Cybersecurity

Current challenges and Inria's research directions

Inria | WHITE BOOK | N°03

Inria white books look at major current challenges in informatics and mathematics and show actions conducted by our project-teams to these challenges. Their goal is to describe the state-of-the-art of a given topic, showing its complexity, and to present existing, as well as emerging, research directions and their expected societal impact.

This white book has been edited by Steve Kremer, Ludovic Mé, Didier Rémy and Vincent Roca. They coordinated the contributions from researchers of Inria teams (the complete list of contributors is given at the end of the book). Many thanks to Janet Bertot for proof-reading this document, as well as to François Pottier, Gabriel Scherrer, and Benjamin Smith who read parts of it.

Publication date: January 2019

Executive summary

Wikipedia defines cybersecurity as "the protection of computer systems from theft and damage to their hardware, software or information, as well as from disruption or misdirection of the services they provide." More precisely, cybersecurity consists in ensuring three properties of information, services, and IT infrastructure: *confidentiality, integrity,* and *availability.* Thus, securing an information system means preventing an unauthorized entity from accessing, altering, or making unavailable computer data, computing services, or the computing infrastructure. Another increasingly important property is privacy, which may be seen as the confidentiality of the link between persons and data. Note that the terms security and safety are sometimes misused. While safety refers to accidental threats, security refers to intentional threats. Security and safety remain quite different and well-identified domains that build on different hypotheses, and the protection mechanisms against accidental and intentional threats are usually complementary. In this white book, we restrict our attention to security.

The digitalization of our society is radically changing the manner in which computer systems are used. A huge proportion of the population is continuously connected to the Internet, using an amazing number of different services. Simultaneously, we are permanently exposed to attacks: our sensitive data may be stolen, modified, or destroyed. We also live with the risk of mistakenly and irreversibly leaking our private information on the Internet. Companies, states and their critical infrastructures, which nowadays are interconnected, are also vulnerable. The economical and societal damage of successful cyberattacks may be considerable. Cybersecurity has thus become a general concern for all, citizens, professionals, politicians, and, more generally, all decision makers.

This book provides an overview of research areas in cybersecurity, illustrated by contributions from Inria teams. The first step in cybersecurity is to identify *threats* and define a corresponding *attacker model*. Threats, including malware, physical damage or social engineering, can target the hardware, the network, the operating system, the applications, or the users themselves. Then, detection and protection mechanisms must be designed to defend against these threats. One of the core mechanisms is cryptography: *cryptographic primitives* can ensure the confidentiality and integrity of data. These primitives must be the object of continuous *cryptanalysis* to ensure the highest level of security. However, secure cryptographic primitives are not sufficient to guarantee secure communications and services: this task requires the use of so-called *cryptographic protocols,* implementing richer interactions on top of the primitives. Cryptographic protocols are

distributed systems: ensuring that they achieve their goals, even in the presence of an active adversary, requires the use of formal verification techniques, which have been extremely successful in this field.

While cryptographic primitives and protocols are fundamental building blocks for security, additional *security services,* such as authentication and access control, are needed to enforce a security policy. These security services, usually provided by the operating system or the network devices, can themselves be attacked and sometimes bypassed. Therefore, activities on the information system are monitored in order to detect any violation of the security policy. Finally, as attacks can spread extremely fast, the system must react automatically or at least reconfigure itself to avoid propagating attacks.

As noted above, privacy has become an intrinsic part of cybersecurity. However, even though it often relies on cryptographic primitives and protocols, it also has its own properties, techniques, and methodology. Moreover, the study of privacy often requires to take legal, economical, and sociological aspects into account.

All these security mechanisms need to be carefully integrated in security-critical applications. These applications include traditional safety-critical applications that are becoming increasingly connected and therefore more vulnerable to security attacks, as well as new infrastructures running in the cloud or connected to a multitude of Things (IoT).

Despite recent significant advances in several areas of cybersecurity, there are still major scientific questions left open. Here are a few selected challenges where Inria could make new, major contributions:

→ *Post-quantum cryptography.* Building a quantum computer is widely believed feasible in the next decades and most cryptography used today could be efficiently broken by such a computer. Therefore, it is important to think now about quantum resistant cryptography, as information encrypted today may still be sensitive when quantum computers appear.

