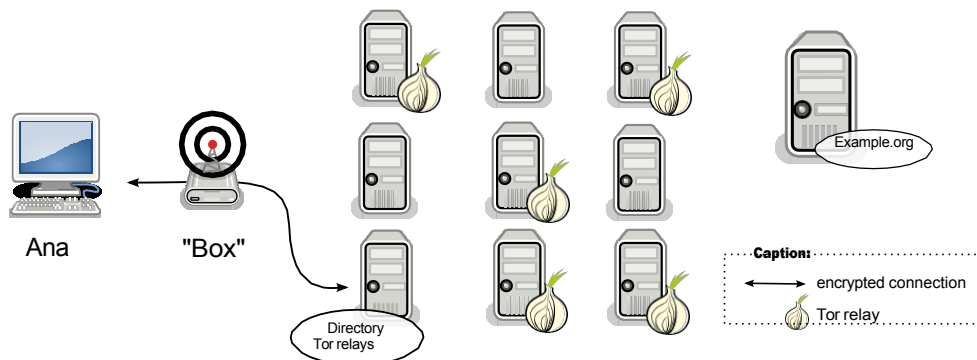
Thirdly, Tor routes connections through several Tor relays, so no single Tor relay can know what is being done. Because these relays are operated by different organizations or individuals, trust distribution provides more security than a simple VPN.

page 227

32.2.1 Creating a circuit

Instead of taking a direct route from source to destination, data packets follow a path through a number of relays, chosen in part at random.³ This makes it impossible for adversaries to associate source and destination by observing a single point.

For example, when Ana wants to connect to *example.org* using Tor, her computer first establishes a Tor circuit.



Connecting to a Tor relay directory

To do this:

1. Tor retrieves a list of available Tor nodes from a directory;
2. Tor chooses a first relay from this list, then establishes an encrypted connection with it;
3. Tor chooses a second relay from the list and establishes an encrypted connection to this second relay via the first relay ;
4. finally, Tor chooses a third relay from the list, called the exit node, and establishes an encrypted connection to this third relay via the first and second relays.

This set of three relays forms what is known as a *Tor circuit*.

32.2.2 Circuit operation

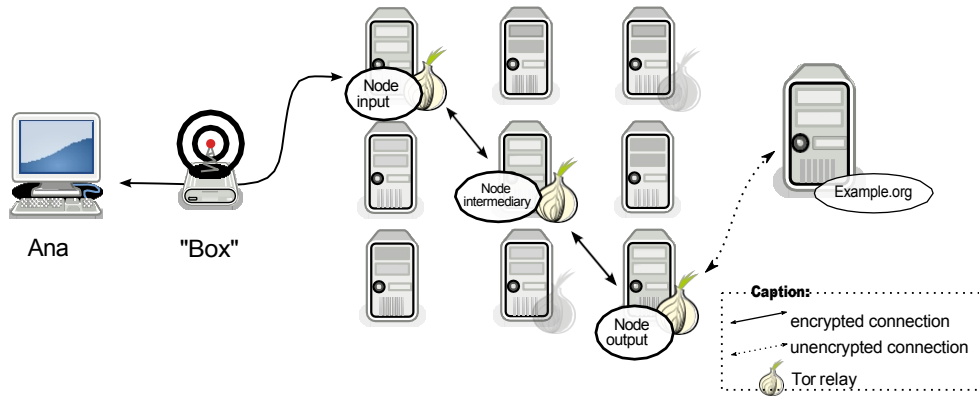
The data then passes through these three relays before reaching the destination server (in this case *example.org*). The server's response will follow the same path, in reverse.

The circuit is traversed step by step, and each relay along the way knows only the one that transmitted the data to it, and the one to which it will retransmit it. No single relay knows the complete path taken by a data packet. A possible intermediary or compromised relay cannot easily analyze network traffic to establish a relationship between the source and destination of a connection. So none of the computers knows that Ana's machine is connecting to *example.org*.

Note that a Tor circuit is made up of three intermediaries. If the circuit were made up of a single relay, compromising it would suff to jeopardize our

page 235

3. The choice of relays is made by obeying various constraints that are listed in the Tor specification: Roger Dingledine, Nick Mathewson, 2021, *Tor Path Specification* [<https://gitweb.torproject.org/torspec.git/tree/path-spec.txt>], section "2.2. Path selection and constraints".



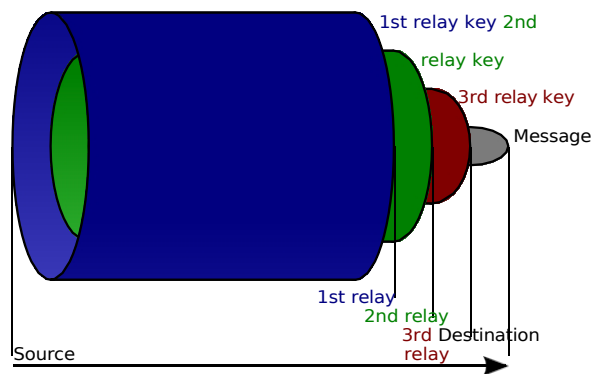
Using a Tor circuit

confidentiality, as this intermediary would be aware of both the origin of a communication and its destination. The use of three relays avoids this overlap without slowing down the connection too much. Apart from the exit nodes, no relay can know the content of the communications it carries.

Exit nodes" differ from other Tor relays in two ways: they're the only ones that can potentially see traffic in the clear (if Tor users don't use HTTPS, for example), and they're the only ones that are exposed on the Internet. In other words, traffic from people using Tor seems to come f r o m these exit nodes. Also, the people running exit nodes are sometimes considered responsible for the traffic passing through that node, and have to explain why. ⁴.

As an added precaution, the Tor circuit used is automatically changed e v e r y ten minutes without activity. ⁵.

32.2.3 Onion encryption



Onion routing principle

We've seen that Ana's computer negotiates an encrypted connection with each relay on the circuit used. As a result, the data she wishes to transmit to *example.org* has several layers of encryption at the output of her computer:

- encryption of the connection to the first relay ;

4. Nos oignons, 2020, *Rapport Moral* [https://nos-oignons.net/Association/Rapport_moral_2019-2020.pdf] p. 5, section Abuses.

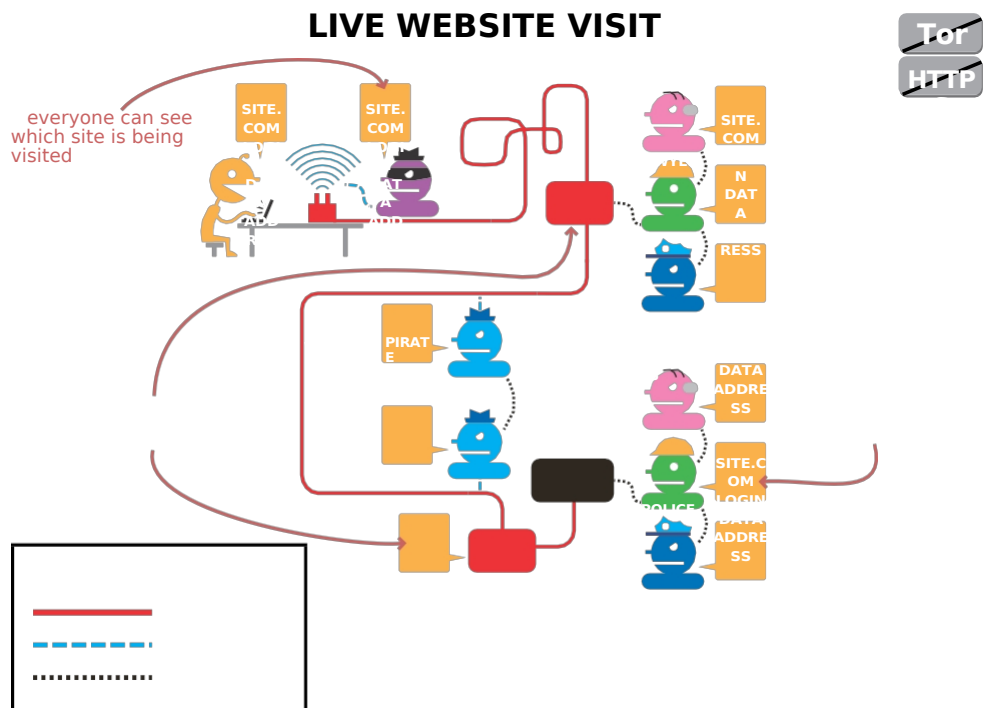
5. Tor Project, 2021, *How often does Tor change its paths?* [https://support.torproject.org/en/about/change-paths/].

- encrypting the connection to the second relay ;
- connection encryption to the third relay.

Like an onion with several skins, Ana's data will be *wrapped* in several layers of encryption. This is why it can be described as *an onion's chif-frement*. Each time it passes through a relay, a layer of encryption *is removed*. Each layer is encrypted so that it can only be read by the relay that has to remove it. This means that none of the relays can decrypt information that is not intended for them.

32.2.4 Tor illustrated

Here are four diagrams that illustrate what third parties can see or spy on, depending on whether or not HTTPS is used, and whether or not Tor is used when visiting a website.



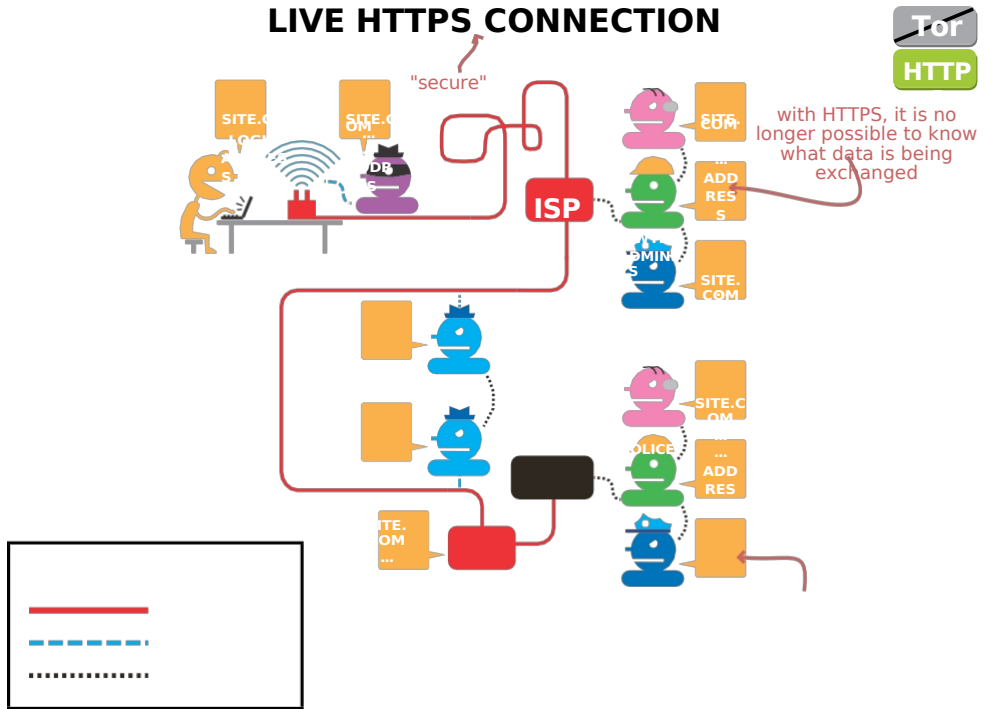
everyone can see the data exchanged

the site knows where our connection comes from

LEGEND

- Internet connection
- Listen
- Data sharing

LIVE HTTPS CONNECTION



LEGEND

- Internet connection
- Listen
- Data sharing

the site can still locate us

HTTPS live connection

WEBSITE VISITS PER TOR



b
u
t

h
e

c
a
n

s
e
e

t
h
a
t

w
e

u
s
e

T
o
r

but it's the
only node
that knows it

the site no
longer knows
where our
connection
comes from

the output node can
observe
exchanged data

LEGEND

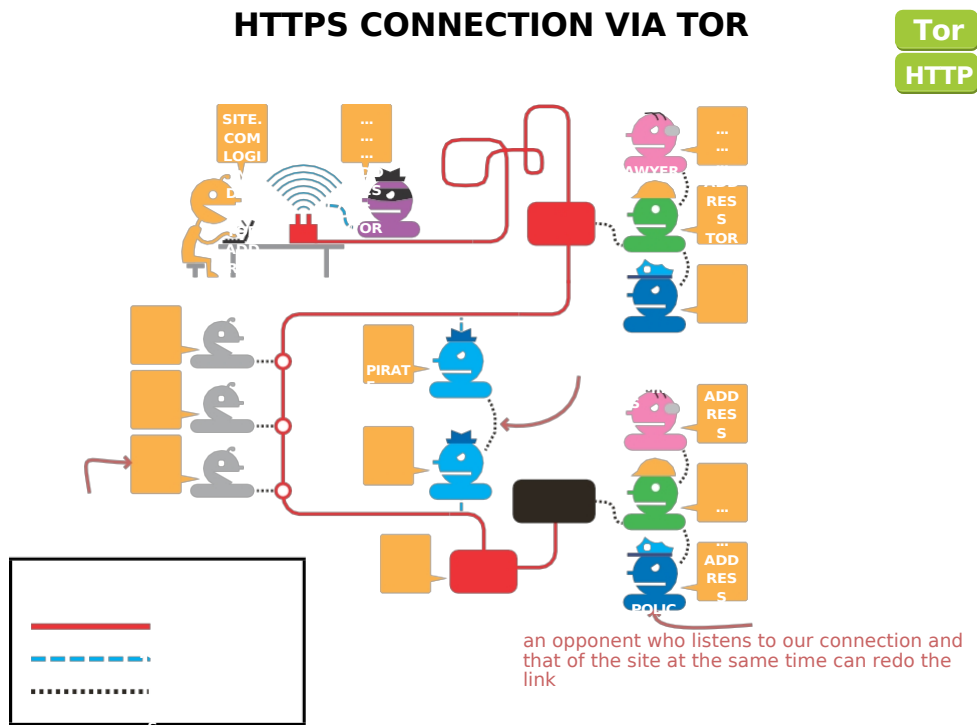
Internet connection

Listen

Data sharing

on the other hand,
it knows that Tor has
been used

Visiting a website via Tor



with HTTPS, the exit node only knows the destination

LEGEND

Internet connection

Listen

Data sharing

the police can also seize the server rather than asking admins

HTTPS connection via Tor

32.3 onion services

If you want to provide services (such as a web site or an instan- tized mail server) without disclosing the server's address (IP), you can use an onion service. ⁶ In the same way as for every Tor user, the IP address of the server set up is not revealed. Anyone wishing to connect to it will have to use the Tor network. In this way, onion services protect the confidentiality of both the server and the people using it.

To connect to it, Internet users will use Tor's "rendezvous point" system. The "rendezvous point" is the third relay for each of the two protagonists in the exchange: the client and the onion service. The customer builds a Tor circuit

with this "rendezvous point" as the third relay. The onion service does the same. Customer and onion service then "meet" and can exchange information.

These onion services can, for example, be used to set up a website without fear of censorship. Identifying the physical location of the server (or of the people publishing or visiting the website), is indeed made much more difficult than with a conventional website: it requires setting up an attack on the Tor network.

[page 268]

32.4 Participate in the Tor network

The Tor network is voluntary. It's open to everyone, since no relay can know where communications are coming from or going to. So anyone can run a Tor relay on the computer of their choice. This relay joins the public network and relays the traffic of people using this network.

6. [The Tor Project, 2013, How do onion services work?](https://community.torproject.org/onion-services/overview/) [https://community.torproject.org/onion-services/overview/]. These onion services have *.onion* addresses.

32.4.1 Setting up a Tor relay

The fact that anyone can set up a relay introduces diversity, thus reinforcing the efficiency of the Tor network as a whole.

Tor relays are legally considered routers⁷ and are therefore not page 29 This is a good thing, because if adversaries had access to the logs of [multiple] Tor relays, they would be able to discover the circuits a posteriori.

As we saw above, people who run exit nodes are sometimes considered responsible for the traffic passing through that node, and have to explain themselves to the authorities. This is why, to avoid exposing yourself individually, it's best to configure Tor so that your relay can't be an exit node, and to contribute to an association dedicated to exitnode management.⁸

32.4.2 Building a Tor bridge

It is also very useful to set up "bridges" or "bridges" Tor⁹. These are special relays that are not listed in the public directories¹⁰ of the Tor network. They can enable people whose ISPs filter connections to Tor to connect to the network anyway.

32.5 Some limitations of Tor

Tor can easily give a false impression of security. It does indeed meet the need to hide one's IP address and to conceal which servers one is communicating with. But Tor doesn't solve every problem¹¹ :

1. Tor won't protect you if you don't use it properly;
2. Even if you configure and use Tor correctly, there are still potential attacks that can compromise the protection Tor provides;
3. No anonymization system is perfect yet, and Tor is no exception: it would be unwise to rely solely on the Tor network if you need absolute confidentiality.

Let's take a closer look at some of these limitations.

32.5.1 The uninformed or uncaring person

When you're uninformed, there's a good chance you'll make a mistake. When using a tool, it's vital to understand not only what it's for, but also what it's not for, as well as its limitations.

For example, if we use the Tor Browser to fill in web forms with personal information, the website won't know our original location, but it will have all the information on the form. So we won't be totally anonymous.

7. Our onions, 2013, *What is it?* [https://nos-oignons.net/%C3%80_propos/index.fr.html].

8. Our onions, 2013, *What is it?* [https://nos-oignons.net/%C3%80_propos/index.fr.html].

9. To understand and use Tor bridges, take a look at the [Tails documentation, Connecting to the Tor network](https://tails.boum.org/doc/anonymous_internet/tor/index.fr.html#index1h1) [https://tails.boum.org/doc/anonymous_internet/tor/index.fr.html#index1h1] and the Tor project's dedicated Tor bridges page [<https://bridges.torproject.org/?lang=fr>].

10. Tor bridge addresses can be obtained by visiting [BridgeDB](https://bridges.torproject.org/?lang=en) [<https://bridges.torproject.org/?lang=en>].

11. For more information, see the Tor website: [Tor Project, 2021, Am I completely anonymous if I use Tor?](https://support.torproject.org/fr/faq/staying-anonymous/) [<https://support.torproject.org/fr/faq/staying-anonymous/>].

Beware of the documents you download. They may contain "Internet resources" (images, videos, etc.) that could reveal our IP address if we open them with an application that is not configured to connect with Tor (a PDF viewer, for example). To avoid exposure, you can open downloaded documents either with Tails, which connects to the Internet only *via* Tor, or with a computer disconnected from the Internet.

32.5.2 Adversaries see that we use Tor

Ana's ISP or LAN admin can easily tell that she's connecting to a Tor relay, not an ordinary web server.¹² In fact, the IP list of Tor entry nodes is publicly available on the Internet. The use of Tor *bridges* allows greater discretion vis-à-vis the ISP or LAN admin.

Similarly, the list of exit nodes in the Tor network is public. Website admins, who can see the origin of incoming visits, will therefore be able to identify those coming from a Tor relay.

Tor doesn't protect by making Tor users look like any random person on the Internet, but by making all Tor users look alike. It becomes impossible to tell who's who among them. The more people who use Tor, and the more varied their activities, the less incriminating the use of Tor will be. The network's strength lies in this indistinguishable set of users - the *anonymity set*.

32.5.3 Tor exit nodes can spy on the communications they relay

Tor does not encrypt communications outside its own network. So Tor can't encrypt what passes between the exit node and the destination server. Any exit node can therefore capture the traffic it relays to the Internet.¹³

For example, in 2007, a computer security researcher intercepted thousands of private e-mails sent by embassies and NGOs around the world by eavesdropping on traffic from the exit node he was administering¹⁴ with a "monster in the middle" attack.

To protect against such attacks, it is necessary to use end-to-end encryption, as described in the section on asymmetric encryption.

32.5.4 Time pattern attack

Tor's design doesn't protect against certain types of attack, especially traffic analysis attacks.¹⁵ One such attack is the "temporal pattern" attack. The idea behind this attack is to observe the rate at which data is sent at two points along its path, for example at the first relay and at the third relay (exit node). For example, let's send a stream like Morse code: three packets sent in a burst, then five seconds of silence, then three packets, *and so on*.

12. This and the following sections are heavily inspired by the Tails website [<https://tails.boum.org/doc/about/warnings/index.en.html#doc-about-warnings.fr.tor>].

13. Tor Project, 2021, *When I use Tor, can electronic eavesdroppers still see me? information I share with websites, such as login information, and what I type into forms?* [<https://support.torproject.org/fr/https/https-1/>].

14. Kim Zetter, 2007, *Rogue Nodes Turn Tor Anonymizer Into Eavesdropper's Paradise* [https://www.wired.com/politics/security/news/2007/09/embassy_hacks].

15. Wikipedia, 2014, *Traffic analysis attack* [https://fr.wikipedia.org/wiki/Attaque_par_trafic_analysis].

Opponents who see that Ana's computer is sending a stream with a given time pattern on the first relay, and who observe a stream with this same pattern on the output node that goes to *example.org*, can deduce that it is probably Ana's computer that is connected to *example.org*.¹⁶

The strength, but also the weakness, of Tor is that anyone can not only use it, but also administer a Tor relay: Ana, Bea, a university, the CIA, *and so on*. If adversaries only have access to information from one of the relays through which the data passes, no problem. If, unfortunately, cooperating adversaries have access to several relays, they can carry out a "temporal pattern" attack.

Internet Service Providers (ISPs) and large providers of content or resources used on many websites - advertising inserts, search and social media functionalities - are also in a good position to observe traffic and thus collaborate in this type of attack.

32.5.5 Tor does not protect against confirmation attacks

We've just seen that Tor's design doesn't protect against adversaries who can measure the traffic flowing in and out of the Tor network. For if adversaries can compare the two flows, it is possible to correlate them via basic statistics.

Now let's consider some opponents who have reason to believe that Ana is the one posting on this anonymous blog. To confirm their hypothesis, they can observe the traffic leaving Ana's fiber connection and the traffic entering the server hosting the blog. If they observe the same data patterns when comparing these two types of traffic, they'll be able to confirm their hypothesis.

Tor protects Ana against adversaries trying to determine who is posting on the anonymous blog. But it doesn't protect against adversaries with more resources who try to confirm a hypothesis by monitoring the right places in the network and then making the correlation.

This type of attack can also be carried out with broader assumptions.

Let's consider adversaries who suspect a group of ADSL connections of connecting to an anonymous blog on which the authors only post via Tor. Let's imagine that these adversaries have access both to the traffic of the group of ADSL connections in question, and to that of the server - thanks to a requisition or a black box, for example.¹⁷ These adversaries can then use a "temporal pattern" attack to find out which connection in the suspect group is responsible for which connection to the server. In this way, the publication of a blog post can be correlated with a connection among a group of people suspected of participating in this anonymous blog.

page 228

32.5.6 Tor doesn't protect against a global organization

An adverse global organization is an entity capable of analyzing the traffic between all the computers on a network. For example, by studying the volume of information flowing through the network at any given moment, it would be statistically possible to identify a Tor circuit, since the same flow of information would appear every few milliseconds at every node on the circuit. The adversary could thus make the link between a Tor user and her destination server.

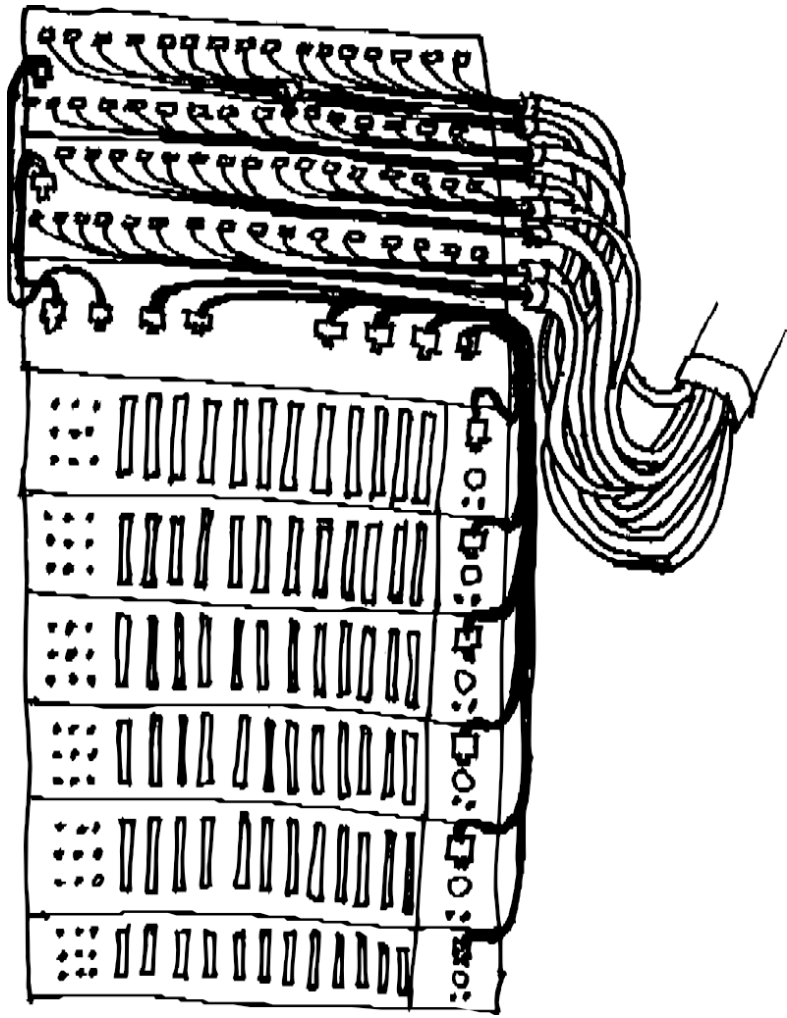
16. See [Wikipedia, 2014, *Tor \(network\)*](https://fr.wikipedia.org/wiki/Tor_(r%C3%A9seau)) [https://fr.wikipedia.org/wiki/Tor_(r%C3%A9seau)].

17. In this case, we're referring to the system that enables intelligence services to automatically analyze and-metadata of Internet communications in France ([Le Monde, 2017, *Une première "black box" of the intelligence law now active*](https://www.lemonde.fr/pixels/article/2017/11/14/les-boites-noires-de-la-loi-sur-le-renseignement-sont-desormais-actives_5214596_4408996.html) [https://www.lemonde.fr/pixels/article/2017/11/14/les-boites-noires-de-la-loi-sur-le-renseignement-sont-desormais-actives_5214596_4408996.html]).

An opposing global organization with resources comparable to those of the NSA, for example, could also set up other attacks aimed at breaking the confidentiality provided by the Tor network. This is a compromise of Tor, which enables reasonable browsing times (for the web or instant messaging, for example).¹⁸

However, the risks resulting from these limitations are not comparable to those encountered when browsing without Tor. Tor is one of the most efficient privacy tools on the Internet. While these risks should be kept in mind, they should not deter us from using it wisely.

18. Roger Dingledine, Nick Mathewson, Paul Syverson, 2004, *Tor Project: The Second-Generation Onion Router* [<https://svn.torproject.org/svn/projects/design-paper/tor-design.pdf>], section "3. Design goals and assumptions".



PART FIVE

Choosing the right answers

Introduction

Don't panic! Protecting yourself is neither impossible nor too complicated. We can go slowly, concept by concept, to design our own digital self-defense strategy.

Before the Internet, we met our buddies every night on the street corner. Solutions that are still possible, but not to be forgotten. Today, we can use tools to encrypt our message and send it to the other side of the world in a matter of milliseconds.

In this section, we'll describe concrete examples, known as *use cases*, in order to propose solutions to some typical situations.

Use case: consulting websites

33.1 Context

The focus here is on consulting information available on the web: reading a periodical, following a blog, *etc.* These are all ordinary activities when you're *online*.

However, we want to carry out these activities discretely, for various reasons, among which we can quote:

- thwart surveillance or circumvent censorship, whether by a leader, a close friend or a state;
- avoid the collection and collation of personal information for commercial purposes;
- generalize the use of discretionary practices, thus protecting those who really need it by "drowning them out".

33.2 Assessing risks

33.2.1 What do we want to protect?

In this case, what matters to us first and foremost is anonymity, or at least pseudonymity: what we're trying to hide is not the content of what is consulted, but *by whom* it is consulted.

We saw earlier that using the Internet, and the Web in particular, leaves many traces of various kinds, in different places. Many of them, like small pebbles, trace a path from the resource consulted to a house, a computer, or even the person behind it. So it's these traces on the network that we want to get rid of, first and foremost the IP address. However, as the IP is necessary for the network to function properly, the strategy here will be to ensure that anyone following this trail ends up in a dead end.

You may also wish to leave no trace of your navigation on your computer, and in particular on its hard disk.

33.2.2 Who do we want to protect ourselves from?

This is an important question: depending on the answer, the appropriate security policy can vary greatly.

Internet Service Provider

Ana works for a large company and accesses the Internet via the company network. She consults her favorite blogs during working hours, but doesn't want her employer to know.

In this case, Ana wants to protect herself from the prying eyes of the people in charge of the network, in this case her company. In this case, the adversary has access to all network traffic passing through her connection, since the company acts as a biller. It does not, however, have eyes on other parts of the Internet.

Content providers

Bea is registered on a national police forum, and spends - not without malicious pleasure - a certain amount of time stirring up trouble in discussions between cops.

In this case, Bea doesn't want to make it transparent to the site hosting the forum that she's the troublemaker. As we have already seen, her IP address will be retained for varying lengths of time by the site visited. The opponent will then have access to the IP headers, as well as the HTTP headers.

[page 225]

[page 217]

[page 217]

A wide variety of opponents

Agathe regularly visits the confidential document publication site on which Bea has posted bank statements. As the subject is sensitive, she knows that the blog in question could be monitored. So she doesn't want anyone to know that she goes there.

[page 194]

The adversary here has no fixed place on the network: she can be located at Agathe's computer, at her "box", at the blog, or anywhere along the path between her computer and the blog. The opponent can also be located in several places at the same time.

[page 208]

33.3 Defining a security policy

[page

65

Let us now ask the questions set out in our methodology: -----

1. What set of practices and tools would sufficiently protect us against our adversaries?
2. Faced with such a security policy, what are the most practical angles of attack?
3. What resources are needed to exploit them?
4. Do we think that our adversaries can use these means?

33.3.1 First step: access one of our servers

[page 225]

The most practical angle of attack for the adversary: analyze the data recorded by the servers providing the connection or hosting the resources consulted.

Resources required :

- connect to the server providing the connection, if the adversary is the ISP, or collaborates with the ISP;
- connect to the server hosting the consulted resource if the adversary is the content provider, or collaborates with it.

[page 226]

If the adversary is the ISP or content provider, it will suffice to consult its connection logs. But it is also possible for other adversaries to access this information, through requisition, commercial contract, voluntary collaboration or even hacking.

[page 228]

[page 235]

Credibility of such an attack: likely if our connection or the site we visit attracts the adversary's attention.

[page 261]

Against this type of attack, an efficient solution is to use onion routing.²

1. Jacques Follorou, *Le Monde*, 2014, *Espionnage : comment Orange et les services secrets coopèrent* [https://www.lemonde.fr/international/article/2014/03/20/dgse-orange-des-liaisons-inc-estueuses_4386264_3210.html].

via the Tor network, as described below. To keep our contextual identities separate, we'll be careful not to mix our daily activities with those we want to keep more discreet. page 243

33.3.2 Second step: look at the computer you're using

If we use onion routing, the adversary observing the data circulating on the network cannot know where this data comes from, nor where it goes. So they have to find another way to get there.

The most practical angle of attack: access to the traces left on the computer by the sites visited. page 27 of

Necessary means: access to the computer used.

Credibility of such an attack: in the case of Ana using her work computer, this is very easy for her opponent. In other cases, and depending on the opponent, it requires either a burglary (also called a search, when legal), or corrupt the computer targeted by the attack, for example by installing malicious software. page 31

To guard against this attack, you need to cipher your hard disk to make any traces left behind difficult to access. Better still, avoid leaving any traces in the first place by using an amnesiac live system, which will record nothing on the computer in use. page 119
page 113

33.3.3 Step 3: Attack Tor

Angle of attack: exploit the limits of anonymity provided by Tor, for example by carrying out a confirmation attack. page 267

Necessary resources: be able to monitor several points on the network, for example the connection used and the site visited. page 269

Credibility of such an attack: an adversary such as a company seeking to monitor its female employees is unlikely to mount such an attack. Ditto for the gendarmes of Saint-Tropez. It may, however, be within the reach of a national or global network service provider, or even specialized cops. Once again, let's not forget that there is a significant difference between "having the technical capability to carry out an attack" and "actually carrying out such an attack". This difference is mainly due to the economic cost and return on investment of such a targeted attack.

Remember that many other attacks against Tor are possible or planned. Above all, it's important to understand the aims and limits of onion routing, so as not to shoot yourself in the foot. page 261

page 267

33.4 Choose from available tools

Depending on your needs and your security policy, you'll need to choose from a number of different tools.

33.4.1 Tor Browser on our system or in Tails

Tor Browser on our operating system

The Tor Browser is a software *package*: it provides a pre-configured web browser to surf confidentially, using the Tor network, from our system.

2. Against some of the adversaries listed here, other technical solutions may suffice, such as the use of a VPN [\[https://fr.wikipedia.org/wiki/R%C3%A9seau_priv%C3%A9_virtuel\]](https://fr.wikipedia.org/wiki/R%C3%A9seau_priv%C3%A9_virtuel) for example. However, onion routing protects against many more of the possible attacks than a VPN, which inserts only one intermediary between us and the resource consulted.

[page 313] usual operating ³. Once the Tor Browser is installed, you can choose to use this Tor-enabled web browser, or your usual web browser. ⁴.

Benefits The Tor Browser lets you browse the web with Tor from your usual operating system. For example, you can work on a document with our usual tools, while searching for information on the web anonymously.

Disadvantages As the Tor Browser runs on the usual operating system, this means that a vulnerability in the latter would enable adversaries to bypass the protection offered by the use of the Tor network. Above all, when used outside an amnesiac system, the Tor Browser is likely to leave traces on the hard disk of the computer in use.

The Tor Browser is based on Firefox, and there may be a delay before updates to the latter take effect. During this time, the Tor Browser will have known and published security vulnerabilities.

If the Tor Browser crashes, you can permanently lose all your current reading and searching. ⁵.

The Tor Browser does not prevent other programs from connecting to the Internet without going through Tor, even if they are opened from the Tor Browser (P2P software, PDF readers, media players, *etc.*).

Tails

[page 113] *Tails* ⁶ is a *live system* designed to protect the privacy and anonymity of its users. It lets you access the Internet anonymously from virtually anywhere, and from any computer. What's more, it leaves no trace of activities carried out on the computer, unless explicitly requested to do so.

Advantages When you use Tails, not only do you leave no trace on the computer you're using, but software that needs to access the Internet is configured to use the Tor network, and direct connections (which don't allow anonymity) are blocked.

What's more, because it's a *live system*, Tails can be booted from a DVD or USB key, without modifying the operating system installed on the computer. So you can use it at home, on someone else's computer, or even at the local library.

For more information, see the "[How Tails works](https://tails.boum.org/about/index.en.html)" page [https://tails.boum.org/about/index.en.html].

[**Disadvantages** First of all, as Tails is an operating system in its own right, you need to restart your computer to use it. ⁷. It is also more complex to install

3. In our case, it's Debian, but the Tor Browser also works with any other GNU/Linux distribution, just as it does with Windows or macOS.

4. It is possible to configure a web browser to use Tor, but this is not advisable, as even with good technical knowledge it is difficult to ensure that all browser requests will pass through Tor. The Tor Browser exists in particular to overcome this difficulty.

5. You can change this behavior by modifying Tor's default settings, settings which aim to make navigation almost amnesiac.

6. See the [Tails website](https://tails.boum.org/index.fr.html) [https://tails.boum.org/index.fr.html].

7. Tails can also be used in a virtual machine [page 163] in the usual system. In this case, the virtual machine's memory and all data will be visible to the user. In addition, if the operating system uses virtual memory (swap), data from the virtual machine may end up being written to the hard disk. What's more, if the operating system uses virtual memory (*swap*) [page 25], it's possible that data from the virtual machine will end up being written to the hard disk. It is therefore virtually impossible to guarantee the amnesia of a Tails system used in this way.

Tor Browser. Last but not least, you'll need a USB stick (with a capacity of at least 8 GB) or DVD containing Tails.

Then, because of the system's amnesia, if the web browser ever crashes, we lose all the pages we were viewing, just like with Tor Browser.

To avoid mixing your normal daily activities with those you want to be more discreet when using Tails, you need to restart your machine when switching from one contextual identity to another.

Another disadvantage of Tails is the delay between security updates for programs otherwise included in Tails, and the same software updates in Tails. This disadvantage is similar to that of the Tor Browser, in that there is a delay between Firefox updates and their inclusion in the Tor Browser.

For further information, see the [Tails "Warnings" page \[https://tails.boum.org/doc/about/warnings/index.en.html\]](https://tails.boum.org/doc/about/warnings/index.en.html).

33.4.2 Making your choice

At the end of the day, you have to choose between :

- use your usual operating system ;
- use an amnesiac *live* system.

In other words, what traces (possibly encrypted) are you prepared to leave on the computer or USB key you're using? Do you need the rest of your environment for anonymous browsing?

Once again, there's no right or wrong answer: it's a matter of choosing the solution that suits you best. What's more, it's perfectly possible to test one solution and then switch to another if necessary.

In the end, there are two possibilities:

- use the Tor Browser from an encrypted Debian. This allows you to surf almost anonymously while using your usual system. On the other hand, (encrypted) traces will probably be left on the computer's hard disk. [page 119]
- use the Tails web browser. No traces are left on the hard disk of the computer used, or even none at all if you don't use persistence. [page 116]

Once you've made your choice, see the corresponding paragraph below.

33.5 Browsing websites with Tor Browser

If, after weighing up the pros and cons, you decide to use *Tor Browser*, there are a few precautions you should take.

33.5.1 Preparing your machine and installing the Tor Browser

First of all, as we don't use a *live* system, traces of browsing (bookmarks, downloaded files, sometimes even cookies or history) will be recorded.

on our hard disk. Applying the same policy as for a fresh start is [page 71](#) a good idea. Next, we need to download and install the Tor Browser. The chapter [installing the Tor Browser](#) describes this procedure. [page 313]

33.5.2 Using Tor Browser

[page 313] The chapter on installing the Tor Browser also explains how to start it up. This tool is specially designed to be as easy to use as possible. When it's launched, all the software we need (Tor and the Tor Browser) will be started and configured.

[page 261] All we have to do is wait for the Tor Browser window to open, and we're ready to start browsing the Tor network.

Please note: only browsing *websites* via the Tor Browser window guarantees a confidential connection. All your other applications (e-mail client, instant messaging, Torrent, *etc.*) will reveal your real IP address.

[page 202] 

What's more, once this window is closed, you'll have to relaunch Tor Browser and wait for a new window to open to resume browsing via the Tor network.

33.5.3 You soon see the limits

The Tor Browser is a great tool because it's so easy to use, but you soon realize its limitations. Only connections initiated by the Tor Browser pass through the Tor network. If you want to use another web browser, the connection will no longer pass through this network, which can be annoying. If you're not careful, you can quickly get the wrong browser and think your browsing is going through the Tor network, when it's not. What's more, it only lets you use Tor to browse the web, which, even if it's widely used, is only one part of the Internet, as we explained earlier.

[page 200]

And online privacy isn't just about falsifying IP addresses. All the other traces we leave on the web and on our computer can betray us sooner or later, and the Tor Browser doesn't protect against that.

33.6 Browsing websites with Tails

33.6.1 Get and install Tails

Tails is free software and can therefore be downloaded, used and shared without restriction. It runs on a computer independently of the system already installed.

[page 39] In fact, Tails can be launched from an external medium, such as a DVD or USB key, without using the hard disk.

First we need to download Tails (see page 114). To ensure that the download has been successful, we then need to check the ISO image of the file (see page 114).

Once this has been verified, you can proceed with installation on a USB key or DVD (see page 115).

33.6.2 Start Tails

Now that Tails has been installed and restarted (see page 115), you can start using it without altering the operating system on your computer.

33.6.3 Connecting to the Internet

Once Tails has finished booting up, i.e. once the desktop has finished affichering, all that's left is to connect to the Internet: see [Tails documentation on how to "Connect to a local network"](https://tails.boum.org/doc/anonymous_internet/networkmanager/index.en.html) [https://tails.boum.org/doc/anonymous_internet/networkmanager/index.en.html]. You can then browse the web.

33.6.4 Limits

Such a solution relies on the use of Tor and Tails, and therefore inherits the limitations of both tools:

Tor's limits have already been discussed in the paragraph

"Third step: attack Tor

page 279

For Tails limits, you'll find an extensive list of warnings [on the project website \[https://tails.boum.org/doc/about/warnings/index.fr.html\]](https://tails.boum.org/doc/about/warnings/index.fr.html).

We urge you to read both documents carefully.

Use case: publishing a document

34.1 Context

Once we've finished writing a sensitive document, we'd like to publish it on page 79 of the Internet, while preserving our anonymity (the fact that it cannot be associated with any name) or our pseudonymity (the fact that it can only be associated with a chosen name).
different from our civil identity).

As a bonus, we'd like to be able to include a public contact address corresponding to this pseudonym.

34.2 Assessing risks

34.2.1 What do we want to protect?

The content of the document is public. We are not interested in its confidentiality. On the other hand, we do seek to hide the links between the document and the people who wrote it. It's **anonymity** or **pseudonymity** that interests us here.

What's more, if we make public a sensitive document whose mere consultation could be held incriminating, we must also seek to limit the possibility of identifying the people accessing it.

34.2.2 Who do we want to protect ourselves against?

As in the previous use case, our aim here is to protect ourselves from prying eyes looking to see *who's* doing *what* on the web. page 277

We'll be all the more careful about the traces left behind, given that we're talking here about publishing a document that we assume may displease one or more people with a certain power to cause trouble. It is therefore likely that a search for clues will be launched in an attempt to find the person(s) who produced the document (or those who consulted it), for example by sending requests to the host. page 228

page 209

34.3 Defining a security policy

We'll look first at publishing and consulting documents, then at using a public contact linked to them.

34.3.1 Publication

Technically, publishing a document means "saving" it on a server connected to the Internet, known as the *host*. This operation is often performed via a website. page 209
However, we won't use the same sites if we want to publish text, sound or video. page 211

We therefore need to choose our host carefully, bearing in mind the many criteria involved: type of document, availability, hosting conditions, resistance of the host to legal pressure, risks to the host posed by our document, possibility of consulting the document without risk of identification, *etc.* A more exhaustive list of these criteria is available in the "Tools" section.

page 319

Once we've made our choice, we need to be sure that our document remains available for consultation: if our host doesn't like our publication, receives pressure or even a request to remove it, our work could become unavailable.

To avoid this kind of inconvenience, you can multi-host the same file, if possible on servers located in different countries. Since putting a file online is much quicker than a legal action, this seems to be a good solution for avoiding censorship.

What angles of attack are available to a potential adversary?

First step: read the document

At first glance, the adversary has a large volume of data to search for traces of: the document content.

Thus, a possible signature such as a pseudonym or city, a date, the language in which the document is written, or even simply the theme of the document are all clues that can lead to its authors. A text describing the abusive practices of the Machinex company in November 2012 was probably written by employees of this company or by people who shared their struggle on that date.

page 245

The adversary can also attempt a stylometric analysis to compare it with other texts, anonymous or not, and try to deduce information about the authors. As far as we know, this type of attack is only really effective when there are already strong suspicions about a subset of potential authors, but this is a recent field of research. Since we want to distribute this document widely, we won't be able to hide the content. However, if you think it's necessary to go to the trouble, you may want to pay particular attention to changing your writing style.

page

Finally, if we publish our document without taking further precautions, the adversary can look for any metadata that might provide some information.

30

These different methods require no great technical skill, and are therefore within the reach of many opponents.

To protect yourself, follow these recipes:

- if possible, we will work on our document using methods that limit the amount of metadata that can be recorded;
- In all cases, it's a good idea to remove any metadata before publishing.

page 185

79

Second step: ask those who see

In the absence of easily exploitable traces inside the document, one of the most practical angles of attack is to look for traces of the publication on the network.

page 228

page 225

page 226

Depending on her powers, our adversary can requisition the content from the host, or find another way of obtaining the connection logs and thus obtain the IP address used. She can then turn to the ISP corresponding to this IP address to obtain the subscriber's name.

Here too, to cope, we'll use Tor to connect to the Internet, scrambling this trail before publishing our document.

As for the choice of hosting, the issues discussed above still apply. What's more, some of the platforms on which we'd like to deposit our document are likely not to work if Tor is used, or, like Facebook, to impose identity checks that are difficult to circumvent and incompatible with our need for anonymity: this will restrict the usable hosts.

To publish our document on a *conventional* web server, we'll start in practice by following the recipe for [finding web hosting](#).

page 319

In most cases, publication will take place via a web browser. We'll therefore follow the "web browser" path of the [previous use case](#).

page 281

It's also possible to host our document ourselves, thanks to Tor's *onion services*: they make a web server or other type of server available without having to reveal its IP address. They don't use a public address, so they can easily operate even behind a firewall or other [network address translation \(NAT\) box](#).

page 266

page 203

If you prefer to host your document on an Onion service, you'll need to follow the recipe for [using OnionShare](#).

page 205

page 359

Third step: look at the computer you're using

This angle of attack is similar to that described in the "looking at the computer in use" section of the [previous use case](#). So let's go back and read (or reread) that chapter to review it all.

page 279

Step 4: Attack Tor

In desperation, the opponent can also try to attack Tor (see section (see "Attack Tor" in the [previous use case](#)).

page 279

34.3.2 Document viewing

Another criterion to be taken into account when choosing a hosting provider is the risk we pose to those who come to consult our document. We prefer hosting providers that limit the possibility of a potential adversary being able to identify these people.

The means of attack available to the adversary are those already covered in the [previous use case](#). We will briefly repeat them here, adapting them to the case of document consultation.

page 278

First step: ask those who see

As seen in the previous use case, a person who comes to consult our document can be identified by their ISP or host, as access to the document will appear in their [connection logs](#).

page 226

To reduce this risk, we recommend that anyone wishing to access the document use the [Tor network](#). We'll also need to make sure that the chosen host is accessible via Tor, or even that it offers access via an onion service.

page 261

It's also important to choose a host that is not a platform on which people might already be authenticated and therefore be "re-known" by the host when accessing the document, even if they're using Tor. So, for example, social media or the content hosting platforms of [Web 2.0 giants](#) should be avoided.

page 266

page 239

[page 228] Finally, we can also opt for hosts who don't keep connection logs, or who refuse to hand them over to the cops in the event of a requisition. -----

Second step: look at the computer you're using

[page 278] We have little control over this situation. We can, however, advise anyone wishing to consult our document to follow the recommendations of the previous use case in this guide, so as to leave as little trace as possible on their computer.