→ *Computing on encrypted data.* The need for computing on encrypted data has emerged with the appearance of the cloud and outsourced computation. This problem can be solved using techniques called homomorphic encryption and functional encryption. In 2009 a theoretical breakthrough was achieved with the first fully homomorphic encryption scheme, but this scheme remained completely impractical due to its poor computational efficiency. A lot of progress has been made since, but more research is needed; any significant technical advance may quickly be exploited as an economical advantage.

→ *End-to-end formally verified cryptographic protocols*

The security of cryptographic protocols is extremely difficult to ensure, and the use of rigorous, formal methods is a necessity. Computer-aided security proofs must include all aspects from the specification down to the implementation. Recent works, in particular around TLS 1.3, have shown that this approach is now feasible. However, this still requires a careful co-design of proof and code that can only be performed by experts. Leveraging such proofs to more general code and more complex security properties, e.g., properties guaranteeing the privacy of the user, remains a huge challenge.

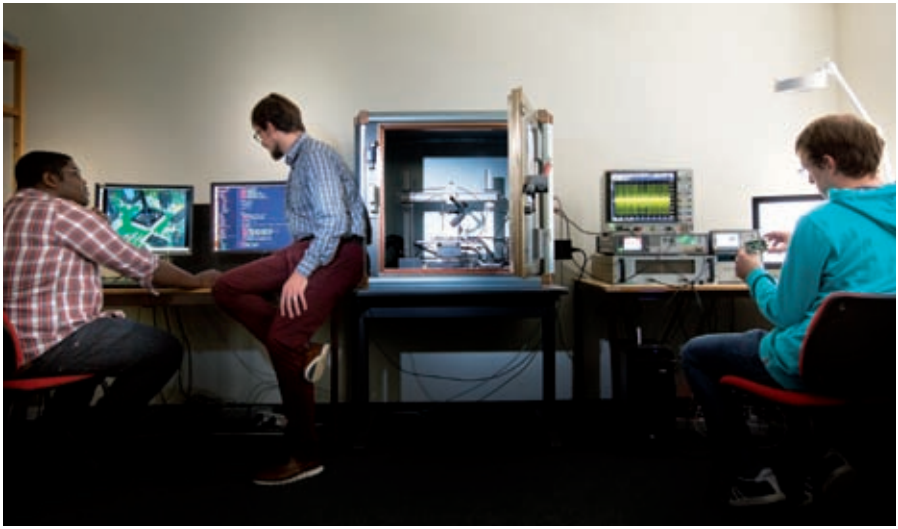
→ *Security for IoT.* for IoT is a major challenge. Attacks are still relatively easy (many devices have not been designed with security in mind), invasive (e.g., in our lives), and have major potential impact because of the huge number of available devices, which increases the attack surface and makes distributed denial of service (DDoS) attacks much easier. The research directions are numerous: let us mention for instance the desire to securely update the software embedded in a device, the need for lightweight cryptographic primitives adapted to limited resources, the analysis of the security of new low-power wide-area wireless technologies, the detection and mitigation of intrusions or misbehaving devices, or the need for secure-by-design frameworks and protocols to facilitate the development of IoT devices.

→ *Protecting citizens' privacy.* Our connected world experiences an unprecedented growth in terms of personal, increasingly intrusive data collection, be it while surfing the web, using a smartphone, or driving a connected car. The lack of transparency, as many services and devices behave as black boxes, and the lack of user control are major issues. How to express consent or opposition in the absence of information or user interface? Identification of such hidden behaviors, which requires data flow analyses, is hindered by the number, complexity, and diversity of underlying applications and communication technologies. Challenging transverse research activities are required to bring transparency, highlight good and bad practices, and enable regulators to enforce data protection laws.

It is essential to include security from the start in system design. The same holds for *cyber-resilience:* massive cyber-attacks are an increasing threat and the security-by-design principle should also apply to resilience of networks and critical, digital infrastructures.

Cybersecurity often lacks incentive, as its benefits are hard to grasp. This is often due to a lack of expertise. Indeed, academics, who have sharp expertise in most areas of cybersecurity, are often underrepresented in national or industrial advisory committees. Education is therefore essential to security, and major

dissemination efforts should be made for all audiences: from teachers, researchers, industrial actors, and specialists, to everyday citizens, including children.



High security lab (LHS) at the Inria Research Center in Rennes – Bretagne Atlantique
© Inria / Photo C. Morel

About this white book

Our goals

The primary goal of this Inria white book on cybersecurity is to detail Inria's view on cybersecurity challenges. To this end, we include a general overview of academic research topics in cybersecurity and, in particular, a cartography of existing research in cybersecurity at Inria. We also take this opportunity to draw general recommendations in the domain of cybersecurity. We have chosen to conduct all these subgoals in parallel in a single, unified document, without a constant distinction between the objectives. Therefore, while we aim for a complete coverage of research in cybersecurity, the level of detail is intentionally non-uniform: domains where Inria has a strong position or, on the opposite end, where Inria's contribution should be increased are emphasized; less structured fields are also given a more comprehensive presentation.

This white book is written to address a wide audience, and to allow different levels of reading. It includes technical presentations of the different cybersecurity domains and a detailed description of the work done in Inria teams which would interest cybersecurity experts or someone looking for detailed information on a particular subdomain. It also includes more accessible information typeset in text boxes: the sections covering technical material include executive **[Summary]**, as well as a list of **[Inria teams]** working in each area with a short summary of their related activities. Other additional information includes **[Highlights]**, pedagogical **[Note]** and **[Research challenge]** boxes.

Therefore, the non-expert reader may first focus on the text boxes, only digging into the full text when necessary. Challenges are described in situ, but they are also collected in a dedicated chapter (§8.1) for convenience.

The methodology

This white book is a collaborative effort, with contributions from many people from Inria and its partners. The writing of this book was coordinated by a working group that consisted of Steve Kremer, Ludovic Mé, Didier Rémy, and Vincent Roca. Given the breadth of the topic, they relied on the help of many other researchers who provided input on their topic of expertise to the working group—a complete list of contributors is given at the end of the book. In a second step, the chapters have been proof-read by researchers working in the respective areas. Citations to scientific papers are intentionally restricted to seminal work, rather than trying to give an extensive bibliography. This white book has been conducted under the supervision of the "*Cellule de veille et prospective*"—the Inria scientific and prospective watch unit—and belongs to a series of other white books[1].

1. https://www.inria.fr/institut/strategie

Outline

The remainder of this white book gives an overview of research areas in cybersecurity and, in particular, the activities of Inria teams. There are certainly many ways to present activities in cybersecurity, sorting them by methodologies, sub-communities, application domains, etc. In this book we follow a winding path, which seemed to be a good compromise for presenting Inria's activities in cybersecurity.

Here is an overview of our journey in cybersecurity:

Chapter 1: INTRODUCTION

We set the scene: we define the scope of cybersecurity, discuss the issues and stakes, giving examples of attacks and their consequences. We also discuss key security properties, as well as some legal (e.g. cybersecurity regulation) and sovereignty considerations.

Chapter 2: THREATS

Works on cybersecurity often start by defining the "attack model": that is, the capabilities of an attacker. In this chapter we discuss different threats that can target the hardware, network, operating system, applications, or even users themselves. We also review some research efforts whose goal is a better understanding of these threats.

Chapter 3: CRYPTOGRAPHY

Cryptography plays an essential role and constitutes the basis of cybersecurity. In this chapter, we cover all aspects of cryptography, ranging from the design of the core primitives to more complex protocols that provide high-level guarantees regarding the security of communications and transactions.

Chapter 4: SECURITY SERVICES

Additional security services are needed in order to design operational systems, even if they are often built upon cryptographic primitives and protocols. These are addressed in this chapter where we present security mechanisms that can prevent or mitigate threats and attacks on the information systems and their components, including hardware, networks, and operating systems.