Step 3: Attack Tor

[page 279] As with the publication of the document, the adversary can also try to attack Tor to try to identify who would view it (see section (see "Attack Tor" in the previous use case).

34.3.3 Public contact

When we publish a document, we may want people who are going to read it to be able to contact us. This contact opens up new possibilities of attack to our adversary in search of loopholes to exploit.

If we have taken every precaution to be as anonymous as possible when publishing the document, but our contact address is `nom.prenom@exemple.org`, these precautions will be pointless: the contact address gives our name directly to the adversary.

[page 243] To avoid this error, we'll make sure we have a pseudonym that will be used only for this document - or for a group of documents - depending on the contextual identity we wish to adopt.

[page 293] Your adversary will then want to know who is hiding behind this pseudonym. To try and hide "who is using this e-mail address", the use case "Exchanging e-mails while hiding your identity" can help.

[page 295] Finally, you might want to hide the content of the e-mails exchanged, but this can be very complex: insofar as you wish to have a public contact address, *accessibility* can conflict with discretion. Nevertheless, in addition to the contact e-mail address, it is always possible to specify an associated OpenPGP public key, so that anyone who wishes to send us encrypted messages can do so. The recipes for creating an OpenPGP key pair and for exporting your public key show how to do this.

[page 339] We can thus take a whole range of precautions to increase our contact's anonymity, but we can't act on the other "end of the pipe". The people who are going to contact us can then take risks by dialoguing with us, without thinking about their anonymity. Reminding them of the conditions of confidentiality and anonymity is essential. What's more, we never really know who is contacting us, so we need to be careful about what we say if we don't want to compromise ourselves.

Use case: exchanging messages

35.1 Context

We now want to exchange messages with other people, be it to wish grandma a happy new year, or to work on a sensitive document. We don't worry about the synchrony of the exchange, unlike a telephone conversation or an instant messaging dialogue. or instant messaging: in this case, we speak of *asynchronous* communication.

Another use case will be devoted to *synchronous dialog*. For now, let's concentrate on electronic mail, or email.

35.2 Assessing risks

35.2.1 What do we want to protect?

When an e-mail is sent, a variety of information is potentially revealed to our adversaries. Which information?

When we ask ourselves this question, it's often the *content* of the message that first comes to mind. While not all the messages we exchange are necessarily top-secret, some deserve more discretion than others: to avoid the details of our intimate relationships being spilled out, or because the content of a message could get us into trouble, from losing a job to going to prison. More generally, we're not overly enthusiastic about the idea of the postwoman reading all the letters we've received over the last few years today, to whet our appetites before eagerly awaiting those that will arrive tomorrow. And yet, when we exchange e-mails without taking any special precautions, intermediaries can read our communications in a totally transparent way, as if they were postcards.

Beyond the content of these postcards, it can be interesting to hide contextual information, such as the date of the exchange, the identities of the protagonists, their locations, *etc.*, which can be revealed, for example, in HTTP headers, email headers, or in the body of the message itself.

The fact that one person writes to another can in itself be a sensitive information. Indeed, relationships between people are sometimes targeted by certain forms of surveillance, for example, in order to reconstitute a network of political opponents.¹ for example. These traces generally persist in e-mail headers and connection logs.

1. Jean-Marc Manach, 2011, *Réfugiés sur écoute* [<https://web.archive.org/web/20221019100157/http://owni.fr/2011/12/01/amesys-bull-eagle-surveillance-dpi-libye-wikileaks-spyfiles-kadhafi/index.html>].

35.2.2 Who do we want to protect ourselves against?

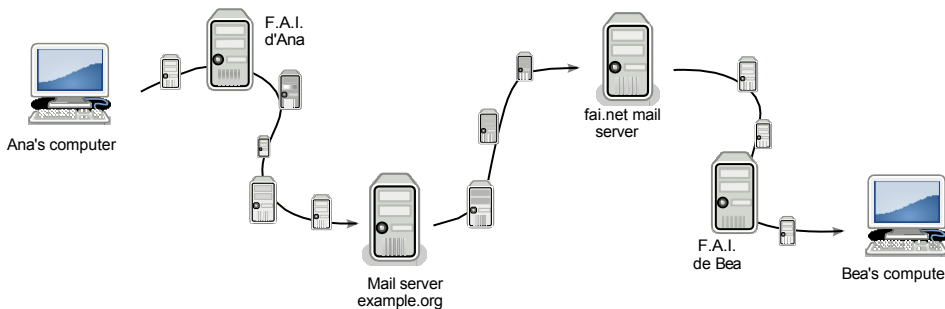
You may wish to conceal some or all of this information from the various machines that can access it, as well as from the people who have access to these machines.

[page 217] Among these machines are the servers involved. At the very least, for a message sent by Ana (*ana@example.org*) to Bea (*bea@fai.net*), these will be :

- of the server Ana uses to send the message: generally, this will be *example.org* ;
- of the server responsible for receiving messages and storing them in Bea's mailbox: *fai.net*.

[page 205] But that's not all. Numerous other computers (*routers*) are located along the route, and have access to the information they carry:

- [page 217]
- between Ana's computer and her ISP;
 - between Ana's ISP and its mail server *example.org* ;
 - between *example.org* and the Bea *fai.net* mail server;
 - when Bea checks her mailbox, the message will travel between the mail server *fai.net* and its ISP,
 - between Bea's ISP and her computer.



An email passes through many intermediaries

The people administering these machines are the first to have access to the information they process, but they don't necessarily have exclusive rights to it. This information can end up in the hands of more or less governmental hackers, with or without requisitions.

[page 235]

[page 228]

[page

27

Last but not least, every time you consult your mailbox, every time you send a message, you're likely to leave traces on the computer you're using. It may be a good idea to conceal these traces from people who might be able to take a look at the contents of our hard drives.

35.3 Two issues

We may want to protect both our identity - and even those of our recipients - and the content of our exchanges. This concerns the information contained in the two parts of our digital postcard: the text on the left and the headers on the right. This information appears throughout the course of our messages and can be the target of attacks. The security policy we define will depend in particular on the way we consult our e-mails. Indeed, its use may involve various protocols which do not have the same consequences in terms of traces.

[page 200]

35.4 Webmail or mail client?

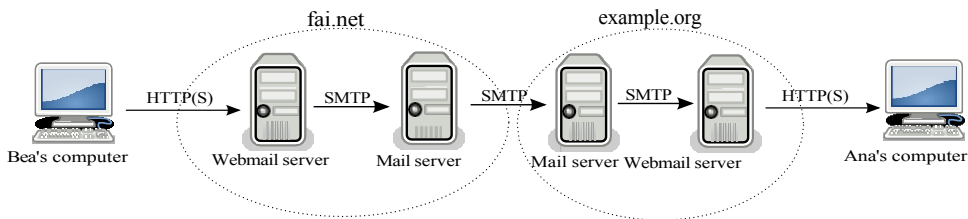
There are two ways of using email, both of which enable the same actions: using webmail or a mail client. The choice is based on various criteria, bearing in mind that

both can be used for the same e-mail address, and the choice of one or the other is not irreversible.

35.5 Webmail

Webmail is a website that lets you check your e-mail *via* a web browser. Its use has spread like wildfire since the early 2000s, to such an extent that we've almost forgotten about other ways of doing email. Hotmail and Gmail are two very popular examples of hosting providers that promote its use (even if they can only be used as webmail). Here again, we're dealing with a Web 2.0 trend: you no longer need to have your own operating system to access your mailbox (whether on your computer or on the USB stick containing a *live* system): Internet access suffit.

page 239



Bea sends an email to Ana, both of whom use webmail.

Webmail is basically a web interface that lets us act on mail servers. Let's diagram an e-mail exchange between Ana and Bea, who both use webmail:

- the "network path" between Bea's computer and her mailbox hosted by *fai.net* will be browsed using a web protocol (HTTP or HTTPS)
- followed a short stint at *fai.net*, which ensured the transition from webmail to email.
- followed by a mail protocol (SMTP) trip between *fai.net* and *example.org*
- once again, at *example.org*, between mail and web protocols
- then from the web (HTTP or HTTPS) to Ana's computer.

35.5.1 Benefits

One of the advantages of webmail, as with all web applications, is that there's no need to install, update or configure mail software. Webmail is also a key Web 2.0 feature: you can access your mailbox from any Internet-connected computer, anytime, anywhere.

If you use a *live* system and don't encrypt your e-mails, this has the advantage of leaving no traces on the disk.

35.5.2 Disadvantages

On the downside, if you're not connected, all your correspondence is inaccessible (unless you've saved all or part of it on a handy medium: USB key, hard disk, *etc.*).

page 151

The fact that it's possible to use any web browser to access our mailbox can quickly encourage us to use *any* web browser at all, and with it computers we have little reason to trust.

Then, depending on the level of trust you place in your mail host, you need to ask yourself how centralized your data is. The massive use of webmail has led to a situation where thousands of mailboxes, with all their contents, end up in the hands of the biggest mail service providers, entrusting them with the custody of a mountain of personal data. These providers can use it for commercial purposes, hand it over to various authorities, or simply lose it. What's more, if we consider our correspondence to be sensitive in one way or another, perhaps we'd prefer not to let it rest on the shoulders of people - for there are still some behind the machines - who don't particularly want to bear the responsibility. This was probably the case in August 2013 for Lavabit² which was hosting an Edward Snowden e-mail account and decided to cease operations. The closure followed requests and even pressure from government agencies such as the NSA and FBI.

Last but not least, using webmail can enable us to take full advantage of a host of publicities displayed in our web browser, when we consult our computer mailbox. Advertisements that can be selected according to the content of our e-mails.

[page 221]

35.6 Mail client

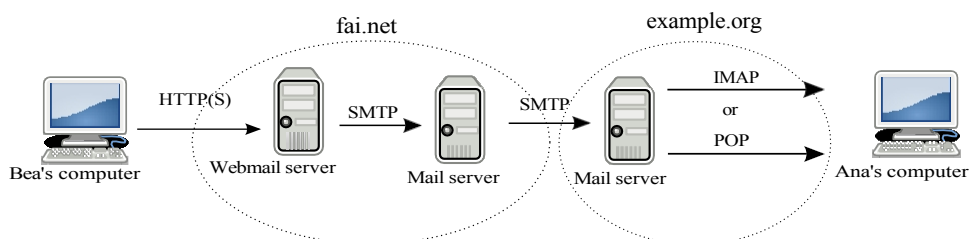
A mail client is a software application used to manage your e-mail: receive, read, send, *etc.* Well-known e-mail clients include Microsoft's Outlook and Mozilla's Thunderbird. There are many others out there which, despite their differences, have a broadly similar interface, close to that of webmails.

Unlike webmail, where you use your web browser to consult your e-mail messages stored on the host's server, here you read your e-mails using software installed on your computer. A local storage device (the computer's hard disk, USB key, *etc.*) is used to store e-mails.

To return to our previous little diagram, we need to replace web protocols with mail protocols. Two different protocols exist for receiving mail, *IMAP* (for *Internet Message Access Protocol*) and *POP* (for *Post Office Protocol*).

The first, IMAP, is used to handle e-mails stored on our host's mail servers. On each connection to the mailbox, a synchronization takes place to ensure the same status (number of emails, drafts, folders, *etc.*) on the mail server as on our mail client, and vice versa. This is achieved without downloading any content from the mail server. Only the list of emails and their headers can be downloaded to our mail client, for example.

The second protocol, POP, will download the various contents of the mailbox directly to our mail client, without necessarily leaving a copy on the remote server.



Bea sends an email to Ana, Bea uses webmail, Ana uses a mail client

2. Wikipedia, 2014, *Lavabit* [<https://fr.wikipedia.org/wiki/Lavabit>].

35.6.1 Benefits

The advantages and disadvantages may be specific to the protocol used to receive mail, but some are common to all.

With a mail client, you can retrieve your mailbox in the same state as when you last checked it, even without an Internet connection. This means you can read, write or delete e-mails offline. And to send or receive them again when a connection is restored. What's more, the use of a mail client means we don't have to endure the myriad of adverts that litter the web.

By using the POP protocol, you'll benefit from other advantages such as e-mail decentralization. Instead of leaving all our correspondence on remote servers, e-mails are repatriated to the computer. This means that we don't have to leave all our e-mails with the major e-mail hosts, and that smaller e-mail hosts don't take up too much disk space. The fact that emails end up on the recipient's system also gives us greater control over their management: for example, in terms of effectively deleting emails that could prove critical. Last but not least, less data is left to companies who don't care about the confidentiality of correspondence. A word of warning, however: the host may still make a copy of the e-mail before it is repatriated to the mail client.

35.6.2 Disadvantages

To use a mail client, you'll need to configure it so that it knows which mailbox to read, which server to connect to and which protocol to use.

It's more complicated to check your e-mail from a computer that isn't your own (at a friend's house or at work, for example), unless you use the mail client of a persistent *live* system (like Tails), installed on a USB stick.

What's more, if your mail client is configured not to leave your mail on the server, it will only be stored on your mail client's storage medium. If this is lost (whether on a computer's hard disk or on the USB key on which you've installed a persistent Live Tails system), you can say goodbye to your precious messages... unless you've made a backup.

[page 151]

35.7 Exchange emails while hiding your identity

The aim here is to conceal from an adversary the fact that we are one of the correspondents in an e-mail exchange. It could be an e-mail exchange with a wanted political dissident, or with a long-lost friend.

35.7.1 Defining a security policy

Our main concern will be to hide the names of people exchanging e-mails, or at least to make their identification as difficult as possible. What would an adversary do to find them?

First step: ask the postwomen

Our mail provider is a network node through which our digital correspondence is bound to pass. An adversary interested in this area would therefore have good reason to take a look at it, especially as it can be very easy for him to do so.

In the same way, intermediaries between Bea and Ana (including their respective ISPs) see e-mail headers, which can deliver a great deal of information (including, for some hosts, the IP addresses of the correspondents). Such an attack is

[page 218]

more than likely if the content of the e-mails or the protagonists of the exchanges attract the attention of authorities with sufficient powers. It's fair to say that, in the first place, not having an e-mail address like *nom.prenom@example.org* is already a good reflex. First of all, you'll need to think about using a pseudonym, to set up a contextual identity.

[page 243]

Having said that, if "Kiwi Poilu" writes regularly to Caroline Carot, Sofiane Carot and Francine Carot, an opponent *might* say that she belongs to the Carot family, or is part of the inner circle: the identities of the people to whom we write are also revealing.

What's more, if you use a pseudonym, but an adversary observes that the e-mails you're monitoring are coming from a particular house or apartment, they can make the connection. This is why, as with web browsing, onion routing or the use of a specially designed *live* amnesia system can be used to cover tracks back to our computer.

[page 261]

[page 113]

Finally, the content of the exchanges can reveal enough about their authors to put names to them. Hiding an identity therefore requires attention not only to the headers, but also to the content of the email.

To protect the content of emails from prying eyes, whether for itself or for what it may reveal about the authors of the emails, we use email encryption.

[opposite page]

Second step: look at the computer you're using

If the Tor network and a pseudonym are used to protect one's identity, a potential attacker can try to access the traces left on the computer to prove that the person under suspicion is indeed in possession of the e-mail account in question.

[page

27

To protect yourself against this attack, encrypt your hard disk or use an amnesiac *live* system.

[page 119]

[page 113]

This is all the more important if you're using a mail client, as it's not just traces that are left on the system, but also the e-mails themselves.

Step 3: Attack Tor

An attacker able to monitor several points on the network, such as the connection used *and* the mail host, might be able to undo the anonymity provided by the Tor network.

Let's not forget that there are many other possible attacks on the Tor network, and that it's essential to understand what it protects against and what it doesn't.

[page 261]

[page 267]

35.7.2 Choose from available tools

There are several tools available for communicating by e-mail, so the choice depends on the various criteria mentioned above. For example, you may prefer not to leave your e-mails on your host's server, but to read and reply to them offline, or you may prefer not to download a copy of your e-mails, but to access them online.

35.7.3 Webmail

As webmail is a specific use of the web, for questions relating to the Tor Browser or Tails - their advantages, disadvantages and use - please refer to the use case dealing with web browsing (see page 277). Certificates or certification authorities used for connection encryption

to the mail server must be authentic, as an attacker with the means to trick the user at this point will be able to retrieve in clear text all exchanges with the mail server, including the mailbox login and password. Care must therefore be taken to verify them (see page 323).

What's more, if you use webmail from Tails on a dubious computer, for example In the event of a keylogger attack, you should use a visual keyboard (also known as a "virtual keyboard") when entering your e-mail account password.

page 327

35.7.4 Mail client

If you prefer to use a mail client rather than webmail, you have a choice of :

- Use Tails (see page 113), and follow the Configure and use Thunderbird tool (see page 329). Any traces left locally will be erased when the system is shut down.
- Use Tails and Thunderbird by configuring persistence (see page 116), then follow the Configure and use Thunderbird tool (see page 329). The contents of your mailbox will be stored on a USB key, which will therefore contain encrypted traces.
- Install a mail client on your encrypted system (see page 119). To do this, install the `thunderbird-110n-en` package³ package, following the install software recipe (see page 135), then follow the configure and use Thunderbird tool (see page 329). Traces will then be left on the computer's encrypted hard disk.

But as with webmail, you'll need to check which certificates or certification authorities offer encryption of the connection to the mail server.

page 323

35.8 Exchanging confidential e-mails

The aim here is to hide the content of our emails so that no-one other than the recipient can read them, which can be useful when the content of our messages is *sensitive* or says a lot about the person who wrote them.

To define our security policy, we need to consider the use of ciphers in a number of ways. Let's take the adversary's point of view and see how we can protect ourselves.

35.8.1 First step: ask the postwomen

Without any special protection measures, e-mail hosting services will be able to read the content of e-mails sent to us. This is because it is on their servers that our e-mails are routed and stored. There's no great difference between using a particular protocol, a mail client or webmail.

Our messages can be stored for years until we repatriate or delete them, or even longer if one of the servers makes a copy, as part of a backup for example. Hence the importance of closing mailboxes once they've outlived their usefulness. This also has the advantage of not taking up disk space and consuming resources for nothing at the mail host.

3. The OpenPGP protocol, used for email encryption [this page], has been integrated and activated by default since Thunderbird version 78.2.1. This means we no longer need to install the Enigmail add-on, which was required with previous versions.



TO FIND OUT MORE...

As long as you like tinkering, you can set up and self-host your own mail server on an onion service (see page 266).

Reading our messages is a violation of the confidentiality of correspondence.

- just like reading a letter that isn't addressed to you - requires no technical effort, not even that of opening an envelope. It's so simple, in fact, that it's been automated by Gmail, which has "robots" read the content of its users' e-mails to detect *spam*, but also to "simplify their lives" by, for example, detecting which plane they're going to take and alerting them if it's delayed.⁴



PRECISION

These "robots" are neither automatons nor androids, but small programs that "automatically" scan content to identify something: for example, Google's "robots" scan web pages to index relevant keywords that might be searched for. Such robots are also used by cops to flag up whenever someone uses certain words from their supposed "dictionary of terrorists".

When it comes to the intermediaries between the computers of the protagonists in the e-mail exchange and the servers of the respective e-mail hosts, there are two possible situations. The first, which is now quite rare, is when the connection between the computer and the mail server is not encrypted.

In this case, the various intermediaries will receive the equivalent of postcards. They'll be in a similar situation to mail hosts, except that the postcards will simply be in transit... unless they're set up to inspect the mail they're carrying in greater depth, whether for statistical purposes to improve the quality of their service, or to keep an eye on us.



Unencrypted connection to mail servers

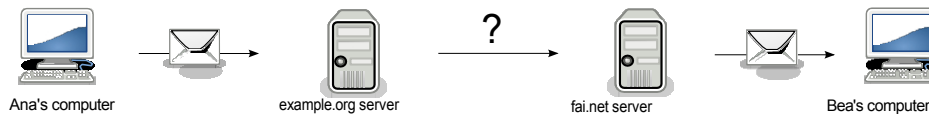
[page 249]

The second situation is where the connection between the computer and the mail server is encrypted using the *TLS* protocol.⁵ This is possible regardless of the protocol used. This time, the intermediaries between the two machines will see postcards stuffed into envelopes. The mail host will not be affected by the encryption and will still have access to the e-mail in its entirety.

4. Janko Roettgers, 2017, *Google Will Keep Reading Your Emails, Just Not for Ads* [<https://va.riety.com/2017/digital/news/google-gmail-ads-emails-1202477321/>] (in English).

5. When we want to encrypt a connection with a web or e-mail server, we use the TLS protocol. This is a standard that encapsulates [page 201] the protocol normally used. For example, the HTTP web protocol, when encapsulated in TLS and therefore encrypted, is called HTTPS. The same applies to the POPS, IMAPS and SMTPS mail protocols.

Finally, there's no guarantee that the connection between Ana's mail server and Bea's is encrypted, in which case the email will travel partly like a letter, partly like a postcard.



Encrypted connection to mail servers

Encrypt your emails

To ensure that the content of our messages cannot be read by any intermediary, including the post office, we can encrypt them directly on our computer, even before we send them. To do this, we'll use the OpenPGP asymmetric cryptographic standard. It would also be possible to use symmetrical cryptography, but its limitations lead us to strongly advise against it.

page 249

With asymmetric cryptography, only the recipient, for whom the encryption has been performed, will be able to decrypt the message. Let's not forget, however, that asymmetric cryptography also has its limitations, which may allow an adversary to reveal the encrypted content.

page 258

In practice, if you haven't already done so, you'll start by importing your recipient's public key. We'll then need to check its authenticity. What's more, if we intend to establish a correspondence and thus receive e-mails in return, we'll also need a key pair: one will be used by our correspondents to encrypt e-mails for us, the other will enable us to decrypt them. If you don't already have a pair of encryption keys, follow the recipe for creating and managing one.

page 337

page 338

Please note, however, that this method encrypts only the content of the e-mail. It will not modify the email headers in any way.

page 333

page 218

Depending on whether you choose to use a mail client or webmail, the method you use to encrypt your e-mails will be different.

Encrypting email in Thunderbird

Follow the recipe for encrypting emails in Thunderbird (see page 333).

Encrypting email for webmail with Tails

If you prefer to encrypt your e-mails using webmail, avoid writing your message in the web browser window and then encrypting it. This is because certain attacks, notably via JavaScript, could access our text from the same web browser. What's more, text written within webmail could be automatically saved unencrypted in drafts. It would be very unfortunate to offer unencrypted text that you wish to encrypt.

page 214

We're not going to explain how to encrypt your e-mails for webmail with encrypted Debian, but only with Tails.

The currently recommended method for encrypting e-mail, as well as for encrypting text, is described in the Tails documentation.

Once Tails has been started (see page 115), afficher the desktop and double-click on the *Tails Documentation* icon. In the index that opens, look for the *Encryption and privacy* section and click on the page *Encrypting text and files with GnuPG and Kleopatra*. At the time of going to press, this page had not yet been translated into French. Follow the section on *Working with encrypted text*.

encrypted text) on this documentation page, and in particular the *To encrypt text* section.

The person receiving the email will have to follow the *To decrypt text* section on the same documentation page.

35.8.2 Second step: look at the computer you're using

Let's suppose that an attacker doesn't have access to our host's data, and can't eavesdrop on the network, but can come and use our data: what traces of our exchanges will he find on our computer?

If this person can get his hands on our computer, or that of our recipient, whether by taking it over or by managing to install malicious software on it, he will be able to access all the emails stored on them as well as the traces left behind; whether these traces are due to the operation of the machine or left by the protagonists.