Chapter 5: PRIVACY

Nowadays, privacy is considered an intrinsic part of cybersecurity. While privacy often relies on cryptographic primitives and protocols, it also has its own properties, techniques, and methodology. In this chapter we focus on privacy, covering technical aspects as well as legal, economical, and sociological issues.

Chapter 6: SECURITY-SENSITIVE APPLICATIONS

While the previous chapters focused on specific services and tools, here we take the opposite approach, looking at a selected set of security-sensitive applications and discussing the specific security questions they raise.

Chapter 7: CYBERSECURITY IN FRANCE

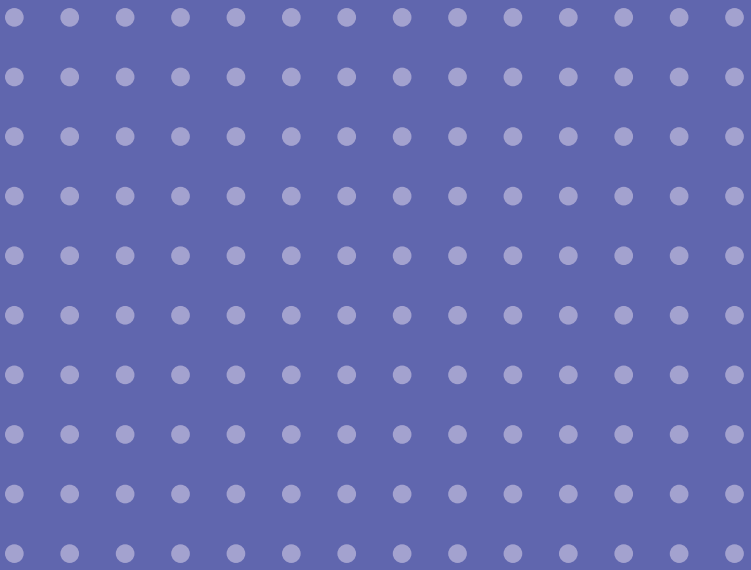
Finally, in Chapter 7, we give an overview of cybersecurity activities at Inria and their positioning in France.

We end the book with general takeaway recommendations.

Contents

1

Introduction



© Inria / illustration Clod

1.1 Cybersecurity, a central concern

A serious concern for our societies

While the word *cybersecurity* was hardly known to the general public two decades ago, it has become a recurrent topic in public media punctuated by frequent cybersecurity attacks, the discovery of new breaches, or the revelation of general surveillance by large companies or state bodies.

The world has changed, and quickly. Cybersecurity has become a general concern for all: citizens, professionals, politicians, and, more generally, all decision makers. It has also become a serious concern for our societies that must protect us against cybersecurity attacks with both preventive and reactive measures, which implies a lot of monitoring, and must simultaneously preserve our freedom and avoid general surveillance.

Cyberattacks may be conducted by criminals, but also by states for industrial espionage, for economic damage to apply pressure, or to inflict real damage to infrastructure as an act of war.

States and their interconnected critical infrastructures are vulnerable. Cyberattacks also put companies—of all sizes—at high risk. The economic damage caused by successful cyberattacks may be considerable. However, our protection level is still considered largely insufficient compared to the risks and potential damages.

While our awareness is improving and protective measures are increasing, they still do so at a slow pace. This is partly due to a lack of incentive: cybersecurity is an investment whose benefits are often hard to grasp, as it only pays off when an

attack that could have otherwise succeeded fails, and this is difficult to measure. This slow progress is also due to a lack of expertise—at all levels.

Cybersecurity is therefore an economic and sovereignty issue for many states, including France.

A challenge for the digital era

The digitalization of our society is completely changing the usage of computer systems. Until the early 90s, computer systems were only loosely connected. Few people had personal computers at home, and those were rarely connected to the Internet. Email spam hardly existed. Cybersecurity was mostly a concern for states and large companies including the financial industry.