To protect ourselves from an adversary who might take over our computer, we take care to encrypt our hard disk to make it harder for him to access the data stored on it. This won't protect us against malicious software wanting to exfiltrate this data, hence the importance of installing only trusted software. We can also use an *amnesiac live* system.

Note that if the stored e-mails are part of an exchange that has been encrypted using asymmetric cryptography, even if she has access to the computer and the data stored on it, the adversary will not be able to read them, unless she also has access to the secret key. If we use Thunderbird to send our encrypted e-mails, this secret key is protected by Thunderbird's master password, provided we have set one; in the desktop OpenPGP keychain, the secret key is protected by a passphrase. Without this master password or passphrase, the adversary will not be able to find the secret key, and so will not be able to read the encrypted e-mails.

35.8.3 Third step: attacking media encryption

If you check your e-mails on an encrypted Debian, the traces on your computer's hard disk will be encrypted, whether you're using webmail or a mail client. As such, they will be of no use to an adversary. However, certain adversaries may have ways of attacking this encryption. What's more, if the person with whom you're conversing by e-mail doesn't do the same, the overall level of content protection will be levelled by the weaker of the two protections. Indeed, having taken great precautions and exchanging e-mails with someone who has, for example, an unencrypted Debian, or one that is permanently switched on⁶ may be more dangerous, as it could give a false impression of security. All the more so if it's easy to locate or put a name to the protagonists of the exchange.

If you use a mail program on an *amnesiac live* system, there will be no traces on the computer used after shutdown, but there will be some on the persistent partition if you have configured it. These will be encrypted, which goes back to the previous case of an encrypted Debian.

If you don't want to leave any traces on the computer you're using, encrypted or not, you can use the live Tails system without persistence, taking advantage of its amnesia.

35.8.4 Fourth step: attack message encryption

An adversary with access to encrypted e-mails can try to exploit the limitations of encryption to decrypt the messages.

6. When switched on, a machine with an encrypted hard disk contains a large amount of decrypted information in its RAM [page 18].

Use case: dialog

36.1 Context

In the previous use case, messages were exchanged asynchronously, just like in an epistolary exchange. However, you may want synchronous communication, as in a telephone call, whether to a meeting to work on a sensitive document, or to chat with a friend. The simplest might be to come and meet up, or to call each other - but this isn't always possible or desirable. Sometimes, instant messaging is a good alternative.

Many people are familiar with and regularly use Skype messaging (Microsoft's replacement for MSN or Windows Live Messenger) or Facebook internal messaging, to name but the best-known examples. It's convenient, yes, but it's possible to have something practical without sacrificing discretion!

36.2 Assessing risks

36.2.1 What do we want to protect?

The possible answers to this question are the same as for message exchange. You may want to protect the content of the exchange, the location of the protagonists, their identities, their links, *and so on*.

36.2.2 Who do we want to protect ourselves from?

Here too, the answers are similar to those given in the case of exchanging messages: you may want to conceal all or part of this information from the various machines through which it passes, as well as from people who might have access to it.

First and foremost among these machines are the instant messaging servers used by the various correspondents.

Next come the routers, located on the path between the protagonists of the exchange, notably those of their respective ISPs (Internet Service Providers).

Finally, traces are left on the computers used.

36.3 Defining a security policy

We now turn to the questions set out in our methodology, adopting our opponent's point of view on page 65.

36.3.1 First step: all the information you need for those who are curious

The internal messaging systems of Facebook, Skype, *etc.*, give many people access to information that doesn't concern them: Facebook or Micro-soft will see all our conversations on their machines, and can archive them for later access. The cops only have to ask for the information, and a security flaw on the server can give access to many other people. Not to mention that Facebook regularly changes its privacy settings without warning, and may decide tomorrow to make public what is "private" today.

What's more, Skype records conversation history on the computer used, so anyone with access to the computer could also access this history (friend, burglar, jealous lover...).

But Microsoft and Facebook didn't invent instant messaging, and there are plenty of alternatives available. There are a number of programs you can install on your computer to communicate using a variety of protocols: Skype, IRC, XMPP *and more*.

By using trusted software, we can deactivate conversation archiving, and thus limit the traces left on our computer.

There are also servers that provide instant messaging addresses and are not in a position to do as much cross-checking as Google, Microsoft or Facebook.

To follow this path on a previously installed (encrypted) Debian system (see page 119), refer to the software installer tool (see page 131) to install pidgin. If you're using Tails, this software is already installed¹.

36.3.2 Step 2: Ask the hosts

By using an instant messaging client and various servers, you don't centralize all the links and dialogues in the same hands. However, both the content of conversations and the parties communicating remain accessible from the computers through which they pass.

While it is often possible to set our software to encrypt the connection to the mail server, the dialogues remain accessible to the server. What's more, there's usually no guarantee that the link between the server and the other correspondent will also be encrypted.

[page 228] An adversary with the means to do so could contact the admins of the server being used, or even the organizations providing the network, to obtain information about the conversations. They could also try to "hack" their machines. The confidentiality of dialogues is therefore closely linked to the trust we place in the messaging servers we use, and even in the network infrastructures, particularly our access provider.

[page 235] To make it even more difficult for an adversary to read the content of our dialogues, we can use end-to-end encryption to ensure **confidentiality**.

To follow this method on a previously installed (encrypted) Debian system (see page 119), follow the software installation tools (see page 131) to install the pidgin-otr package, then use instant messaging with OTR (see page 351).

1. Tails is discussing replacing Pidgin with another instant messaging program. This proposed change is tracked on the Tails gitlab [<https://gitlab.tails.boum.org/tails/tails/-/issues/8573>].



TO FIND OUT MORE...

Technical solutions are currently being developed and integrated into Debian to enable end-to-end encryption in group conversations. One example is Dino², which has announced the integration of the OMEMO chiffrement protocol.³

36.3.3 Step three: Keep links visible

If we use end-to-end encryption in an instant messaging dialogue, an adversary can no longer access the content of the conversation, unless he breaks the encryption, accesses our computer, or even hacks into it.

page 258

However, an adversary with access to the network or mail server used can always see who we're talking to. To hide links, we need to use contextual identities and connect anonymously, for example using Tor. This not only ensures confidentiality thanks to encryption, but also pseudonymity.

page 238

page 243

By using an amnesiac live system like Tails, you also take care of any traces that might be left on the computer you're using. Unless you use persistence, in which case encrypted traces will be kept in the persistent partition of the Tails USB key.

page 261

If you don't already have one, you'll need to start by making a Tails USB key or DVD (see page 113).

Next, after booting into the medium containing Tails (see page 107), we'll need to define a contextual identity to use and set up Tails persistence (see page 116) for this identity by activating the "Pidgin" option.

Finally, we'll be able to follow the tool's use of instant messaging with OTR (see page 351).

Two criteria are combined here: confidentiality and anonymity. In the previous step, we saw how to achieve confidentiality using OTR encryption. Here we've just seen how to have anonymity and confidentiality using OTR encryption under Tails, as well as a contextual identity. However, you may want anonymity or pseudonymity alone, i.e. without confidentiality. Indeed, we may want to hide who we are without hiding the content of our conversations, for example when chatting in

"To follow this trail, start up Tails and then use Pidgin. To follow this trail, start Tails (see page 115), then use Pidgin (see page 351) without using OTR encryption, with an account created for the occasion.

36.4 The limits

First of all, this method remains vulnerable to the encryption attacks we've just been talking about, and to attacks on Tor.

page 279

But there are also some limitations specific to real-time conversations. For example, the "online" or "offline" status of an identity is usually publicly accessible. An adversary can thus see when an identity is online, and possibly correlate several identities: because they are always online at the same time; or, on the contrary, because they are never online at the same time, but often successively, and so on.

2. Dino [<https://prism-break.org/fr/projects/dino/>].

3. OMEMO protocol [<https://prism-break.org/fr/protocols/omemo/>].



PRECISION

To make identities appear to be "always on", it is possible to use a "ghost" or proxy⁴ on a trusted computer that is always on and connected to the instant messaging server. In this way, it is this computer, and not the server, that "sees" when you are connected or not, and this state is no longer public. Setting up such an infrastructure, however, is beyond the scope of this guide.

[page 244]

Then, in the particular case where anonymity (or pseudonymity) takes precedence over other constraints, for example if you want to chat in a public room, other limits are added to those mentioned above. A contextual identity always runs the risk of ending up linked to a civil identity, as we saw in the section on pseudonyms. Indeed, even under a pseudonym, the content and form of our conversations can reveal a great deal about the person behind the keyboard.

[page 289]

It's worth bearing in mind that when you're trying to define a security policy for a relationship between several people, whether on the phone, exchanging e-mails or using instant messaging, the overall level of security will be levelled by the level of security of the least cautious protagonist. If, for example, we take care to use Tails so as not to leave any trace of our conversation on the computer, while our interlocutor uses her usual operating system without any particular protection, then the latter will undoubtedly be the weakest point in our communication security policy.

Finally, as already mentioned, OTR encryption does not currently allow more than two people to converse at the same time. However, research is moving in this direction⁵.



TO FIND OUT MORE...

In the meantime, and provided you like tinkering, it's already possible to set up your own instant messaging server (e.g. XMPP) on an onion service (see page 266).

4. Wikipedia, 2014, *Proxy* [<https://fr.wikipedia.org/wiki/Proxy>].

5. Ian Goldberg *et al*, 2009 *Multi-party Off-the-Record Messaging*, CACR Tech Report 2009-27 [<http://www.cacr.math.uwaterloo.ca/techreports/2009/cacr2009-27.pdf>]; Jacob Appelbaum *et al*, 2013, *mpOTR* [<https://libraries.io/github/ioerror/mpOTR>].

Use case: sharing documents sensitive

37.1 Context

We've seen how to publish documents you want to make public. But it is also sometimes necessary to share information with a restricted group of people. sensitive documents such as confidential work documents, photos of vacations, or the contact details of a source willing to divulge internal company documents

In this case, we'll be concentrating on sharing sensitive documents *via the* Internet, which is the subject of this second volume of the *guide*. Depending on our situation, it may also be possible to exchange encrypted USB sticks, paper documents *and so on*.

37.2 Assessing risks

37.2.1 What do we want to protect?

Document content

The content of shared files is confidential. Only the recipients should be able to access them, in the same way as when sending an e-mail message. For example, if you want to share vacation photos with your family, you need to hide the photos themselves. The fact that the recipients are family members is not, *a priori*, sensitive information. It's all about *protecting what you share*.

Source and destination

The identity of the source and recipient can also be part of the information to be protected. In the case of leaked company documents, who sent the documents and to whom are two particularly sensitive pieces of information (the protection of journalists' sources of information is, in fact, the basis of journalistic ethics). It's a question of *protecting who shares with whom*.

So, we can divide this issue into three parts: the first deals with source protection, the second with recipient protection, and the third deals specifically with the confidentiality of documents to be shared.

37.2.2 Who do we want to protect ourselves against?

The aim is to protect you from prying eyes looking to see *who's* doing *what* on the web, as in the case of *web site browsing*. But also from prying eyes that might *stumble* across these files.

37.3 Protect the source

Like first aid: *protect yourself so you can treat others.*

As our files are confidential, their contents are not normally supposed to be made public. That said, there's no guarantee that they won't end up in the public domain, whether by our own error, by people who also have access to them, or by adversaries who might jeopardize our strategy or its implementation.

[page 285]

The process is very similar to that of publishing a document that can be read or reread. However, a few considerations specific to this situation are necessary.

37.3.1 First step: traces in the document

When we want to share confidential documents, especially if we've produced them ourselves, there's nothing to indicate *a priori* that we can trust the people with whom these documents will be shared.

Let's imagine, for example, that we want to give documents attesting to the extravagant comp- ability of our political party to a journalist so that she can write an article about it without publishing them. *A priori*, we have no confidence in this journalist and would therefore prefer her not to know from whom these documents originate.

It's important to avoid leaving any traces that could lead back to us. Whether they're obvious, like a civil identity, or more discreet, like metadata:

[page

30

- all this document production work must be carried out in a suitable environment (see page 79);
- take care to delete the metadata (see page 185).

37.3.2 Step two: protect yourself from intermediaries

Using our previous example, if the people with whom we share files are untrustworthy, they could, willingly or unwillingly, reveal the site on which they found them.

[page 225]

[page 228]

If the adversary has the power to access the connection logs, through pres- sions or requisitions, he could find out the source of the connection that enabled these files to be put online. As a result, and if we haven't put in place a number of safeguards on our computer, our adversary could trace us back to the public IP address we used, or even to our computer's MAC address.

[page 261]

[page 315]

[page 266]

[page 359]

To avoid indiscretion by the various intermediaries between our computer and the server where our files will be hosted, we will use the Tor network, via the Tor Browser.

We can go a step further and avoid using a third-party server: by sharing our files directly from our computer with an Onion service (using the OnionShare tool).

In this case, even if the web address used to retrieve the documents is revealed, it doesn't help to know where the computer is located, and therefore can't be traced back to us.

[page 279]

However, it's important to bear in mind the possibility of the opponent attacking Tor.

1. Connection logs can be found in the box, at the Internet Service Provider and at hosting companies.

37.3.3 Third step: look at the source computer

Confidential documents or traces of them may remain on your computer, whether intentionally or not.

The solutions are either to have your hard disk encrypted, or to avoid leaving any traces from the outset by using an amnesiac *live* system.

page 119

page 113

37.4 Protecting recipients

Having taken the necessary precautions to protect ourselves, we also need to think about the recipients of our files. Even if we can't always know the complete list of people who will have access to these documents, or protect them for them, we can always make sure that a minimum of protection is necessary for them to access them.

The simplest, efficient and achievable way is to use an onion service, which will force recipients to also use the Tor network. To do this, you'll need to follow the OnionShare tool.

page 266

page 359

37.5 Protect confidential files

Once you've thought about protecting the people who share your documents, it's time to think about protecting the files themselves.

The procedure here is similar to that for exchanging confidential e-mails. But we won't be using e-mail, either because our files are too voluminous, or because we don't have a precise list of recipients, and therefore no list of e-mail addresses to send these files to. We prefer to share our files online on a server, as in the case of a private publication.

page 295

page 285

The solutions we use all talk about encryption in different ways, on page 47 depending on our security policy and our approach to sharing.

37.5.1 Choose from available tools

There are several tools available for encrypting our files before sharing them. The choice between them depends on the level of sharing and the quality of encryption required.

37.5.2 Host-provided encryption

First of all, the solution that seems to require the least effort is to put our documents on a file hosting service that offers to encrypt them directly on the server hosting them.

page 319

Typically, these services encrypt the files in the user's browser before sending them to the server. The site then creates a download link with the decryption key included in the link.² One of the advantages is that this key is not held by the service host, who therefore has no access to the user's files and, even in the event of pressure or requests from the cops, is unable to provide them in clear text. The main disadvantage of this method is that the decryption key is contained in the download link. In other words, whoever has access to this link also has access to the files.

page 228

2. Various software packages can be used by the servers hosting these services. Lufi and Up1 are two examples. We can trust them *a priori*, and the people writing these lines know of no other software of this type.

Using these services to share confidential files therefore depends on the level of trust you can place in the software providing the service and in the host who has configured it, as well as on the confidentiality of the download link.

To limit the risks, however, you can check the option that activates file deletion immediately after the first download. This ensures that the files are downloaded only once, and allows you to find out whether the files have already been downloaded, and whether the communication method used to transmit the link was not confidential.

If, however, you wish to encrypt using the file hosting service, you will need to :

- use the Tor Browser (see page 315) to access the web ;
- consult the share a file section (see page 321) of the *Find a web hosting tool* ;
- have a way of transmitting the download link confidentially, for example by sending it in an encrypted e-mail (see page 333).

37.5.3 Encryption before sharing

Another option is to encrypt files before putting them online. This solution is a little more complex to implement, but it has the advantage of not requiring you to trust the host. You choose how your files are encrypted, and even who can decrypt them.

Once again, several options are available: depending on the number of recipients, we can encrypt our files with a passphrase or with one or more public keys.

In both cases, pay close attention to the name of the file containing the encrypted document(s): if this name is explicit, it may reveal information about the content of the documents. Rename files with a neutral name, such as "document" or "archive".

Encrypting with a passphrase

Encrypting our files to be shared with a passphrase means that whoever has it can decrypt and access our documents. However, we will need to know their location, i.e. the web address from which they can be downloaded, or have access to one of the computers on which they are stored.

An important detail is that everyone with access to the files must know the passphrase used to encrypt them in order to make them readable. A confidential means of communication must therefore be used to share this secret between all recipients, which can sometimes prove complicated.

Finally, we'll come up against the same limitations as those discussed in the chapter on symmetrical cryptography.

Encrypt with one or more public keys

If we have a defined list of people with whom to share our documents, and each of them has an OpenPGP key pair, we can encrypt these files with their keys, so that only they can decrypt them in the end.

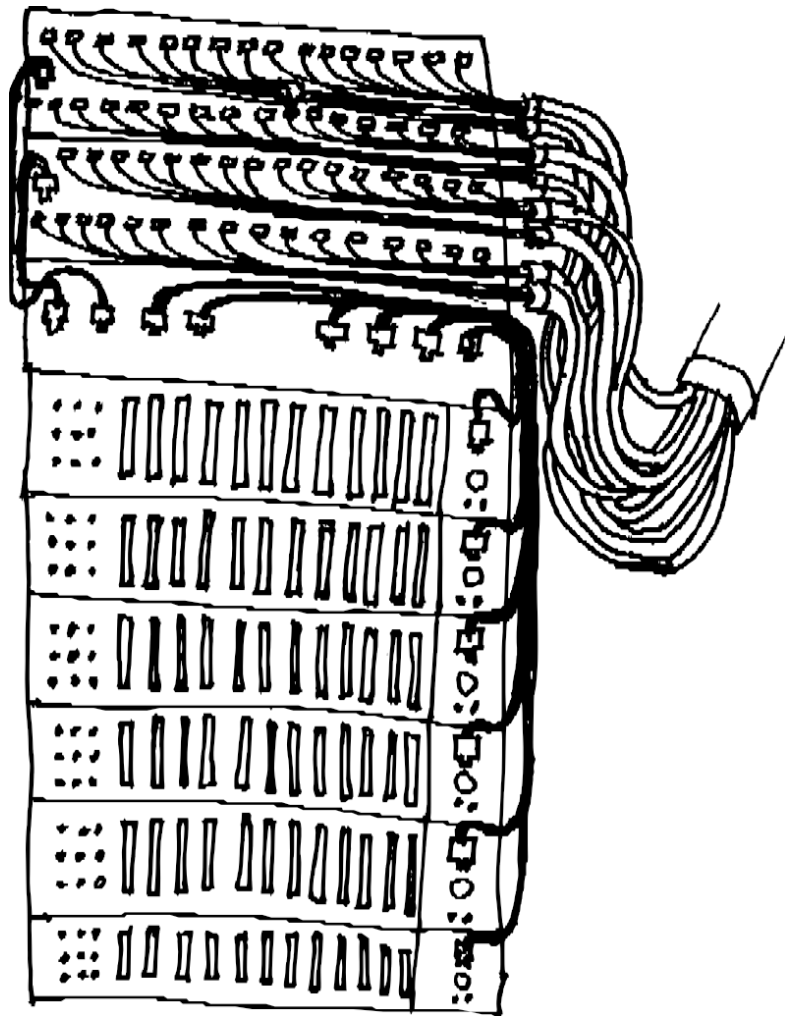
Let's go

You'll first need to follow the encrypt data tool (see page 347), then choose one of the two solutions mentioned above to host these files:

- use a web hosting service (see page 319)
- or host them yourself with OnionShare (see page 359).

Decrypting files

The recipients of the documents will have to decipher them following the appropriate recipe (see page 348).



PART SIX

Tools

Introduction

In this third section, we'll explain how to apply some of the above ideas in practice.

This section is a technical appendix to the previous ones. Once the issues surrounding privacy in the digital world are understood, and the appropriate responses chosen, the question of "How do we do it?" remains, to which this appendix provides some answers.

page 275

For most of the recipes presented in this guide, we assume that you're using GNU/Linux with the GNOME desktop; these recipes have been written and tested under Debian GNU/Linux version 11 (nicknamed Bullseye) ¹ and Tails version 5 ² (*The Amnesic Incognito Live System*).

However, these are generally adaptable to other Debian-based distributions, such as Ubuntu ³ or LinuxMint ⁴.

If you're not yet using GNU/Linux, take a look at the use cases in the first section.

to me, chapter a fresh start, or use a live system.

page 71

Procedures are presented step-by-step, and wherever possible, the meaning of the proposed actions is explained.

page 113


The order in which each recipe is detailed is important. Unless otherwise stated, it is recommended not to skip a step and then go back. The result could be very different from the one expected.

Finally, it's important to use the most up-to-date version of this guide, as software evolves. It can be found on the <https://guide.boum.org/> website.

-
1. <https://www.debian.org/releases/bullseye/index.fr.html>
 2. <https://tails.boum.org/index.fr.html>
 3. <https://www.ubuntu-fr.org/>
 4. <https://linuxmint.com/>

Installing and configuring the Browser Tor

 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

 *Duration: 15 minutes.*

As we've seen, when we surf the web, the sites we visit can register our IP address, making it easy for adversaries to trace us. This is why we sometimes need to hide our IP address. Tor is a software program that allows you to route your connection through a network of

page 202

"This hides our real IP address. This is called onion routing.

page 261

To use the *Tor anonymizing network*, you need to configure not only the Tor software itself, but also the software that will use it, such as your web browser. These settings are often complex, so much so that it's difficult to be sure of the resulting anonymity.

That's why it's best to use Tor either on a dedicated *live* system, or with a "ready-to-use kit": the Tor Browser. This is a tool that makes it very easy to install and use Tor on a "classic" system. No configuration is necessary, and all the software required for Tor *browsing* is included.

page 113

The Tor Browser brings together :

- Firefox web browser, set to use Tor ;
- Tor software ;
- a launcher, to start everything with a simple double-click.



Please note that the Tor Browser does not provide anonymity for the entire computer: only connections to websites initiated in this browser pass through Tor. **All other connections (client**

mail, RSS aggregators, Torrent, other web browsers, etc.) are not anonymized. What's more, even when the Tor Browser tries to minimize the traces left behind, browsing data such as cookies or history may still be saved on the hard disk, as may downloaded files or browser bookmarks. In the course of our browsing, we may also click on a link that opens another program (such as a music player), which does not pass through Tor. These warnings are not We can't take this lightly, as clues as to the nature of our navigation could be leaked.

Here's how to install the Tor Browser on an encrypted Debian. However, to be able to use a system that connects to the Internet only *via*



page 119

Tor and being able to use Tor with software other than a web browser, the easiest way is to turn to a *live* system like *Tails*.

page 113



38.1 Install Tor Browser

To install the Tor Browser in Debian :

- Add contrib deposit (see page 136).
- Install the *Tor Browser Launcher* software (see page 134) by searching for *torbrowser* in the software list.
- Launch *Tor Browser Launcher* by pressing  ( on a Mac), type `torb` and click on *Tor Browser Launcher*.


A *Tor Browser Launcher* configuration window opens. You can leave the default options and click *Install Tor Browser*.

The first time it's run, the *Tor Browser Launcher* downloads the Tor Browser from the **official Tor website** [<https://www.torproject.org/fr/>] and automatically checks the signature of the archive, eventually extracting and running it.

At the time of writing, the name of the browser has not been translated and is called *Tor Browser* in English. After installation, a shortcut appears on the computer, to find it afficher the Activities overview by pressing the  key ( on a Mac), then type `tor`.


If Tor is blocked (by our ISP, for example), or if using Tor might look suspicious to someone monitoring our Internet connection, we can configure the Tor Browser to use Tor bridges to hide our Tor u s a g e .