Since the end of the 90s, the situation has completely changed. The growth of the web, and the appearance of ADSL and smartphones quickly brought cheap and fast Internet connections to almost every home in developed countries and revolutionized the role of the Internet. Smartphones and 3G/4G have also spread Internet access into developing countries. A huge proportion of the population is now continuously connected to the Internet, using an amazing number of different services. Simultaneously, we have become permanently exposed to attacks, with our sensitive data at risk of being stolen or damaged. We also live with the risk of mistakenly and irreversibly leaking our private information on the Internet. Cybersecurity is now a serious issue for everyone: citizens, small and large companies, administrative and state bodies. For example, a successful ransomware attack on a small company that does not have (well isolated) backups may put the company's business in danger, or even in bankruptcy. The company may lose access to all of its orders, and the listings of its providers and clients; the company may have to pay a huge amount of money for the decryption key, which in the worst case may never be delivered.

The attack surface has considerably increased due to the number of connected devices, some of them being weakly secured or completely unsecured, and most of them operated by uninformed users. This has tremendously increased the chances for an attack to be successful, to the benefit of the attacker, making cyber-criminality more profitable. Besides, the attack may be launched from anywhere in the world.

The situation is likely to worsen in the upcoming years with the spread of connected devices forming the Internet of Things (IoT), which is still only in its infancy. The number of connected digital devices may increase by at least another order of magnitude. This interconnection exposes everyone to the weakest level of security of the trusted machines to which anyone is connected, as they may serve as an attack relay. By relying on low protection levels, we not only increase the risk for ourselves, but also the risk of contributing to the spread of a large-scale attack. This means that each actor has a responsibility and a moral obligation

towards the community to deploy sufficient protection measures. The IoT increases both the attack surface and the contamination aspect of cybersecurity breaches, which calls for yet higher security levels.

Diffusion across computer science

Since an entire system is only as secure as its weakest link, the interconnection and interaction of machines and applications (which often run in a distributed environment) makes cybersecurity a central question for almost all software today. It is therefore not surprising to observe the diffusion of expertise in cybersecurity to other domains of computer science.

There are several converging reasons for this evolution, leading to cross-fertilization between disciplines. For example, cybersecurity has been a predilection domain for formal methods research: cybersecurity was in urgent need of expertise in the area of formal methods and, simultaneously, formed an extremely challenging and stimulating playground for researchers in formal methods. Formal methods are now well-established in cybersecurity, especially in France, although this is still a growing research domain (see §3.3.2 and §3.3.3).

In the area of databases, researchers have been more and more exposed to security issues. At first they were cybersecurity users, importing off-the-shelf techniques to their research domains, but they have quickly become active contributors to cybersecurity. For example, the database community has proposed new paradigms such as the private cloud to enforce privacy by design.

In other areas, such as system security, distributed computing, and network services, the reasons are more intricate; but in each case, the evolution reflects the increasing exposure of these domains to security issues. More recently, there have been new and fruitful interactions between cybersecurity and machine learning, in both directions: on the one hand, machine learning methods are being applied to security, especially in reactive security (§4.4 and §4.6). On the other hand, machine learning raises new security and privacy issues (see §6.3.3).

Diffusion across other sciences

Even more importantly, cybersecurity is becoming a major concern for application domains outside of computer science that use computer devices or digital services in critical ways: e-health, medicine (§6.3.1), robotics (§6.3.2), power and water supply plants, smart cities, and, more generally, all critical infrastructure (§6.2.2).

This interdependence of computer science subdomains regarding security issues is both a real challenge and an opportunity for Inria to seize, since it is present in most of these subdomains. The diffusion of cybersecurity across other sciences is a chance to apply cybersecurity research, for which there will be lasting demand—and expectations.

1.2 The scope of cybersecurity

Wikipedia defines cybersecurity as follows:

Computer security, also known as cyber security or IT security, is the protection of computer systems from the theft and damage to their hardware, software or information, as well as from disruption or misdirection of the services they provide.

However, the exact notion of cybersecurity differs depending on the context[1]. Security in general includes both cybersecurity and physical security. However, cybersecurity requires some form of physical security, since physical access to computer systems enables a whole class of attacks. Conversely, physical security may depend on cybersecurity to the extent that it uses computer systems, e.g., to monitor some physical space or maintain a database of authorized persons. Still, the difference between cyber- and physical security should always be clear, and we only address cybersecurity hereafter. Moreover, in many places, we will just use the word security to mean cybersecurity.