[page 267]


The Tor Browser documentation can be consulted by clicking on 


→ *Help* → *Tor Browser User's Guide* then going to the *Bridges* page.

38.2 Tor Browser update

The Tor Browser automatically downloads the required updates, then offers to apply them. To do this, click on the  menu, then on *Restart to update the Tor Browser*.

Browsing the web with Tor

 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

 *Duration: Five to ten minutes.*

The aim of this tool is to browse the web confidentially using the Tor Browser. There's not much difference from using a "classic" web browser, which we'll consider a prerequisite. [page 261]

If you're not using the live Tails system (see page 113), you'll need to install the Tor Browser (see page 313).

Once you've launched Tor Browser, you can use it almost as you would an ordinary web browser. However, there are a few details to note.

First of all, you need to understand what Tor protects against, but above all what it doesn't protect against, so that you don't do just anything in the belief that you're protected. [page 267]

 **Please note:** unless you're using Tails, only browsing with the Tor Browser benefits from the confidentiality provided by Tor.

In addition to these limitations, you should be aware that the websites you visit may know that you're connecting *via* the Tor network. Some, like Wikipedia, use this to block anonymous editing. Others, like Google, will ask you to solve challenges called "captcha" to show that you are indeed a person. ¹ to prove you're a person (and not a robot) before accessing their services. Solving these challenges means producing unpaid work, usually for the GAFAMs. ²...

Certain functions are deactivated to avoid leaving traces, such as storing cookies on the disk or saving passwords.

39.1 Go to the Tor Browser download folder



All files downloaded from the Tor Browser are saved in a specific folder, which is well hidden. ³ The easiest way to find this download folder is to download a document from the Tor Browser and *open the folder containing the file*.

For example, on a web page with images, you can right-click to *save the image as* Once the download is complete, a new symbol 




1. Wikipedia, 2017, *CAPTCHA* [<https://fr.wikipedia.org/wiki/CAPTCHA>].

2. Xavier de La Porte, 2016, *Le " captcha " ou l'art de faire travailler sans rémunérer*, L'Obs [<https://www.nouvelobs.com/rue89/rue89-ce-qui-nous-arrive-sur-la-toile/20140217.RUE2129/le-captcha-ou-l-art-de-faire-travailler-sans-remunerer.html>].

3. If the Tor Browser is installed from the Tor Browser Launcher and the language of the operating system is French, the download folder can be found in the personal folder: `.local/share/torbrowser/tbb/x86_64/tor-browser_fr/Browser/Téléchargements`.

(or ) appears next to the address bar. This arrow affiches the download list and the  symbol prompts us to *Open the folder containing the file*. A new window opens, this is Tor's download folder, and the path to get to this folder is affiche in the menu bar. Now we can move the downloaded files wherever we like.

If you want easier access to this folder later, you can also create a shortcut to the Tor *Downloads* folder:

- In the Tor Browser, after downloading, to the right of the address bar, click on  (or )
- In the drop-down menu, click on the  icon to open the folder containing the downloaded file.
- At the top of the window that opens, in the address bar, click on Downloads▼.
- From the drop-down menu, select *Add to bookmarks*.
- The bookmark appears in the left-hand column.
- Right-click on it and select *Rename...*
- Give it a clear name, like *Tor Browser Downloads*.

39.2 Geolocation limitations

When using Tor, for the website we're visiting, our connection appears to come from the location of the exit node being used. Some sites use the IP address of their visitors to choose the language of affichage. These sites may therefore affich in unexpected languages.

What's more, some government agencies locate their users based on their IP addresses, so using Tor can pose problems when dealing with government agencies.⁴



TO FIND OUT MORE...

It seems that the French authorities that monitor their users' IP addresses only monitor the country of origin of the connection, and not (yet?) the addresses of Tor exit nodes.

It's possible to temporarily configure the Tor Browser to use only exit nodes in France, which will always give a French IP when browsing.

Warning: using this option reduces confidentiality.

To do this when using a Tor Browser installed with torbrowser-launcher :

1. Close Tor Browser.
2. From the home directory, find the Tor configuration file called *torrc* :
 - Go to *Personal folder*.
 - Press Ctrl + to afficher hidden files.
 - Click on the magnifying glass symbol in the menu bar and type *torrc* in the search bar.⁵
3. Open this *torrc* file with a text editor (right-click → *Open with Text Editor*),
4. Add a line containing `ExitNodes {FR}`, then save and close the file.
5. Start the Tor Browser, browse a few sites and click each time on the padlock in the address bar to check that the last node is still in France.

4. Anonymous, 2019, *Account of a CAF inspection* [<https://nantes.indymedia.org/posts/45908/>].

As soon as you've finished your administrative activities, don't forget to shut down the Tor Browser immediately, and re-edit the *torrc* file to remove the line you've added.

To do this in Tails :

1. On startup, set an *admin password*⁶ then *Start Tails*.
2. Connect to Tor, then open a terminal (*Applications* → *System Tools* → *Terminal*).
3. Type `sudo gedit /etc/tor/torrc` in the terminal, press *Enter* (or return), then enter the administration password you configured at startup. The Tor configuration file opens.
4. Add the `ExitNodes {FR}` line to this file, then save and exit the editor.
5. In the terminal, type `sudo service tor reload` to restart Tor with the new configuration. Re-enter the admin password set at startup.
6. Restart the Tor Browser, navigate to a few sites and click each time on the padlock in the address bar to check that the last node is still in France.

Once administrative activities have been completed, restart Tails.

5. If the Tor Browser is installed from Tor Browser Launcher and the operating system language is French, the path should look like this:
`.local/share/torbrowser/tbb/x86_64/tor-browser_en/Browser/TorBrowser/Data/Tor/torrc.`

6. https://tails.boum.org/doc/first_steps/welcome_screen/administration_password/index.fr.html

Choosing web hosting

🔄 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

🕒 *Duration: Half an hour to an hour.*

The aim of this section is to find out where to host a document on the web. There are too many possibilities to provide a "turnkey" answer to this question. What's more, it doesn't seem like a good idea to recommend a short list of hosting providers, where many "at-risk" contents would be centralized. Instead, this recipe will give you a few pointers to help you make the best choice of hosting provider.

It's also possible to host our own document anonymously using Tor's onion services. To do this, you'll need to go to the recipe on using from OnionShare.

page 242
page 266
page 359

40.1 A few selection criteria

There are so many possible hosts that you can quickly feel lost in the jungle of possibilities. Here are a few criteria to help you ask the right questions. We'll talk about documents below, but these criteria also apply to a more ambitious project, such as a blog or a video documentary.

- **Type of organization:** many sites offer to host documents "for free". Many of these are commercial services that find it profitable to publish content created by their users. But there are also associations or collectives that host projects, under certain conditions.
- **Hosting conditions:** if the host doesn't like the document, there's nothing to stop them deleting it without even warning us. The host's charter (which we must accept when we host our document) can often give us an idea of what the host will or won't tolerate.
- **Identification requirements:** the extent to which the host requires us to provide details and guarantees about our personal data in order to use its services.
- **Resisting pressure:** the state may also want to prevent our document from remaining online. In many cases, it will suffice to intimidate the host into deleting our document. In fact, depending on the host chosen, it may be able to withstand more or less pressure: some will wait until a legal action has been taken, while others will delete our document as soon as the first slightly threatening email is sent.
- **Document deletion:** conversely, you may wish to delete your document at some point. However, as document hosting is a service that is entrusted to other people, who may or may not be trusted, we don't

page 240

can't guarantee that our files will actually be deleted at our request. In some cases, getting to know the host better can give us more guarantees.

- **Risks for the host:** depending on the content of our document, it may put the host at risk, especially if the host doesn't want to cooperate with the cops. In such cases, you need to ask yourself whether you're willing to put a host at risk, since they could disappear if the cops come.
- **Document size:** if our document is "too big", some hosts will refuse to accept it. This may also be the case if our document is "too small". The size allowed is specified in some offers, but beware: some hosts charge for features such as hosting very large files.
- **Hosting duration:** depending on the hosting provider, there are many different offers regarding hosting duration. For example, some automatically delete the document after a given time, others if it hasn't been downloaded for a certain period of time, *and so on*.
- **Identification conditions for consultation:** in order to minimize the possibility of the host or the cops being able to identify the people who come to consult our document, it is important not to use a host on which they could already be identified. For example, social networks and similar platforms (Facebook, Twitter, YouTube, *etc.*) should be avoided.
- **Use via Tor:** for the same reasons, it's best to ensure that the document can be deposited and/or accessed via the Tor Browser, or even via an onion service.
- **Retention of connection logs:** both sending and viewing the document may leave compromising traces in the host's connection logs. Choosing a host that does not keep these logs, or regularly deletes them, reduces this risk.
- **Document confidentiality:** depending on our needs, we may want the hosting provider to offer an encryption system so that the document content cannot be read on the server, or on the contrary, we may not care, since the document will be publicly accessible.

[page 261]

[page 266]

[page 225]

40.2 Content type

Now that we've got a few selection criteria in mind, let's try to make it more concrete. The right hosting for our project depends on the type of content we want to publish: text, images, video, sound, *etc.*

40.2.1 Publish text

Publishing text is often the easiest thing to do.

If the text to be published is related to another text already published, it is often possible to post a comment, whether on a blog, a forum or a participative site. For this type of publication, registration is not necessarily required. This doesn't mean, however, that the publication is anonymous unless special precautions are taken, such as using onion routing. What's more, as our text is a comment and not a main topic, it is not necessarily highlighted on the site.

[page 261]

It's also possible to publish a text on an existing site or blog. In this case, you'll need to send it to the site in question *via* a form or e-mail, and publication will then depend on the administrators. Some sites ¹ offer free publication of articles on a given theme.

1. For example, the sites of the **Indymedia** network [<https://fr.wikipedia.org/wiki/Indymedia>] and those of the **Mutu** network [<https://reseaumutu.info>].

40.2.2 Have a blog or other site

If you want to publish texts on a regular basis, you can also choose to administer a blog: many organizations offer blogs that are already configured and easy to use. You could also manage a website, but this requires a bit of training.

In many cities, groups of people interested in free software or freedom of expression on the Internet can be a good source of advice. Some lists are also available on the web:

- a list of large blog platforms on Wikipedia [https://fr.wikipedia.org/wiki/Cat%C3%A9gorie:H%C3%A9bergeur_de_blogs];
- a list of free web services on the wiki of the French-speaking Ubuntu community [https://doc.ubuntu-fr.org/liste_de_services_web_libres];
- there's also the noblogs.org host [<https://noblogs.org/>].

40.2.3 Publish audiovisual files

There are several solutions for publishing images, videos or sounds to accompany the text of an article, for example. Firstly, most of the sites where you can publish text offer the option of including audiovisual documents. These sites offer either to take files from our computer (which will then be hosted on their server), or to indicate the address of files already hosted on another server.

There are also sites dedicated to sharing audiovisual files. Here are a few examples:

- The non-profit *Internet Archive* aims to be a [free digital library](https://archive.org/) [<https://archive.org/>].
- The CHATONS collective ² maintains a [list of numerous free tools and services](https://entraide.chatons.org/) [<https://entraide.chatons.org/>], including [video sharing services](https://www.chatons.org/search/by-service?service_type_target_id=152) [https://www.chatons.org/search/by-service?service_type_target_id=152] and [photo album hosting services](https://www.chatons.org/search/by-service?service_type_target_id=150) [https://www.chatons.org/search/by-service?service_type_target_id=150]. Some services also allow files to be stored encrypted on their servers (in the case of image hosting, however, depending on the server, this is not always automatic: you may have to explicitly request encryption of the file when sending it).
- Finally, you can use the tools mentioned in the next section on file sharing.

40.2.4 Share a downloadable file

To publish documents that you want to make downloadable, look no further than Direct Download *Link* (DDL) services.

In French, this means "direct download link": we "upload" our file to a direct download server, and then obtain a link (a web address) which, when typed into a web browser, launches the file download.

There are also file-sharing and file-hosting sites. Here are a few examples:

- The Riseup project, a collective providing secure communication tools, also offers a [lightweight file-sharing](https://share.riseup.net/) tool [<https://share.riseup.net/>].

2. [CHATONS](https://chatons.org/) [<https://chatons.org/>], for Collectif d'Hébergeurs Alternatifs, Transparents, Ouverts, Neutres et Solidaires, is an initiative that came into being in 2016. The aim of this collective is to bring together organizations wishing to offer services that respect the privacy of who use them.

- Some CHATONS ³ provide a file-sharing service [https://www.chatons.org/search/by-service?service_type_target_id=148].

Some services, such as those based on *Lufi* software, allow files to be stored encrypted on their servers.

40.3 In practice

More concretely, the first thing to do is to choose a file host. The criteria outlined above will help you make this choice. It's very important to make a well-informed choice of host, since our anonymity may depend in part on this choice.

[page 305] It is also possible to encrypt the file to be hosted. There are two ways of doing this:
[page 347] either we encrypt the file before hosting it online; or we choose a hosting company that encrypts the file in our web browser before storing it on its servers, like CHATONS for example.

To host our file, the exact method differs from host to host, but the principle remains the same. First, we open our web browser and use it discreetly. Then [page 277] we'll go to the host's site and find the page where we can "upload" our file. There, you'll need to follow the host's specific method for transmitting your file. In general, this method is easy to follow and, although it varies, remains relatively similar from one host to another. Once the *upload* is complete, the web address where the file can be found is affichée.

It is sometimes necessary to enter an e-mail address in order to receive this web address: the use case on e-mail exchanges and the chapter on contextual [page 293] identities will enable us to decide which e-mail address to provide in this case.

[page 243] Once you've obtained the link, you can distribute it as you see fit. People who have the link can download the file by typing it into the address bar of a web browser.

3. CHATONS [<https://chatons.org/>], for Collectif d'Hébergeurs Alternatifs, Transparents, Ouverts, Neutres et Solidaires, is an initiative that came into being in 2016. The aim of this collective is to bring together organizations wishing to offer services that respect the privacy of who use them.

Verify an electronic certificate

🔄 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

🕒 *Duration: Fifteen to thirty minutes.*

We have already seen that, in order to establish an encrypted connection, it is often necessary to trust a certification authority (CA). Most of the time, CAs are already registered on the computer, in the web browser for example. But this isn't always the case: our web browser or other software will then present us with a message explaining that it was unable to authenticate the service's certificate. page 255

It can also happen that the visited service, due to a lack of trust, does not use a certification authority. In this case, we have to check the certificate ourselves.

41.1 Verify a certificate or a certification authority

To view a website's certificate, in a web browser, click on the padlock🔒 in the address bar, then on *Secure Connection*, then on *More information*. A new window opens, displaying a wealth of information about the web page.

By clicking on the *Afficher le certificat* button, you can take a closer look at the certificate, and find out, for example, who issued it, for how long, *and so on*. In this window, there are usually several tabs, each corresponding to a certificate. The first tab corresponds to the site certificate itself; subsequent tabs correspond to the certification authorities that authenticate the site certificate (by means of a digital signature). page 252

We're particularly interested in the certificate presented by the site to our web browser. Its SHA-256 fingerprint can be found in the *Digital fingerprints* section of the first tab.

For the <https://guide.boum.org/> certificate used on December 13, 2021 ¹for example, we will obtain the following character string :

```
72:7 E:9E: A3 :1E:2E: B9: E1 :5B: D5 :88:93:01:38:7 A:70:
8B: C6 :81: E2: F3: D0 :5F: CC:63:40:51: CF:22: EC:28:41
```

1. This certificate is available at <https://crt.sh/?id=5796332967>.

Sometimes the browser affiche a security warning.



Attention : risque probable de sécurité

Le Navigateur Tor a détecté une menace de sécurité potentielle et n'a pas poursuivi vers `untrusted-root.badssl.com`. Si vous accédez à ce site, des attaquants pourraient dérober des informations comme vos mots de passe, courriels, ou données de carte bancaire.

Que pouvez-vous faire ?

Le problème vient probablement du site web, donc vous ne pouvez pas y remédier.

Si vous naviguez sur un réseau d'entreprise ou si vous utilisez un antivirus, vous pouvez contacter les équipes d'assistance pour obtenir de l'aide. Vous pouvez également signaler le problème aux personnes qui administrent le site web.

[En savoir plus...](#)

Retour (recommandé)

Avancé...

page 254

The notion of "stolen information" mentioned in the previous message refers to the attack by the monster in the middle. Once you've read this warning, you can click on *Advanced...*, which will reveal the reason why the web browser didn't want to accept the certificate, as in the following screenshot.



Attention : risque probable de sécurité

Le Navigateur Tor a détecté une menace de sécurité potentielle et n'a pas poursuivi vers `untrusted-root.badssl.com`. Si vous accédez à ce site, des attaquants pourraient dérober des informations comme vos mots de passe, courriels, ou données de carte bancaire.

Que pouvez-vous faire ?

Le problème vient probablement du site web, donc vous ne pouvez pas y remédier.

Si vous naviguez sur un réseau d'entreprise ou si vous utilisez un antivirus, vous pouvez contacter les équipes d'assistance pour obtenir de l'aide. Vous pouvez également signaler le problème aux personnes qui administrent le site web.

[En savoir plus...](#)

Retour (recommandé)

Avancé...

Quelqu'un pourrait être en train d'essayer d'usurper l'identité du site. Vous ne devriez pas poursuivre.

Les sites web justifient leur identité par des certificats. Le Navigateur Tor ne fait pas confiance à `untrusted-root.badssl.com`, car l'émetteur de son certificat est inconnu, le certificat est auto-signé ou le serveur n'envoie pas les certificats intermédiaires corrects.

Code d'erreur : `SEC_ERROR_UNKNOWN_ISSUER`

[Afficher le certificat](#)

Retour (recommandé)

Accepter le risque et poursuivre

In the case of a self-signed certificate, for example, you may read *The certificate is not secure because it is self-signed*. It is also possible that the validity date of the certificate has passed, which does not necessarily prevent its use. In any case, it's always a good idea to read this section and ask yourself whether you want to continue in the light of this information. It is then necessary to check the site's certificates and those of any certification authorities. Otherwise, the connection will be encrypted, but not *authenticated*. In other words, the communication will be encrypted, but you won't really know who you're communicating with - which is far from ideal.

page 254

page

53

Verifying a certificate usually means viewing its digital fingerprint and comparing it with another source to make sure it's correct. We

we prefer to use the SHA-256 digital fingerprint, rather than the MD5² or SHA-1³ which are no longer considered secure.

It remains to find other sources for obtaining this fingerprint. There are a number of techniques you can use to verify the authenticity of a certificate:

- If a trusted person in our vicinity already uses the site or CA in question and has already verified its certificate, we can compare the certificate fingerprint they know with the one presented to us. We can also request it by email from people who will send it to us encrypted *and* signed for added security. It's even better if you're in contact with several of these people, who would have verified the certificate using different Internet connections. In this case, you need to follow the procedure explained below to find the fingerprint of a certificate already installed in the web browsers of these people.


[page 297]

- If we have access to several Internet connections from our location, For example, in an urban area where there is a lot of Wi-Fi access, you can visit the website or download the CA certificate using several of these connections, and compare the certificate fingerprint presented to you each time.

- If you're using the Tor Browser, you can take advantage of the change of circuit, and therefore of exit node on the Internet, to check the certificate's fingerprint several times. This will prevent a malicious person with their hands on the exit node, or who is positioned between the exit node and the site consulted, from usurping its identity.

[page 261]

[page 254]

To find out the IP address of the exit node used to access a site in the Tor Browser, click on the padlock  on the left of the address bar, just before the site address. A *Site Information [...]* insert then appears, detailing, among other things, the *Tor Circuit* used for this site. The exit node is the second-to-last node in the list, just before the node corresponding to the site visited. Its geolocation (country) and IP address are indicated. (Please note: there is no exit node when consulting an onion service, i.e. a site whose domain name is *.onion*).

In the same insert, it is possible to change the Tor circuit used to access this site by clicking on the *New circuit for this site* button, located just below the representation of the current circuit. You can then ensure that the IP of the exit node changes each time the circuit is renewed.

Each time the exit node changes, we can reload the visited site or CA certificate, and compare its fingerprint with those collected the previous times. After a few successful attempts, the probability that it's the right certificate becomes sufficient high enough to accept it. Finally, it's up to us

to be judged on the basis of our safety policy!

[page 65]

Used in isolation, these techniques are not necessarily very robust, but their combined use will provide sufficient credibility in the fact that the certificate we're going to use is the right one. And that no one has succeeded in deceiving us.

Bear in mind, however, that this does not protect against all attacks on connection encryption.

[page 255]

Once you've been able to establish with a sufficient degree of confidence that the certificate presented corresponds to the site you wish to visit, you can click on the *Accept risk* button *and continue* to the warning page. The certificate will then be accepted by the web browser, and the site will appear.

2. Chad R Dougherty, 2008, *MD5 vulnerable to collision attacks* [<https://www.kb.cert.org/vuls/id/836068>] (in English).

3. Julien Cadot, 2017, *SHattered: Google has broken the SHA-1 hash function* [<https://web.archive.org/web/20211122073218/https://www.numerama.com/tech/235436-shattered-google-a-ca-sse-la-methode-de-chiffrement-sha-1.html>].

41.1.1 The special case of onion services

It's currently very difficult to obtain valid certificates for onion services (sites whose domain name ends in *.onion*), so you'll always get a warning message from the Tor Browser when you want to connect to such a site in *https*.

In most cases, the certificate used by the onion service is self-signed: this means that the site itself has signed its own certificate. You can then check the validity of the certificate by other means, as described in the previous section.

[page 323]

In the case of onion services that are also accessible with a domain name


In the "classic" version, the certificate presented is generally a valid certificate for this domain name, but not for the *.onion* name. It is then sufficient to check that the domain name for which the certificate is valid actually corresponds to the site to which you wish to connect.

In any case, the confidentiality and authenticity of the connection to an onion service are ensured by the onion routing protocol and the "rendezvous point" system: if you're sure that the *.onion* address you're connecting to is correct, then you can be convinced with a fairly high degree of confidence that you're accessing the corresponding onion service.

[page 261]

[page 266]

41.2 Find the fingerprint of an installed certificate

This fingerprint can be viewed by clicking on  in our browser to afficher the Firefox or Tor Browser menu and then going to *Settings*. Choose the *Privacy & Security* page, then scroll down to the *Certificates* section. Here, click on *Afficher les certificats...* Certificates for sites already installed can be found by selecting the *Servers* tab in the window that opens. Finally, by selecting the desired site from the list and clicking on the *View...* button, you can view the certificate's digital footprint. The same operation can be performed for certification authorities by selecting the *Authorities* tab.

Using a visual keyboard in Tails


🔄 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

🕒 *Duration: a few minutes.*

We saw in the first volume that a computer can be compromised in hardware...
ment. In particular, it can contain hardware keyloggers that can record everything typed on the keyboard. Texts you write, actions you perform, but above all the passwords you enter.

When in doubt about whether to trust a computer on which you're going to use Tails, it's possible to use a visual keyboard (formerly known as a "virtual keyboard") to make it inefficient to retrieve keystrokes from the keyboard. Attention however, this method does not protect against a bug recording the affichage page 31 of the screen.

A visual keyboard is software that looks like a keyboard and lets you enter characters without using the computer's hardware keyboard. It can be used with a variety of pointing devices, such as a mouse, touch screen or touchpad.

The GNOME desktop environment provided by Tails allows you to use a visual keyboard among the various accessibility options available. To do this, click on the *Universal Access* icon (🗑️) in the top bar, then activate the *Visual Keyboard* option. Alternatively, press  (⌘) on a Mac), type param, then click on *Settings*: you can then activate the *Visual Keyboard* option in the *Input* section of the *Accessibility* page.

Once activated, the visual keyboard affiche as soon as you have the opportunity to enter text. It then suffit to type your passwords using your mouse, touchpad or other pointing device.

It should be noted that this visual keyboard can be activated from the Tails welcome screen, so it can also be used to enter the passphrase to unlock the persistent volume.

Configuring and using the mail client Thunderbird



🔄 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

🕒 *Duration: Fifteen to thirty minutes.*

This section describes how to set up and use the Thunderbird e-mail client for all your e-mail-related tasks. It has been tested with Thunderbird version 91. The interface may be slightly different with more recent versions.

Under Debian, if Thunderbird is not yet installed, you need to install the package `thunderbird` following the recipe to install software. page 135

43.1 Launch Thunderbird

Launch Thunderbird by pressing  ( on a Mac), type `th` then click on *Thunderbird Messaging*.

When Thunderbird is launched and no e-mail account has been set up, a configuration tab entitled *Configure your existing e-mail address* appears to help you add your first account to an existing e-mail address.

Nevertheless, it's generally best to take a little time to set up some of Thunderbird's privacy options before configuring this first e-mail account: you can therefore close this tab by clicking on *Cancel*, so that you can carry out the operations described below. However, if you wish to set up an e-mail account right away, you can go straight to the corresponding section. next
page.

43.2 Configuring onion routing for Thunderbird

If you're using Tails, Thunderbird is already configured to work with Tor. You can proceed directly to the next step next
page.

If you're using Thunderbird on a Debian system and want it to use the Tor network to connect to your mail server, you'll need to configure it accordingly. page 261

First, install the `tor` Browser if it's not already installed. Then : page 313

- open the *Preferences* tab by going to  to afficher the Thunderbird menu.

1. The OpenPGP protocol, used for email encryption [page 295], has been integrated and activated by default since Thunderbird version 78.2.1. We therefore no longer need to install the Enigmail add-on, which was required with previous versions.

- Make sure you are in the *General* section in the left-hand column.
- Scroll down to the *Connection* section.
- Click on *Settings...* under *Configure how Thunderbird connects to the Internet*.
- Check *Manual proxy configuration*.
- Fill the *SOCKS Host* field with `127.0.0.1` and *Port* with `9150`.
- Check *SOCKS v5*.
- Check *Use remote DNS when SOCKS v5 is active*.
- Click on the *OK* button.
- Close the preferences tab by clicking on the corresponding **X** button.

From now on, with this new configuration, to be able to send and receive emails, you'll always have to open the Tor Browser and connect to Tor. If, for some reason, Tor Browser isn't working, sending and receiving emails won't work either.

43.3 Set a master password in Thunderbird

By default, Thunderbird proposes to remember passwords for access to configured e-mail accounts. In addition, if you wish to use the OpenPGP encryption or digital signature features described in the next chapter, Thunderbird will need to store your *private key* in its configuration.

To restrict access to these passwords and private keys stored by Thunderbird, a *master password* must first be defined. This passphrase will be requested each time Thunderbird is opened.

To do this, in Thunderbird, click on **☰** to afficher the menu, then on *Preferences*. In the list on the left, choose *Privacy & Security* and scroll down to the *Passwords* heading. Check the *Use master password* box. A new window opens, asking for a password to protect the key. Choose a good passphrase, type it twice and click *OK*. A message confirming that the master password has been changed appears. Confirm with *OK*, then close the preferences tab by clicking on the corresponding **X** button.

43.4 Setting up an e-mail account

To add a new, existing e-mail account to Thunderbird, click on **☰** to afficher Thunderbird's menu and go to **+** *New* → *Existing mail* account.... This opens a tab entitled *Configure your existing e-mail address*.

You then need to fill in the first two fields: *Your full name* and *E-mail address*. The name we put in the *Your full name* field will appear in the e-mails we send out, and will therefore be readable by our correspondents and by the intermediaries forwarding our messages. We therefore suggest you fill in this field with the pseudonym you want to appear in your e-mail headers.

However, it is not necessary to fill in the *Password* field, unless you want Thunderbird to remember your password for connecting to this e-mail account (in which case we strongly recommend that you first set a *master password* as described above).

Once you've entered your details, click on *Continue*.

If *Configuration found at mail provider* affiche, the automatic configuration has worked. If Thunderbird is unable to find the configuration automatically, it's possible to search the official documentation of our mail host to check the specific settings for IMAP, POP and SMTP. If

this information can't be found on the web site of the mail host, but it's possible to find the mail contact of the admins and ask them for it.

The wizard then offers a choice of two protocols, IMAP or POP. Select the one that suits you best and click on *Done*, then close the account configuration tab by clicking on the corresponding **X** button. page 292

Thunderbird is now ready to receive messages. You can repeat the procedure if you wish to add further e-mail accounts. Otherwise, you can skip ahead to the next section, devoted to advanced Thunderbird configuration.

43.5 Advanced Thunderbird configuration

Once Thunderbird has been set up for an e-mail account, you may want to optimize its configuration to make it more user-friendly or to reduce IT security risks.

To do this, click on **☰** to afficher the Thunderbird menu, then choose *Account Settings*. We're not going to take an exhaustive tour of the configuration options, but just a few that seem useful to us.

43.5.1 Message retention time

First of all, if you have chosen to use the POP protocol, in the *Server settings* section you can set the time after which messages will be deleted from the servers after repatriation. This is, of course, without any great guarantee and depends in particular- we can only hope that he really does delete page 42 of our data. []

43.5.2 Ports used

Finally, if you're having problems sending or receiving e-mails, it's possible that the protocol ports you're using are not the correct ones with the default settings. If this is the case, you'll need to make changes based on the configuration information available from your e-mail host.

These settings are accessed by clicking on the e-mail address in the left-hand column, then on *Account Settings* in the top right-hand corner. A new tab opens where you can change the SMTP port in the *Outgoing Server (SMTP)* section at the very bottom. Then click on *Modify SMTP server...* and finally change the *Port* number to the one provided by our host. To modify the incoming server, return to the left-hand column, select *Server settings* and modify the *Port* corresponding to *the Server type* previously chosen (IMAP or POP3).

43.5.3 Using an onion service

If your mail host has set up onion services, you can configure Thunderbird to use the corresponding onion addresses. page 266

To find them, you need to look for the information published by our mail host: the onion addresses and their ports for SMTP, IMAP and/or POP services. This information is not always readily available. You can search the Internet using the following keywords: "*configuration smtp onion service [and the name of the mail host]*". If you can't find what you're looking for, you can also ask the people who administer the mail hosting service directly.

Once the onion addresses have been found :

- configure onion routing in Thunderbird as described above. page 329

- For the POP or IMAP server: in the left-hand column below the e-mail account concerned, go to *the Server settings* section, then replace the address indicated in *Server name* with the address of the onion POP or IMAP service.
- To modify the SMTP server address, go to the *Outgoing Server (SMTP)* section at the very end of the left-hand column, select the e-mail account concerned, click on *Modify...* and finally replace the SMTP server address in *Server Name* with the address of the onion SMTP service.


Use OpenPGP encryption in Thunderbird


The Internet standard ¹ OpenPGP is a cryptographic format that enables digital signatures to be created and verified, and messages and files to be encrypted and decrypted.

This chapter describes how to use OpenPGP in Thunderbird to manage keys and encrypt or sign messages. However, certain uses of OpenPGP that are not possible in Thunderbird are covered in the next chapter.

page 343

44.1 Create a key pair

 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

 *Duration: Fifteen minutes to an hour.*

This tool details the creation and part of the management of an encryption key pair. It's worth recalling a few basic notions that should always be borne in mind:

page 249

- Not all encryption keys use the same algorithm. We've talked about RSA encryption, but there are several others.
- The algorithm does not strictly define the size of the key, which can be varied to play with security levels.
- Some keys have expiry dates, others do not.

page 251

44.1.1 Generate key pair

First of all, before generating an OpenPGP key pair, you need to have defined a *master password*, as detailed in the previous chapter. This passphrase is requested each time Thunderbird is opened. It is used to restrict access not only to registered passwords, but also to the private key you are about to create.

page 330

To create our new key pair, in Thunderbird, click on  → *Tools*

→ *OpenPGP key manager*. Choose *Generation* → *New key pair*. An *Add a personal OpenPGP key for [...]* window opens. Check that the *Identity* selected corresponds to the contextual identity used, as well as the e-mail address associated with it.

page 243

It is advisable to choose a *key expiration* date. If this is the first time you've created a key pair, choose an expiry date between one year and

1. Wikipedia, 2014, *Internet Standard* [https://fr.wikipedia.org/wiki/Standard_Internet].

two years, for example. So that you don't forget to renew your key on time, it's a good idea to make a note of the expiry date somewhere.

In *Advanced settings*, the default *Key type* is *RSA*. It's advisable to select *ECC* (*elliptic curve*) here, as the corresponding cryptographic algorithms offer equivalent security to RSA-type keys, while being more efficient. However, it is still possible to use an RSA key.

If *RSA* is chosen as the *Key Type*, the default *Key Size* of 3072 bits is considered secure until beyond 2030² but if you wish to protect your communications more strongly or for longer, we advise you to choose the highest key size available, i.e. 4096 bits. In the case of ECC keys, on the other hand, it is not currently possible to choose the key size.

Once you have selected the key parameters, click on *Generate key* and then on *Confirm*.

This can be almost instantaneous, or it can take several minutes. This is the time to move your mouse, use your keyboard or even use your hard disk, if possible, to help your computer generate random data. These are necessary for the key generation process.³

Once this operation has been completed, our key will appear in bold in the *OpenPGP Key Manager*. It may happen that the key is not visible. In this case, move up or down in the key list.

44.1.2 Save your private key

Once the key creation stage is complete, it's time to think about how we're going to save our key pair, and in particular our private key: since it's secret, we mustn't leave it lying around. The private key must be accessible only to the person who is supposed to have access to it. The best way to do this is to keep the key pair on an encrypted volume, whether it's a USB key, an internal or external hard disk, or the Tails persistence.

If you're saving to Tails persistence, it's a good idea to have a backup of your live system:

- From the *OpenPGP Key Manager*, select the key and choose *File* → *Save one or more secret keys to a file*.
- Choose where to place the file and its name, then click *Save*.
- Then choose a passphrase to protect the backup of the secret key. This can be the same passphrase you chose earlier as Thunderbird's main password, as it's virtually the same information we're protecting: our secret key. Then click on *OK*.
- A dialog box should confirm that *the keys have been saved correctly*. You can close it.
- Check that the backup is in a safe place.

44.1.3 Keep a revocation certificate safe

If adversaries get their hands on our private key, or if we simply lose it, we need to *revoke* it, so that our correspondents are aware that we have lost it.

2. Agence nationale de la sécurité des systèmes d'information, 2020, *Mécanismes cryptographiques - Règles et recommandations concernant le choix et le dimensionnement des mécanismes cryptographiques* [https://www.ssi.gov.fr/uploads/2021/03/anssi-guide-mecanismes_crypto-2.04.pdf], p. 20.

3. Zvi Gutterman, Benny Pinkas, Tzachy Reinman, 2006, *Analysis of the Linux Random Number Generator* [<http://www.pinkas.net/PAPERS/gpr06.pdf>].

page 145

page 116




page 151

page 330



must no longer use the corresponding public key. A *revocation certificate* is used for this purpose.

The revocation certificate comes in the form of a file or a few lines of text, which we'll need to store in a safe place - on an encrypted USB key, with a trusted person or on a well-hidden piece of paper, for example. Anyone who has access to this file can revoke our public key, thus preventing us from communicating in encrypted form.

When creating an OpenPGP key pair, Thunderbird automatically creates a revocation certificate, but hides it in its configuration folder. To find it :

- launch Files: press  ( on a Mac), type `fic` then click on *Files* ;
- in the left-hand panel, go to *Personal folder* ;
- click on  then *Afficher les fichiers cachés* ;
- open the `.thunderbird` folder ;
- find the folder with the weird name ending in `.default` (for example `7u6xu6tq.default-default` or `profile.default`) and open it;
- our private key's revocation certificate is stored in a file whose name begins with the key's identifier ⁴ and ends with `_rev.asc` (e.g. `0xC7BF166A096820DA_rev.asc`);
- double-click on the file to open it.

Depending on our choice, we can then :


- save it by clicking on  then *Save as...* and choose a clear file name. For example `Revocation certificate for key 0xC7BF166A096820DA.asc` ;
- print it by clicking on the printer icon in the  menu.

If our private key were to be compromised, we would use this certificate to revoke page 340 the associated public key.

44.1.4 Setting up encryption for an e-mail account

Asymmetric cryptography can be used to encrypt or sign e-mails, or both. You therefore need to configure the e-mail account you want to use with the key pair you've just generated.

To do this:

- click on  to afficher the Thunderbird menu, then open *Account Settings* ;
- click on the *End-to-end encryption* section of the mail account to be edited;
- choose the key corresponding to our contextual identity instead of *None. Do not use OpenPGP for this identity*.

Further down, in the *Default settings for sending* messages, you can select different options.


By default, emails are not encrypted, and encryption must be manually activated for each email. You can *enable encryption for new messages*; you will then have to disable encryption to write to someone who does not use OpenPGP.


We can also check the *Sign unencrypted messages* box to sign all emails we send from this account (encrypted messages are always signed).

⁴ In case of doubt, you can find our key identifier (without the initial `0x`) in Thunderbird's *OpenPGP Key Manager*, opposite the corresponding contextual identity.

signed by default). This allows recipients to authenticate all emails, including those that are not encrypted. It also shows them that OpenPGP is being used. Be careful, however, as this cryptographically proves that the e-mail was sent by a person holding the corresponding secret key, which is not always desirable.

44.2 Exporting and sharing our public key



 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

 *Duration: a few minutes.*

[page 338]

To send us encrypted e-mails and to verify our e-mail signature, our correspondents need our public key. But before they can use it, they also need to have verified the fingerprint of this key.

44.2.1 Send our public key by email


It is possible to send our public key by e-mail. In the message window, click on the  button just to the right of the  *Attach button*, then tick *My OpenPGP public key*. Our key will then be automatically added as an attachment when the email is sent.

44.2.2 Publish your public key on key servers

[page 243]

If the existence of the contextual identity to which the key corresponds is not itself confidential, we can publish our public key on a key server, so that anyone wishing to send us encrypted e-mails can download it for this purpose.

Start by exporting your public key to a file, which can then be shared on a server or via an encrypted USB key. The procedure is the same under Tails or with an encrypted Debian :

- In Thunderbird, click on  → *Tools* → *OpenPGP Key Manager*.
- Select the OpenPGP key you wish to export; in the menu, click on *File* → *Export public key(s) to a file*; choose an export location and file name, then click on *Save*.

Then publish the key on a key server:


[page 315]

- Open Tor Browser and enter the address <https://keys.openpgp.org/>.
- Click on *upload*.
- Click on *Browse...* and choose the file to which you have exported your public key.
- Click on *Upload*. A page confirms receipt of the key.
- Visit the link in the confirmation email (copy and paste it into the Tor Browser address bar) to confirm that it's us behind the email address associated with the published key.

44.2.3 Obtain a key fingerprint

If we transmit our public key by unauthenticated means, we need to send our correspondent the fingerprint (see page 54) of our key by authenticated means, so that she can check that it's the right key belonging to the right person.


To obtain the fingerprint of our key :


- In Thunderbird, click on  → *Tools* → *OpenPGP Key Manager*.
- Double-click on our OpenPGP key to afficher the *Key Properties*.

- Write down or copy the key fingerprint for secure sharing.

Methods for sharing the fingerprint over a secure channel and verifying the authenticity of the key (see next page) are explained below.

44.3 Import, verify and export public keys

 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

 *Duration: From a few minutes to half an hour.*

We use other people's public keys to encrypt the emails we send them, and to verify the authenticity of the messages they have signed.

To obtain these public keys, we need to import them. Before using them, we need to check their authenticity, to make sure we have the right public key from the right person. It is also sometimes useful to export these keys to a file for use in other software.


this page
next
page 339

44.3.1 Import a public key

The aim of this chapter is to import an OpenPGP key, which we'll use to verify digital signatures or encrypt messages. The procedure is the same under Tails or with an encrypted Debian.

Importing a key does not mean checking that it actually belongs to the supposed owner. We'll see in the next section that this requires other operations, such as studying its signature or digital fingerprint.

next
page.

In Thunderbird, key import goes through the *OpenPGP Key Manager*. To access it, click on  → *Tools* → *OpenPGP Key Manager*.

If the key is available in a

In the *OpenPGP Key Manager*, click on *File* → *Import public key(s) from file*. In the window that opens, select the file containing the key, then click *Open*.

You can then proceed to the import confirmation stage (see next page).

If you want to search for the key online

Still in the *OpenPGP Key Manager*, click on *Key server* → *Re-search online keys*.

In the window that opens, type the e-mail address or identifier corresponding to the key you're looking for, e.g. `guide@boum.org`, `0x326F9F67250B0939`⁵ or `D4874FA4F6B688DC0913C9FD326F9F67250B0939`, and select *OK*. You'll need to check the fingerprint afterwards, as we'll see later.

Note that if you've previously configured Thunderbird to use onion routing, you'll also need to have Tor Browser running for the online key search to work.

page 329

5. This is the short identifier of a key, which is not sufficient for uniquely selecting a key. Riseup, 2017, *Best practices for using OpenPGP* [<https://help.riseup.net/en/security/message-security/openpgp/best-practices#-don't-know-%C3%A0-identifier-de-cl%C3%A9>].

Import confirmation

Once Thunderbird has found the key (either in the specified file or online, depending on the procedure used just before), a results window opens, which displays the key's full identifier and associated e-mail addresses. If this is the key you wish to import, select *Accepted (unverified)* and click *OK*.

If the import is successful, a *correctly imported keys* window opens, with a summary of key information. Close it with *OK*.

The imported key should now be visible in the *OpenPGP Key Manager*.

It is still necessary to check its authenticity.

44.3.2 Verify the authenticity of a public key

page 253

When using asymmetric cryptography, it's crucial to ensure that you have the true public key of your correspondent. Otherwise, you leave yourself open to attack by the monster in the middle.

page 254

First of all, we'll need to choose a method for ensuring that we have the right public key. We'll then tell Thunderbird that we trust this key.

page

Depending on the requirements of our threat model and our possibilities, we can choose different ways of verifying the authenticity of a public key. Let's say we need to verify the authenticity of Ana's public key.

63

Transmit the key to yourself via a secure channel...

Whenever possible, the easiest way is to hand over the file containing the public key, using a USB key for example. Ana then exports (see opposite page) her public key to a file, which she stores on a USB key, possibly encrypted (see page 145), which she then gives to us. We then import (see previous page) Ana's public key directly from this file.

...or transmit the fingerprint to each other via a secure channel.

One of the disadvantages of the previous method is that it requires a computer file to be transferred by a secure means. This is not always possible. Fortunately, it doesn't have to be: you just need to obtain, by a secure means, a checksum of the public key, known as a "fingerprint".

page

Ana can publish her public key on the Internet, for example on her blog or on a key server. On our side, we download this key unauthenticated, then check that the key fingerprint matches the one Ana sent us *authenticated*. To see Ana's key fingerprint obtained from the Internet, you'll need to import it (see previous page) into the *OpenPGP Key Manager*, then double-click on her key.

53

What do we gain by using this method? Instead of having to pass around a file, it's sufficient to pass around a line of characters like this:

```
A490 D0F4 D311 A415 3 E2B B7CA DBB8 02 B2 58 AC D84F
```

For example, Ana, who is a well-organized person, can carry a copy of her public key imprint written on a piece of paper at all times. We then suffice to pass it to her: no need for a computer or USB key.

If we can't meet Ana, she can also send us the print by post, and we can call her to read it over the phone. The verification won't be as good as seeing each other directly, but it's still more difficult.

for opponents to send us postal mail with his key, and to answer Ana's phone number by reading out her fingerprint and imitating her voice.

It gets even more complicated if we don't know Ana. In this case, we'll have to trust people who claim to know her. Once again, there's no magic formula, but combining different means of verification can complicate the task of possible adversaries wishing to mount a "monster in the middle attack": we can ask several people who claim to know Ana rather than just one, use several different means of communication, *and so on*.

page 254

Registering trust in a key

Once we've established trust in Ana's key, it's useful to inform Thunderbird that it can trust this key.

To do this, open Thunderbird's *OpenPGP Key Manager* by clicking on  → *Tools* → *OpenPGP Key Manager*.

Once you've located Ana's key in the main window, double-click on it to affview the key's details. Check that it's the right key, for example by verifying its fingerprint. In the *Your acceptance* tab, select *Yes, I have checked in person that the fingerprint of this key is correct*, then validate by clicking *OK*.

Thunderbird now knows that Ana's key is trusted.



TO FIND OUT MORE...

In Thunderbird, when we trust a key, our choice remains on our computer only. To make the web of trust work (see page 257), the OpenPGP protocol allows you to sign a key and make that signature public, so that any user of the web of trust can benefit from the verifications you've made.


For the time being, it is not possible to use Thunderbird's interface for public signing. It's only possible in the system's OpenPGP keychain, which can be accessed using the *Kleopatra* application, for example.

44.3.3 Exporting a public key to a file

The purpose of this tool is to export an OpenPGP key, for example, for use with other software.

The file created by this operation will contain the public key needed to encrypt messages intended for the corresponding identity, or to verify signatures made by this identity.

To do this:


- In Thunderbird, click on  → *Tools* → *OpenPGP Key Manager*.
- Select the OpenPGP key you wish to export; in the menu, click on *File* → *Export public key(s) to a file*; choose an export location and file name, then click on *Save*.


44.4 Manage your key pair: extend, change or revoke it

When creating our key pair, we were able to choose an expiry date. Before the key pair expires, it is possible to change the expiration date to extend its validity. However, as technologies evolve, we may want to change our key pair and transition to a new one. Finally, it sometimes happens that a private key is compromised and needs to be revoked.

this page
this page
this page


44.4.1 Extending your pair of keys

 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

 *Duration: a few minutes.*


If our key pair is about to expire, but there's no reason to switch to a new pair, we can extend its validity.


To do this:

- in Thunderbird, click on  → *Tools* → *OpenPGP Key Manager* ;
- double-click on our key pair ;
- click on *Modify expiry date* ;
- select *The key will expire in* and choose a number of months, for example twelve or twenty-four months (one or two years);
- click on *OK* to validate.

Here we go again for another season with our pair of keys!

44.4.2 Transition to a new key pair

 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

 *Duration: Fifteen minutes to an hour.*

Before our key pair expires, or when advances in cryptography force us to use more secure keys, we'll need to create a new key pair.


To do this, use the create key pair tool (see page 333).

We'll then export our new public key (see previous page) and send it to the people we communicate with.

Some time later, we'll be able to revoke our old key (see this page).

However, we will keep our old private key, so that we can decrypt messages received previously, encrypted with the old public key.

44.4.3 Revoke a key pair

 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

 *Duration: Fifteen to thirty minutes.*

If our own key pair is compromised, for example if we've lost our system or suspect it's been hacked, the key is to let our correspondents know. This way, they'll know that the key is no longer trustworthy and can stop using it.

To do this, we'll use the revocation certificate created earlier (see page 334) with our key pair.





Warning: the following instructions will irreversibly revoke our key. Use only when necessary!

Preparing the revocation certificate

First of all, we need to find the revocation certificate we saved when we created our key pair. We had stored it in a safe place, for example on an encrypted USB key, with a trusted person or on a well-hidden piece of paper. [page 334]

If it was on a piece of paper, you need to create a file containing the revocation certificate information:

- launch Text Editor: press  ( on a Mac), type `gedi` then click on *Text Editor* ;
- in the document, type precisely the part that begins with `-----BEGIN PGP PUBLIC KEY BLOCK-----` and ends with `-----END PGP PUBLIC KEY BLOCK ;`
- Save the file in `.asc` format, for example `revocation.asc`.

Otherwise, if you have saved the file containing the revocation certificate on another medium (encrypted USB key or similar), you must first :

- open this file by right-clicking on it, then *Open with another application*;
- in *Choose an application*, select *Text editor* and click on *Select* ;
- remove the `:` character at the beginning of the line `-----BEGIN PGP PUBLIC KEY BLOCK-----`;
- click on *Save* and close the text editor.

The revocation certificate is now ready. We can now use it to revoke our key.

Revoke our OpenPGP key

To revoke an OpenPGP key, you need the corresponding public key. If our private key has been compromised, we may no longer have access to the system it was on. So, if you don't have the public key you want to revoke, start by importing it (see page 337).

Next, import the revocation certificate.

In Thunderbird, open the *OpenPGP Key Manager* by clicking on  →

Tools → *OpenPGP Key Manager*. Then :

- choose *File* → *Import revocation(s) from a file* ;
- select the file containing the revocation certificate, then click on *Open* ;
- the revoked key appears grayed out.

If you've already published your public key on a key server, you now need to publish the revoked key there, following the recipe below.

Publish revoked public key

If our public key was previously published on a key server, the best thing to do is to publish our revoked key again, so that our public key is now also revoked there, allowing all our correspondents to be notified by updating it from the key server.

To do this, whether under Tails or Debian, follow the recipe for publishing your public key on key servers. [page 336]

Once this synchronization is complete, all that remains is to let our correspondents know.

Notify our correspondents of our key revocation

The most important step in revoking our key is to notify our correspondents so that they no longer use it.

To do this, you can choose :

- email them the revocation certificate, which they can then import to revoke our public key in their keyring;
- export our revoked public key (see page 339) and then send it to them by e-mail, so that they can import it again;
- ask them to update our revoked key from the key server on which we've published it; there's no ready-made recipe for this, between sending them an encrypted e-mail, letting them know in person, *etc.*


44.4.4 Revoke a correspondent's public key


If one of our correspondents has informed us that her key pair has been compromised and she has revoked it, we need to update her key on our computer so that Thunderbird takes this revocation into account.

To do this:





- if our correspondent has sent us the revocation certificate for his key, follow the recipe above to import this certificate;
- if it has sent us its revoked public key, it must be imported again (see page 337);
- if it has published its revoked public key on a key server, then we need to update our copy of the key by importing it again from the key server (see page 337).

44.5 Encrypt and/or sign emails in Thunderbird

 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

 *Duration: a few minutes.*

 page 329 Once Thunderbird has been started and configured :

- click on the *Write* button to start writing a new message;
- a *Redaction* window opens, in which we'll write our email;
- if you wish to encrypt the email, click on the  *Encrypt* button, if it is not already selected (the padlock is crossed out when encryption is disabled); you can also activate email encryption by clicking in the *Security* → *Encrypt* menu;
- if you wish to digitally sign the email, click on  *OpenPGP* → *Sign digitally* or in the menu *Security* → *Sign digitally* ;
- if our correspondent does not already have our public key, it is possible to attach it to the email automatically by clicking on the  button just to the right of the  *Attach* button and then ticking *My OpenPGP public key* ;
- once you've completed your e-mail, click on *Send*.

 page 252

If the person receiving the email also uses Thunderbird, they will see an *OpenPGP* button with :

- a closed padlock if the email is encrypted;
- a stamp if the email is signed.

Clicking on this button displays details of the encryption and signature.

If the person does not possess the private key for which the message was encrypted, a message *The secret key needed to decrypt this message is not available* will appear instead of the email body.

Use OpenPGP encryption in the office

The Internet standard ¹ OpenPGP is a cryptographic format that can be used to create and verify digital signatures, and to encrypt and decrypt messages and files.


Most of the tools in this guide use OpenPGP, using Thunderbird wherever possible to manage OpenPGP keys, as its interface is more ergonomic and that's how it works in Tails. However, some uses of OpenPGP that are not possible in Thunderbird are grouped together in this chapter.

[page 333]

Thunderbird and the rest of our desktop environment (whether we're using Debian or Tails) use two different OpenPGP keyrings. The keys we'd like to have in both keychains have to be manually exported from one, then imported into the other.

45.1 Import a key into the office keychain

 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*


 *Duration: a few minutes.*

The purpose of this tool is to import an OpenPGP key into the desktop OpenPGP keychain. Note that this keychain is not the same as Thunderbird's.

If you're using an encrypted Debian, you first need to install the *Kleopatra* software package, which contains the key management tool you'll be using. If you're already using Tails, this package is already installed.

[page 119]

[page 134]

 When *Kleopatra* is launched, it may display a warning message entitled *Kleopatra Automatic Test Results*, in which *the sddaemon Configuration Check* appears to have failed. This isn't serious, but it can quickly become disturbing. So that this message doesn't appear every time *Kleopatra* starts up, you can uncheck the box *Launch these tests at startup* and then click *Continue*. Another possibility is to install the *sddaemon* package, even though it won't be of any use to us.

[page 135]

45.1.1 Import a secret key

You may need to import your secret key (also called private key) into the office keychain, for example to sign or decrypt files, or to sign public keys.

1. Wikipedia, 2014, *Internet Standard* [https://fr.wikipedia.org/wiki/Standard_Internet].

[page 334] If it's in Thunderbird's keychain, start by saving your secret key, in the form of a file with the `.asc` extension.

Then import it into the desktop keychain by double-clicking on it.

A *You have imported a private key* window appears. The software asks *Is this your own key?* Answer *Yes*.

A new *Certificate Import Result* window appears. Confirm with *OK*.

45.1.2 Import a public key


Importing a public key into the desktop OpenPGP keychain lets you verify digital signatures or encrypt files.


If you've received or downloaded a file containing the key (usually with the `.asc` or `.pub` extension), you suffice to double-click on the file to import it into the desktop hole- seau.

[page 339] If you want to retrieve a key you already have in Thunderbird's keychain, you'll need to export it to obtain the file containing the key to be imported. You can then import the key by double-clicking on the file.

A *You have imported a new certificate (public key)* dialog box appears. The software offers to guide you through the process of certifying its authenticity. It's a good idea to select *Yes* if you have a secret key (as this is needed to sign the key you've just imported).


45.2 Signing a key

 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

 *Duration: a few minutes.*

The purpose of this tool is to sign an OpenPGP key in the desktop OpenPGP keychain. Note that this keychain is not the same as Thunderbird's.

[page 119] If you're using an encrypted Debian, you first need to install the *Kleopatra* software package, which contains the key management tool you'll be using. If [page 134] you're using Tails, this package is already installed.

 When *Kleopatra* is launched, it may display a warning message entitled *Kleopatra Automatic Test Results*, in which the *sddaemon Configuration Check* appears to have failed. This isn't serious, but it can quickly become disturbing. So that this message doesn't appear every time *Kleopatra* starts up, you can uncheck the box *Launch these tests at startup* and then click *Continue*. Another possibility is to install the *sddaemon* package, even though it won't be of any use to us.

[page 135] But why sign a key? Let's say we've first verified the authenticity of Ana's key by following the recipe described in the previous chapter. It is then useful to inform OpenPGP that it can trust this key. This operation is called signing the key. *Kleopatra* also calls it *certifying* the key. The procedure is the same under Tails or with an encrypted Debian.

[previous page.] To be able to sign a key, we first need to import our secret key into *Kleopatra*.

Next:


- Go to *Kleopatra*, by pressing  ( on a Mac) to open the activity overview, then typing `kleo` and clicking on the corresponding software.


- If the key you wish to sign is not present, import it.
- Once you've located Ana's key in the main window, double-click on it to affview the key's details. Check that it's the right key, for example by checking its fingerprint (found at the bottom of the window).
- Then click on *Certify*.
- Enter the passphrase for our secret key in the dialog box that afficheiche if necessary.².
- A *Certification successful* window should appear. Click on *OK*.

opposite page

OpenPGP now knows that Ana's key is trusted.

45.3 Verify a digital signature

 As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.

 Duration: a few minutes.

The purpose of this tool is to verify the authenticity of a file with an OpenPGP digital signature.

page 252

If you're using an encrypted Debian, you first need to install the *Kleopatra* software package, which contains the key management tool you'll be using. If you're using Tails, this package is already installed.

page 119

page 134



When *Kleopatra* is launched, it may display a warning message entitled *Kleopatra Automatic Test Results*, in which the *scdaemon Configuration Check* appears to have failed. This isn't serious, but it can quickly become disturbing. So that this message doesn't appear every time *Kleopatra* starts up, you can uncheck the box *Launch these tests at startup* and then click *Continue*. Another possibility is to install the *scdaemon* package, even though it won't be of any use to us.



page 135

In order to verify the digital signature of a file, you must first find the public key of the person or group who produced the signature, and then import this key into the office keychain. In general, the public key required to verify the signature can be downloaded from the website where the file and its signature were retrieved. If the signature has been created by one of our correspondents, it is her public key that must be used to verify the signature.

page 343

This signature takes the form of a small file, usually bearing the same name as the file containing the signed data, with a *.sign* extension, *.sig* or *.asc*.

45.3.1 Perform signature verification

- If the signature file ends with the extension *.sign*, right-click on it and choose *Rename....* Remove the trailing *n* so that it ends in *.sig*.
- Go to *Kleopatra*, by pressing the key ( on a  *mac*) to open the activity overview, then typing *kleo* and finally clicking on the corresponding software.
- In the toolbar at the top of the window, click on *Decrypt/Verify....*
- In the window that opens, select the signature file.


A *Verify Files* window appears. It contains the progress, then the result of the verification.


² If you've already typed in your passphrase shortly before, it won't be requested again. OpenPGP keeps it in memory for ten to thirty minutes.


45.3.2 Interpret the result of the verification

- *Valid signature* means that the file has been signed by the key specified under *With certificate*.
- *Unable to verify the data* can mean two things:
 - If the result box shows *With certificate* followed by the name of a key from our keychain, this means that the file is indeed signed by the specified key, but that we have not confirmed the authenticity of this key. If you wish to verify this, follow the corresponding tool (see page 338), then sign the key (see page 344).
 - If the result box shows *With certificate unavailable* followed by a key identifier, this means that the file is indeed signed, but that the public key needed to verify the signature is not in the desktop OpenPGP keychain. In this case, find the key and import it into the desktop keychain (see page 343) using the *Import* button.
- *Invalid signature* means that the file verified does not correspond to the one signed. You may have uploaded the wrong file, the wrong signature file, or been the victim of an attack. In all cases, the downloaded file cannot be considered authentic.

45.4 Signing data

 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

 *Duration: a few minutes.*

 The purpose of this tool is to digitally sign data. In particular, this can enable other people to authenticate a message, document, software, *etc.*, as originating from us. This tool requires you to have first created a key pair and imported its secret key into the desktop OpenPGP keychain.







45.4.1 Signing text

This method only works for signing text. To sign any other type of file, follow the next section.

To sign the text :

- Go to *Kleopatra*, by pressing  ( on a Mac) to open the activity overview, then typing *kleo* and clicking on the corresponding software.
- In the toolbar at the top of the window, click on *Notepad*.
- In the *Notepad* tab, type or paste the text to be signed.
- Go to the *Recipients* tab.
- Check *Sign as* (choosing the right contextual identity if you have more than one).
- Uncheck *Encrypt for me* and *Encrypt for others*.
- Click on *Notepad Sign*.
- Enter the passphrase for our secret key in the dialog box that afficheiche if necessary.³



Signed text can be found in the *Notepad* tab. It can be copied and pasted into a file.

3. If you've already typed in your passphrase shortly before, it won't be requested again. OpenPGP keeps it in memory for ten to thirty minutes.

45.4.2 Signing a file

To sign a file, we first need to import our secret key into the office keychain. page 343


To sign the file :

- Go to *Kleopatra*, by pressing  ( on a Mac) to open the activity overview, then typing `kleo` and clicking on the corresponding software.
- In the toolbar at the top of the window, click on Sign/encrypt....
- Select the file to be signed and click *Open*.
- Check *Sign as* (choosing the right contextual identity if you have more than one).
- Uncheck *Encrypt for me* and *Encrypt for others*.
- Click on *Sign*.
- Enter the passphrase for our secret key in the dialog box that afficheiche if necessary.⁴
- A *Signature Success* message should afficher.

The signing process can take up to several minutes, depending on the size of the file and the power of the computer being used. Once the signature is complete, it takes the form of a small file with the same name as the original file, but ending with the `.sig` extension, located in the same place as the original file. Each time the original file is transmitted, this signature file must be attached so that recipients can verify its authenticity. What's more, so that recipients can verify our signature, they will need to have imported our public key beforehand.


45.5 Encrypting data

 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

 *Duration: a few minutes.*

The purpose of this tool is to digitally encrypt data. This can be used, for example, to transmit one or more confidential documents on an unencrypted medium that already contains data, or to put these same documents online. page 249

If you're using an encrypted Debian, you first need to install the *Kleopatra* software package, which contains the key management tool you'll be using. If you're using Tails, this package is already installed. page 119
page 134

 When *Kleopatra* is launched, it may display a warning message entitled *Kleopatra Automatic Test Results*, in which the *scaemon Configuration Check* appears to have failed. This isn't serious, but it can quickly become disturbing. So that this message doesn't appear every time *Kleopatra* starts up, you can uncheck the box *Launch these tests at startup* and then click *Continue*. Another possibility is to install the *scaemon* package, even though it won't be of any use to us. page 135

Getting started:

- Go to *Kleopatra*, by pressing  ( on a Mac) to open the activity overview, then typing `kleo` and clicking on the corresponding software.
- In the toolbar at the top of the window, click on Sign/encrypt....

⁴ If you've already typed in your passphrase shortly before, it won't be requested again. OpenPGP keeps it in memory for ten to thirty minutes.

- Select the file to be encrypted and click *Open*.

You can choose to encrypt the file with one or more public keys, or use a passphrase.

45.5.1 Encrypting data with a passphrase

If you use a passphrase, you'll have to share it with the people who have to decrypt the data.

- Uncheck *Sign as*.
- Uncheck also *Encrypt for me* and *Encrypt for others*.
- Check *Encrypt with password*.
- Click on *Encrypt*.
- Enter the *Secret Phrase* twice, then click *OK*.
- An *Encryption Success* message should be displayed.

The encryption process can take up to several minutes, depending on the size of the file and the power of the computer being used. Once the encryption operation is complete, the encrypted file appears next to the original, unencrypted file, with the `.pgp` extension at the end of its name.

45.5.2 Encrypt data with one or more public keys


If you are encrypting with public keys, you need to have the public keys of *all* the people with whom you wish to share the file in your keychain. If you haven't already done so, you'll need to import them.

- Uncheck *Sign as*, unless you also wish to sign the file digitally. In this case, you'll need to choose the right contextual identity (if you have more than one).
- If you also wish to encrypt the file for your own key, check *Encrypt for me* too.
- Check *Encrypt for others* and select the keys of the people with whom you wish to share the file.
- Click on *Encrypt* (or *Sign/encrypt*).
- An *Encryption Success* (or *Signature and Encryption Success*) message should afficher.

The encryption process can take up to several minutes, depending on the size of the file and the power of the computer being used. Once the encryption operation is complete, the encrypted file appears next to the original, unencrypted file, with the `.pgp` extension at the end of its name.

45.6 Decrypt files


 *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

 *Duration: a few minutes.*


 The purpose of this tool is to decrypt a digitally encrypted file. In particular, it can be used to read confidential documents transmitted in encrypted form.

 If you're using an encrypted Debian, you first need to install the Kleopatra software. If you're using Tails, this package is already installed.





 When Kleopatra is launched, it may display a warning message entitled *Kleopatra Automatic Test Results*, in which the *scaemon Configuration Check* appears to have failed. This isn't serious, but it can quickly become disturbing. So that this message doesn't appear every time Kleopatra starts up, you can uncheck the box *Launch these tests at startup* and then click *Continue*. Another possibility is to install the scaemon package, even though it won't be of any use to us.

page 135

 **Caution:** always move the file to be decrypted to the location where you wish to store it in its decrypted form. For example, if the encrypted file is stored on an unencrypted USB key, it is very important to move it before decrypting it, otherwise the decrypted file will be stored in clear text on the USB key.

To decrypt the :

- Go to *Kleopatra*, by pressing  ( on a Mac) to open the activity overview, then typing *kleo* and clicking on the corresponding software.
- In the toolbar at the top of the window, click on *Decrypt/Verify...*
- Select the file to be decrypted and click *Open*.
- Enter the shared passphrase or the passphrase of our secret key in the dialog box that afficheicht⁵.
- If the file is not only encrypted but also signed, the result of signature verification is displayed in the same way as when verifying a simple signature. Otherwise, a message indicates *Decryption Success*.
- Click on *Save all* to save the decrypted file.

page 346

5. If you've already typed in your passphrase shortly before, it won't be requested again. OpenPGP keeps it in memory for ten to thirty minutes.

Use instant messaging with OTR

C *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

🕒 *Duration: Half an hour to an hour.*

The aim of this tool is to dialogue with a person using instant messaging with encryption and authentication. To achieve this, we'll be using the OTR¹ protocol, which adds encryption, authentication and persistent confidentiality to a number of instant messaging protocols.

page 258

To be able to use OTR to chat with our correspondent, she must also activate OTR in her instant messaging software. To do this, she can also follow the instructions given in this chapter.



46.1 Install the Pidgin instant messaging client

We're going to use the Pidgin e-mail client for this. Pidgin supports OTR encryption. It also supports various instant messaging protocols, such as XMPP² or IRC³ among others.⁴ This software is installed in the live Tails system, but only the XMPP and IRC protocols are supported, the others being difficult to anonymize. On an encrypted Debian, you'll need to start by installing the `pidgin` and `pidgin-otr` packages.

page 119

page 135

46.2 Launch Pidgin

To open the instant messaging software, open the activity overview by pressing  ( on a Mac), then type `pidgin`, and finally click on *Pidgin Internet Messaging*.

1. Wikipedia, 2014, *Off-the-Record Messaging* [[https://fr.wikipedia.org/wiki/Off-the-Record Messaging](https://fr.wikipedia.org/wiki/Off-the-Record_Messaging)].

2. Wikipedia, 2014, *Extensible Messaging and Presence Protocol* [<https://fr.wikipedia.org/wiki/XMPP>].

3. Wikipedia, 2014, *Internet Relay Chat* [https://fr.wikipedia.org/wiki/Internet_Relay_Chat]. IRC normally accepts use without creating an account. Today, most IRC servers refuse connections *via* Tor; with the exception of a few servers, including OFTC [<https://www.oftc.net/Tor/>]. The use of IRC is not explained in this guide.

4. For an exhaustive list of protocols supported by Pidgin, please refer to their [website](https://www.pidgin.im/) [<https://www.pidgin.im/>].

46.3 Setting up an e-mail account

When Pidgin is opened and no e-mail account has been set up, a window appears offering to add a new account.

To set up a new account, click on *the* Add... button.

An *Add Account* window opens. If you already have an instant messaging account, fill in the necessary account information, starting by selecting the *Protocol* you wish to use.

46.4 Create an XMPP instant messaging account

As with an e-mail account, you'll need a login and a passphrase (see page 103). To avoid using the same one over and over again, or running the risk of forgetting it, you can use a password manager (see page 355).

You can also use community servers where registration is free. For example, lists of free XMPP servers are available on the jabberfr.org site.⁵

Once the account has been created on the chosen server and the necessary information⁶ entered in the Pidgin window, check the *Create this new account on server* box.

46.5 Encrypting the server connection

By default, Pidgin configures the new account to encrypt communication with the server.

[page 255] If the certificate is properly signed by a certification authority, the connection will go ahead without a hitch, and Pidgin will save the server certificate in its configuration.

[page 323] If the server's certificate is not signed, or if for some reason Pidgin is unable to verify its authenticity, it is necessary to use the same techniques as when verifying a certificate in your web browser, otherwise adversaries could usurp the server's identity.

[page 254] In this case, on our first connection, Pidgin affichera a window asking if we want to *Accept the certificate for [example.org]*? It will also explain why it didn't want to accept the certificate (*The certificate is self-signed. It cannot be verified automatically*, if for example the certificate is not signed by a certification authority). By clicking on *View certificate...*, Pidgin affichera the digital fingerprint of the certificate, allowing us to verify it.

53

46.6 Activate the OTR (*Off-the-Record*) plugin

[page 251] You now need to activate end-to-end encryption with OTR.

In the Pidgin *Tools* menu, click on *Plugins*. Find the line "Off-the-Record confidential messaging" and check the corresponding box to activate the plugin. Click on *Configure Plugin* to select options such as *Do not archive OTR conversations*.

5. A list of community XMPP servers [https://wiki.jabberfr.org/Serveurs#Serveurs_communautaires]. If account creation fails with one server, don't hesitate to try with another .

6. For more details on how to create an XMPP account, see the [Linuxpedia website](https://www.linuxpedia.fr/doku.php/internet/pidgin-jabber) [<https://www.linuxpedia.fr/doku.php/internet/pidgin-jabber>].

46.7 Set up a private conversation

46.7.1 Add a contact or join a chat room

Depending on our situation, we'll either have to add the contact we want to talk to to Pidgin, or we'll have to join the lounge where we can find him.

Add a contact

To add a contact in Pidgin, click on *Contacts* in the software menu bar and go to *Add a contact....* Then fill in the relevant contact information and click on *Add*.

Our contact will then receive an authorization request to be added to our contact list. Once our contact has accepted the request, we'll be able to start chatting.

Join a chat room

If, on the other hand, you want to join a chat room where the person you want to chat with is likely to be, click on *Contacts* in the software menu bar and go to *Join a chat....* In the same way, fill in the necessary information and click on *Chat*.

Unfortunately, it will not be possible to use end-to-end encryption in Pidgin chat rooms. The OTR protocol does not work for Pidgin chat rooms.

46.7.2 Start a private conversation

To start a private conversation, double-click on a name in the right-hand column of the chat window you're in, or click on your partner's name in the main Pidgin window. A conversation window opens. Click on the *OTR* menu → *Start a private conversation*.

If this is the first time using OTR with this account, Pidgin will then generate a private key and afficher a *Private Key Generation* window. This key is unique for a given account. If you have several instant messaging accounts, you'll therefore have several keys. When it affichs that the generation of this key *is complete*, you can close this window by clicking on *Validate*.

Pidgin affiche then *Ana has not been authenticated yet. You should authenticate this contact*. This means that our conversation is encrypted, but an adversary could impersonate Ana. To be sure of speaking with Ana, you need to authenticate the conversation.

page 254

46.7.3 Authenticate a correspondent

To authenticate a correspondent, it is necessary either to have agreed on a secret beforehand, or to have a means of communication other than instant messaging, which is considered secure. This may be a live conversation, an encrypted e-mail, *etc*.

OTR offers three ways to authenticate a contact:

- by question-answer: we define a question and its answer. The question is then put to our correspondent;
- with a shared secret: a secret known only to the two interlocutors is requested in order to check that we are indeed talking to the person we want to talk to;

- manual fingerprint verification: we check that the fingerprint of the key of the person with whom we are about to have an encrypted conversation is the same as the one provided to us by an *authenticated* means.

Once the secrets, Q&A or fingerprints have been exchanged, click on *OTR* → *Authenticate contact*. Choose the authentication method below *How would you like to authenticate your contact*, then answer the questions. Finally, click on *Authenticate*.

If authentication is successful, the conversation status changes to *Private*, meaning that it is not only encrypted, but also authenticated.

[page 116]

If you are using a non-live system or have activated Pidgin persistence in Tails, this authentication step only needs to be carried out once for a given contact.

46.7.4 End a conversation

Once our dialog is complete, click on *OTR* → *End private conversation*. This deletes the temporary encryption key generated for this conversation from the computer's RAM. Even if adversaries were to obtain our private keys, they would not have access to the key enabling them to decrypt the conversation *a posteriori*.

Managing passwords

C *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

🕒 *Duration: Fifteen to thirty minutes.*

When you create an e-mail address, an account on a website, *etc.*, this account is usually protected by a password.

It's important not to use the same password for different accounts or different purposes.

It's also important not to use the same password for different `context identities`, so that compromising one doesn't compromise the others. page 243

There are two good schools for password management:

- Choose and remember a different passphrase for each use;
- randomly generate passwords and store them in a *password manager*, which in turn is protected by a passphrase.

47.1 Choosing a good passphrase

The first school has the advantage of requiring no storage media: you always have your passphrases with you. To apply it, consult the right passphrase (see page 103).

However, when you multiply accounts and contextual identities, it can be a lot of passphrases to remember.



47.2 Use a password manager

The second method can then be useful. In practice, we'll have one passphrase to remember per identity, with our password manager then taking care of storing the various passwords linked to this identity. This can be done on an encrypted Debian system or on an *amnesiac live system* using persistence. page 119

47.2.1 Install the password manager

We're going to use the KeePassXC password manager. If it is not installed on our system, install the *KeePassXC* software (see page 134). KeePassXC is installed by default in Tails. page 113
page 116

47.2.2 Launch KeePassXC

Press  ( on a Mac) to open the activity overview, then type `keepassxc` and click on the *KeePassXC* icon.

47.2.3 Create and save a password database

A password database is a set of passwords stored in the same KeePassXC database and encrypted with the associated passphrase.

If you choose to use KeePassXC in Tails, you must first activate persistence (see page 116) and enable the *Personal data* option.

When KeePassXC is launched, you must first create a new password database and save it for future use. To create a new password database, select *Create new database*.

To store the newly created password database for future use, enter a name and optionally a description. Then click on *Continue*.

You can then set the *Database Encryption parameters*, which can also be modified at a later date. Click on *Continue*.

Next, we need to choose a passphrase that will be used to decrypt the password database. Since this database will contain some of our passwords, it's important to choose a good passphrase (see page 103). Specify this passphrase twice in the *Password* text box.



TO FIND OUT MORE...

We can also decide to *Add extra protection* by generating or indicating a *Key File* comprising random bytes; however, we must keep this file secret and not lose it: without the Key File, it will be impossible to access our database.

The advantage of such protection is that we can store this key file on a medium other than our password database, such as an encrypted USB key that we keep in a safe place. In addition to our passphrase, we'll need to have this USB key with us to be able to access the passwords contained in the database; and someone who manages to guess our passphrase won't be able to decrypt the database without our key file.

On the other hand, the risk with this technique is that of misplacing the USB key containing the key file: if we hadn't made a backup copy of our key file, we'd be completely unable to access our passwords.



Click on *Finish* when finished.

Next, you need to tell KeePassXC where to save this database. If you're using Tails, the default location is the *Persistent* folder: leave this as it is, so that the database is saved in the Tails persistent volume. Otherwise, choose the desired location to save the database. Click on *Save*.

Since it will contain some of our passwords, remember to regularly back up a copy (see page 151) of this database.


47.2.4 Generate and save a random password

KeePassXC also lets you generate random passwords that are more robust than passwords you can remember.

In KeePassXC, click on *Entries* then  *New entry...* Fill in the relevant fields. For the *Password* field, click on the dice-shaped button () located to the right of the input field.

A window containing various password generation options opens.

Among the options available, it's best to use lowercase letters, ma- juscules and numbers, then increase the number of characters in the password (to at least 32), since you won't have to remember it. Special characters can sometimes cause problems with certain software programs or websites.

To select the desired characters, click on the corresponding buttons in the *Character types* section. When a button is highlighted (in green), the password will be generated with this type of character; when the button is deselected (i.e. when it appears in light grey), the corresponding characters will not be used in the generated password. Clicking on the crossed-out eye  to the right of the generated password makes the password visible, enabling you to check what has been generated.

The *Entropy* indicator measures the robustness of the generated password. It is directly linked to the type of characters selected and the length of the password. The recommended minimum entropy is 128 bits.

Click on *Confirm password*, then on *OK*.

47.2.5 Restore and unlock the password database

If you want to use a previously registered password database, you need to unlock it. To do this, launch KeePassXC. In general, if it finds it, KeePassXC automatically suggests opening the last password database used. It then indicates *Unlock KeePassXC database*, followed by the name of the corresponding file. If this is the database you wish to open, you can skip the next paragraph.

If this is not the case, or if *KeePassXC* does not automatically suggest a database to unlock, go to the *Database* menu, click on *Open a database...*, then browse through the folder list to find the *.kdbx* file corresponding to the database you wish to open. Select this file and click on *Open*.

Whether KeePassXC has automatically found the database you wish to open, or you have specified it yourself, it will ask you to unlock it. To do this, in the *Enter password* field, enter the passphrase we had configured when we created the database. If you have defined additional identifiers to protect the database (such as a key file, for example), these should also be entered here. Finally, click on *OK*.

If the passphrase is incorrect, the following error message appears:



Database read error : Invalid identifiers have been supplied, please try again.
If the problem recurs, the database file could be corrupted.

47.2.6 Use a registered password

Once the password database has been restored and unlocked, the passwords stored in it can be used.





There are two ways of using a registered password: manually, by copying and pasting the user name and password, or using automatic entry.

Automatic input

KeePassXC can record "window associations", which link an entry with the name of a window and an autocomplete sequence, i.e. the entry information to be typed directly into this window.

To do this, you need to open the window in which you wish to perform automatic entry, such as the web browser with the mailbox login page.

Next, search KeePassXC for the entry you wish to use for this window, then double-click on it to modify it. The *User Name* and *Password* fields must be filled in. In the left-hand column, go to *Autocomplete*. Make sure that the *Enable auto-completion for this entry* box is ticked. Click on the **+** symbol at the bottom of the *Window Associations* section to create a new association. You can then choose the window to which you wish to associate the entry from the *Window title* drop-down menu on the right. Finally, click on *OK*: the settings are complete.





From now on, you can use the automatic entry sequence by positioning the focus ¹ in the window where you wish to enter your login details, e.g. the e-mail field in your mailbox browser. Then switch to KeePassXC on the corresponding entry and start auto-completion. This is done with the combination of keys    ² or by clicking on the keyboard icon  in the top toolbar.








Please note: autofill can also be used to make some very tricky mistakes, such as pasting your password into an instant messaging window... and sending the message automatically. So be very careful where you place the cursor before executing autofill.

This automatic input method may not work for all interface types. In this case, you'll need to switch to manual input.

Manual input

In KeePassXC, to retrieve the username from the clipboard, go to the entry you wish to use, then right-click and choose *Copy username* or key combination  . Then paste the contents of the paper in the window field where to enter the login. Proceed in the same way to copy the password by right-clicking on the entry and choosing *Copy password*, or use password entry  +  and then stick it in the field.

1. The place where the next characters will appear when typing.

2.  is the notation for the shift key. This key can be found under various notations depending on the keyboard:   , .

Using OnionShare

C *As software evolves, we strongly advise you to use the most up-to-date version of this tool, which is available on the <https://guide.boum.org/> website.*

⌚ *Duration: Five to ten minutes.*

To make one or more files available to others, you can host them on a web server.

page 319

However, there's no *a priori* reason to trust the people or administrations who run these servers.

If you prefer not to rely on a third party, you can host the documents you want to share yourself, and do so *via* an onion service.

page 266

One of the advantages of this is that it strongly protects the location of the hosting server, which in this case is our own computer. However, this anonymization system is not infallible.

page 267

To do this, we'll be using OnionShare software, which in just a few clicks lets you create an Onion service and host the files of your choice on it.


48.1 Using OnionShare in Tails

OnionShare is installed by default in Tails. You can follow the official Tails documentation, which is available from any Tails support, even without an Internet connection.

Start Tails first. On the desktop, double-click on the *Tails Documentation* icon. In the index that opens, look for the *Uncensored and anonymous Internet* section and click on the *Share files with OnionShare* page under *Internet Applications*. This is the page to follow.

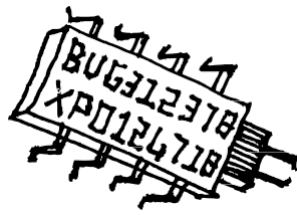
page 115

48.2 Using OnionShare in Debian

Start by installing the *OnionShare* software if you haven't already done so. To launch it, open the Activities overview by pressing  ( on a Mac), then type `onion` and click on *OnionShare*.

page 131

OnionShare will connect *via* Tor. You can let the software guide you by choosing the file you want to share.



Who's speaking?

Unfortunately, we don't have a simple answer to this question, but we'd like to say a few words about it.

First of all, there are several reasons why we believe it's important to publish a book anonymously. One of them, which we developed in the preface, is that when asked "Nothing to hide?", we answer in unison "Yes! Anonymity is therefore first and foremost a way of protecting ourselves. What's more, we choose not to put ourselves forward individually, to keep the *who out* of the limelight and the *what in* the spotlight.

Secondly, since the first issues of this *guide*, the number of people who have participated, from near or far, in its writing, correction and editing, makes the collective that brings this project to life wide-ranging, evolving and not clearly defined.

Finally, we feel that we have left sufficient traces in these pages to enable anyone reading this to situate us, at least partially, in our relationship with IT, whether technical, political or ethical.

*
* *

Two features of this work, however, force us to face up to questions about its provenance from certain angles. On the one hand, this work claims to transmit technical knowledge and know-how, usually reserved for specialists. On the other hand, the accuracy of the information provided may have implications for the peace of mind of those who use it. Small errors that we may have overlooked can have serious consequences.

So it's important to say a few words about the people who contributed to this guide. Clarifying the extent of our knowledge and know-how - and their limits - helps us to find a more appropriate learning relationship with this document, but also to decide on the level of *technical* confidence it deserves. So, let's say that, within the :

- the issues raised by this guide have been important to us, both technically and politically, for over a decade;
- we run transmission workshops and advise people in need of digital privacy;
- we're pretty familiar with the workings of some operating systems, especially Debian GNU/Linux ;
- we have a good grounding in cryptography, but we're a long way from being able to claim to have mastered the subject.

And finally, affirm one last time that the word carried by this book, like any word of *guide*, must be taken with tweezers commensurate with the consequences at stake.

Index

- CA, *see* certification authority
- administrators, *see* admins
- admins, **206**
- adminsys, *see* admins
- address
 - .onion address, *see* onion service
 - IP address, **202, 205, 217**
 - MAC address, *see* hardware address
 - physical address, **198, 215**
 - private address, **205**
 - public address, **205**
- ADSL, **199**
- algorithm, **48, 333**
- AMD Platform Security Processor, *see* Intel Management Engine
- AMD PSP, *see* Intel Management Engine
- anonymity, **243**
- anonymity set, **268**
- application, **22**
- architecture, **17**
- archiving, **89**
- argument, **98**
- ARPANET, **197**
- AS, *see* Autonomous Authenticity System, **53**
- self-hosting, **242, 319**
- certification authority, **255, 323**
- backbone, *see* backbone
- backdoor, *see* backdoor
- library, **23**
- binary, **17**
- BIOS, **20, 76, 107, 126**
- bit, **17**
- BitTorrent, **200**
- boot, *see* Internet
- box startup, **205, 215**
- black box, **269**
- bridge, *see* switch
- bridge Tor, **267, 268, 314**
- bug, **29**
- cache, **43, 213**
- motherboard, **16**
- network card, **198**
- CD or DVD, **171**
- administrative censorship, **233, 234**
- electronic certificate, **255, 323**
- file path, **98, 142** Trojan
- horse, **32** encryption, **47, 47, 249, 347**
 - end-to-end encryption, **251, 333, 343, 352**
 - repudiable encryption, **52** encrypting a hard disk, **145** encrypting a system, **119** encrypting a USB key, **145**
- chipset, **20**
- virtual keyboard, *see* visual keyboard
- visual keyboard, **327**
- mail client, *see* mail client
- mail client, **292, 333**
- encryption key, **48, 50, 249, 333, 343**
- Internal Security Code, **32, 224, 229, 237**
- Code of Criminal Procedure, **31, 52**
- Penal Code, **51**
- source code, **39**
- cold boot attack, **27, 50, 75**
- collision, **53** switch, *see* switch
- switch confidentiality, **47**
- cookie, **214, 222**
- CPU, *see* processor
- cryptanalysis, **47**
- cryptography, **47**
 - asymmetric cryptography, **55, 249, 333**
 - symmetrical cryptography, **55**
- cryptology, **51**
- Debian, **22, 119**

- deep packet inspection, *see* deep packet inspection
- Déjà Dup, **153**
- start-up, **107**
- package depot, **136**
- dereferencing, **234**
- DHCP, **204**
- hard disk, **18, 42**
- SSD disk, *see* flash memory, *see also* hard disk
- distribution, **23, 40**
- DNS, **210, 232**
- top-level domain, **232**
- IPR, *see* in-depth examination of packets
- plausible deniability, *see* reputable encryption
- package depot, **23**
- overwriting data, **42**
- deletion, **42**
- electricity, **21**
- imprint, *see* checksum header, **202, 217**
- encapsulation, **201**
- keystroke recorder, **35**
- backbone, **208**
- exchange space, *see* virtual memory
- Ethernet, **199**
- in-depth examination of packages, **217, 230, 236**
- ext2, ext3 or ext4, **24**
- Facebook, **222**
- ISP, *see* Internet Service Provider
- FAT32, **24**
- fiber optics, **199**
- filtering, *see* phishing filtering, **236**
- firewall, *see* firewall
- firmware, *see* firmware hash
- function, **53, 161**
- brute force, **77**
- file format, **24**
- formatting, **24, 44, 146**
- Internet service provider, **205, 224**
- GAFAM, **31, 224**
- Virtual machine manager, **84, 163**
- password manager, **355**
- Synaptic package manager, **23, 135**
- GNU/Linux, **22, 40**
- GnuPG, **49**
- Google, **221**
- hash, *see* hash function phishing, **234**
- hibernation, **28**
- historical, **29**
- man in the middle, *see* monster in the middle
- HTTP, **200, 217, 291**
- HTTPS, **200, 255, 291**
- accommodation, **211, 233, 242, 285, 319, 359**
- contextual identity, **243**
- disk image, **114, 169**
- ISO image, **114, 123, 169**
- IMAP, **200, 292, 331**
- IMAPS, **200, 255, 292**
- printer, **36**
- installer, **119**
- Intel Management Engine, **20, 33**
- Intel ME, *see* Intel Management Engine
- Internet, **205, 209, 239**
- Internet Protocol, **202**
- interoperability, **199**
- integrity, **53**
- IP, *see* Internet Protocol
- IPv4, *see* Internet Protocol
- IPv6, *see* Internet Protocol
- IRC, **200, 351**
- Java, **214, 241**
- JavaScript, **214, 241**
- logging, **43**
- newspapers, **29, 215, 216, 218, 224**
- KeePassXC, **355**
- keylogger, *see* keylogger
- LAN, *see* local area network library, *see* library license
- library license
 - free license, **40, 132**
 - proprietary license, **39**
- command line, **97**
- authorized list, **66**
- blocked list, **66**
- log, *see* software
- logs, **22, 131**
 - software installation, **131**
 - spyware, **32**
 - free software, **39, 40, 132**
 - malware, **31, 32, 76**
 - open source software, **40**
 - portable software, **44**
 - proprietary software, **39, 39**
- laws, **31**
 - law for confidence in the digital economy, **225**

- Intelligence Act, **32** Act to strengthen the fight against terrorism
 - organized crime, terrorism [...], **31**
 - loi renforçant les dispositions relatives à la lutte contre le terrorisme, **52**
- LOPPSI2, **231**
- requête légale, *see* réquisition judiciaire
- LUKS, **50, 145**
- MAC, *see* hardware address
- middle machine, *see* middle monster
- machine-in-the-middle, *see* monster in the middle
- malware, *see* malicious software
- man-in-the-middle, *see* monster in the middle
- MAT2, **185**
- memory
 - flash memory, **18, 20, 42, 140**
 - read-only memory, *see* permanente
 - sistante
 - persistent memory, **18**
 - virtual memory, **25, 28, 44, 73**
 - living memory, **18, 27**
- instant messaging, **351**
- metadata, **30, 218, 221** microcode, *see* firmware
 - firmware, *see* firmware
 - firmware, **20, 76, 107, 120** update, **175**
- modem, **199, 205**
- modem-router, *see* Internet box
- threat model, **63**
- monster-in-the-middle, *see* monster in the middle
- monster in the middle, **254**
- password, **41, 355**
- NAT, **205**
- net neutrality, **207**
- domain name, **210, 232**
- core, **22**
- NTFS, **24**
- digitization, **17**
- Office central de lutte contre la criminalité liée aux technologies de l'information et de la communication, **231, 233**
- onion, **261, 313**
 - .onion address, *see* OnionShare service, **359**
 - onion service, **266, 326, 331, 359**
- waves, **21**
- open source, **40**
- OpenPGP, **333, 343**
- option, **98**
- OS, *see* operating system OTR, **351**
- Outlook, *see* mail client
- key pair, *see* encryption key
- package (software), **23, 155**
- packet (network), **202, 217**
- firewall, **203**
- score, **23**
- peering, **207**
- peripheral, **20**
- phishing, *see* phishing
- passphrase, **47, 103**
- Pidgin, **351**
- pilot, **22**
- access point, **204**
- security policy, **65**
- bridge, *see* switch
- Tor bridge, *see* bridge
- Tor POP, **200, 292, 331**
- POPS, **200, 255, 292**
- port, **203**
- captive portal, **216**
- back door, **39, 216**
- forriél, **31, 32, 296**
- processor, **16**
- program, **22**
- protocol
 - application protocol, **200, 217**
 - communication protocol, **199** IP protocol, *see* Internet Protocol collar
 - network protocol, **202, 217**
- pseudonymity, **243**
- radio, *see* Wi-Fi
- RAM, *see* RAM
- judicial requisition, **51, 228**
- requête légale, *see* réquisition judiciaire
- local network, **204**
- risks
 - risk assessment, **63**
 - risk reduction, **61**
- RJ-45, *see* Robot Ethernet, **296**
- rootkit, **32**
- routing, **208**
- router, **205, 207, 208, 217**
- VPN, *see* VPN
- Virtual Private Network, *see* VPN
- VPN data retention, **224**
- backup, **151**

- automatic backups, **29**
- secure-delete, **139**
- shred, **142**
- Signal, **200**
- digital signature, **55, 345**
- steganographic signature, **36**
- mirror site, **231**
- Skype, **201**
- SMTP, **200, 291, 331**
- SMTPS, **200, 255, 291**
- checksum, **53, 161** spam,
see spam spyware, *see*
spyware
- SSD, *see* flash memory, *see also*
SSL hard
disk, *see* TLS
- local web storage, **214**
- steganography, **36**
- attack surface, **66**
- swap, **25, 28**
- switch, **204**
- Synaptic, *see* Synaptic package
manager
- syntax, **98**
- stand-alone system, **206**
- file system, **24, 43**
- operating system, **22** system
installation, **119** host system,
84
guest system, **84**
live system, **22, 44, 82, 113, 113**
- Tails, **44, 82, 113, 175, 267, 280, 295,**
301, 327, 343, 359
persistent storage, **151, 356**
- TCP, **202**
- terminal, **97**
- Thunderbird, *see* mail client
- TLD, *see* top-level domain TLS, **255,**
258
- top level domain, *see* top level
domain
- Tor, **261, 313**
- traces, **27, 213**
- transistor, **17**
- transit, **207, 209**
- TrueCrypt, **40**
- UDP, **202**
- UEFI, **20, 76, 126, 128**
- upgrade, *see* USB update,
20
- eve, **28**
- VeraCrypt, **52**
- Virtual Private Network, *see* VPN
- virtualization, **83, 163**
- virtual machine, **212**
- hardware virtualization, **164**
- virus, **32**
- VPN, **227**
- watermarking, *see* stegano-graphic signature
- webmail, **291**
- WebRTC, **215**
- Wi-Fi, **199, 204**
- Windows, **82, 165**
- wipe, *see* data overwriting
- XMPP, **200, 351, 352**

Credits

Cover, back cover and drawings on pages [i](#), [xvi](#), [8](#), [188](#) and [360](#) by the Digital Self-Defense Guide team.

Photo page [16](#) by Darkone, license CC BY-SA 2.5, found on:

https://secure.wikimedia.org/wikipedia/fr/wiki/Fichier:ASRock_K7VT4A_Pro_Mainboard.jpg.

Photo page [16](#), public domain, found on:

<https://secure.wikimedia.org/wikipedia/fr/wiki/Fichier:Pentium-60-back.jpg>.

Photo page [18](#), by Topory, CC BY-SA 3.0 license, found on:

https://fr.wikipedia.org/wiki/M%C3%A9moire_vive#/media/Fichier:RAM_n.jpg.

Photo page [18](#), public domain, found at:

<https://secure.wikimedia.org/wikipedia/fr/wiki/Fichier:Hdd-wscsi.jpg>.

Photo page [19](#), public domain, found on:

https://commons.wikimedia.org/wiki/File:MSATA_SSD_16_GB_Sandisk_-_SDSA3DD-016G-2494.jpg.

Photo page [20](#) by Zac Luzader Codeczero, CC BY 3.0 license, found on:

https://secure.wikimedia.org/wikipedia/fr/wiki/Fichier:AT_Motherboard_RTC_and_BIOS.jpg.

Drawing on page [77](#) of XKCD, CC BY-NC 2.5 license, found at:

<https://xkcd.com/538/>.

Photo page [199](#) by David Monniaux, CC BY-SA 3.0 license, found on:

https://commons.wikimedia.org/wiki/File:Ethernet_RJ45_connector_p1160054.jpg.

Photo page [207](#) by Geek2003, CC BY-SA 3.0 license, found on:

https://commons.wikimedia.org/wiki/File:Avaya_Secure_Router_2330.jpg.

Photo page [211](#) by Victor Grigas, CC BY-SA 3.0 license, found on:

https://commons.wikimedia.org/wiki/File:Wikimedia_Foundation_Servers-8055_08.jpg.

Diagram page [263](#) by HANTwister, CC BY-SA 3.0 license, found at:

https://en.wikipedia.org/wiki/Onion_routing#/media/File:Onion_diagram.svg.

Diagrams on pages [264](#), [265](#), [265](#) and [266](#) from Nos Oignons and Electronic Frontier Foundation, CC BY license, found on :

<https://nos-oignons.net/Diffusez/index.fr.html>.

Body text icons based on Font Awesome 4 by Dave Gandy, SIL OFL 1.1 license

(<https://fontawesome.com/v4/>).

To go further" icon by Mr Minuvi of The Noun Project; "window content" icon by Gregor Cresnar of The Noun Project; "law text" icon by Handicon of The Noun Project; "details" icon by Colourcreatype of The Noun Project: CC BY 3.0 license (<https://thenounproject.com/>).

The other diagrams are made by the guide team and use icons: from GNOME Project, CC BY-SA 3.0 license; from Silvestre Herrera, GPLv2 license, found at <http://>

www.silvestre.com.ar/; from the public domain found at <https://openclipart.org>.