[Note] Physical security *vs.* cybersecurity

Physical security and cybersecurity are quite different in nature.
Digital information is *immaterial:* duplicating and exchanging data and code with anyone anywhere in the world is nowadays a trivial, extremely fast process, with almost zero cost. Hence, an attack or malware launched by a single person can spread worldwide, at a large-scale, in less than an hour.
Digital information is of *discrete nature:* a single bit flip may introduce a critical failure and turn a perfectly working system into a malfunctioning one, which is then more vulnerable to compromise. This contrasts with the laws of physics, which tend to be continuous at a macroscopic level, and usually let one observe a slow deformation of a structure before it reaches its breaking point. Digital information ignores borders, and may even play with contradictions between the legislations of different countries or their maladaptation to the digital age.
This makes cybersecurity much harder to achieve than other forms of security.

Safety *vs.* security

Software *safety* is concerned with the absence of misbehavior, both in normal and exceptional situations, but still in a *neutral* environment when no one is trying to intentionally attack the system. Software safety is not just a matter of chasing bugs: it also calls for an analysis of the possible sources of misbehavior

1. ENISA, "Denition of Cybersecurity: Gaps and overlaps in standardization", version 1.0, December 2015. https://www.enisa.europa.eu/publications/definition-of-cybersecurity.

and how to handle them in a fail-safe manner. This requires a specification of the software's expected behavior, including a model of the environment, and some justification as to how or why the software respects its specification.

In contrast, software *security* aims for the absence of misbehavior in an *adversarial* environment, where an attacker intentionally tries to misuse a system, putting it in an erroneous state that is not part of its intended specification. Security can also be approached by modeling the environment, but this is much harder to achieve exhaustively, because attackers do not comply with predefined rules, but rather continuously search for previously unknown means of attack. Hence, security also requires us to keep up-to-date with attackers' progress in all areas (software breaches, algorithms and techniques, hardware capabilities, etc.). A complementary approach consists in describing normal execution paths and monitoring execution, so as to raise an alarm and react appropriately when some trajectory goes outside of normal executions.

[Note] Safety vs. security

The terms *security* and *safety* are sometimes misused. Safety refers to *accidental* threats, due to internal misbehaviors or non-intentional misuse of the system, while security refers to *intentional* threats. Safety deals with fault-tolerance, while security deals with resistance to attacks. For example, a car may crash because of a software specification or an implementation bug (safety issues), or because of an attacker taking remote control of the vehicle (a security issue).

Despite these differences, safety and security are often tied to one another. Since security works in an adversarial mode, it should also consider accidental threats which may be exploited by the attacker. Hence, security is a stronger requirement than safety. In many situations, however, the software is exposed to the outside world, typically connected to the Internet, where attacks are the norm—and safety without security would often not make much sense!

Safety and security also share a lot in their methodologies. Dealing with the safety of large software systems that interact with the physical world, such as Cyber-Physical Systems (often known as CPS—see §6.2), has led to some well-established methodologies. One should start with a safety risk analysis phase, where all situations that may lead to catastrophic consequences are explored. Representations such as fault trees can be used to systematically describe such situations. The identified risks are then quantified to estimate the probability of the occurrences of these situations. Ensuring safety means ensuring that this probability remains below a given threshold. Of course, a first step to satisfy the safety property is to ensure the absence of internal faults (bugs) in the software, as these faults are the primary cause of failures. Formally, one writes a software specification to describe the expected behavior of the software, and then shows

that the actual implementation satisfies the specification. Unfortunately, not all cases may have been considered in the specification. Moreover, there may be external faults (for example, an erroneous value coming from an external sensor) that are not considered in the software specification, and that can lead to disasters. Hence, we must also use additional mechanisms, essentially based on dynamic error detection and recovery mechanisms, used to treat the errors due to external faults before they lead to catastrophic consequences.

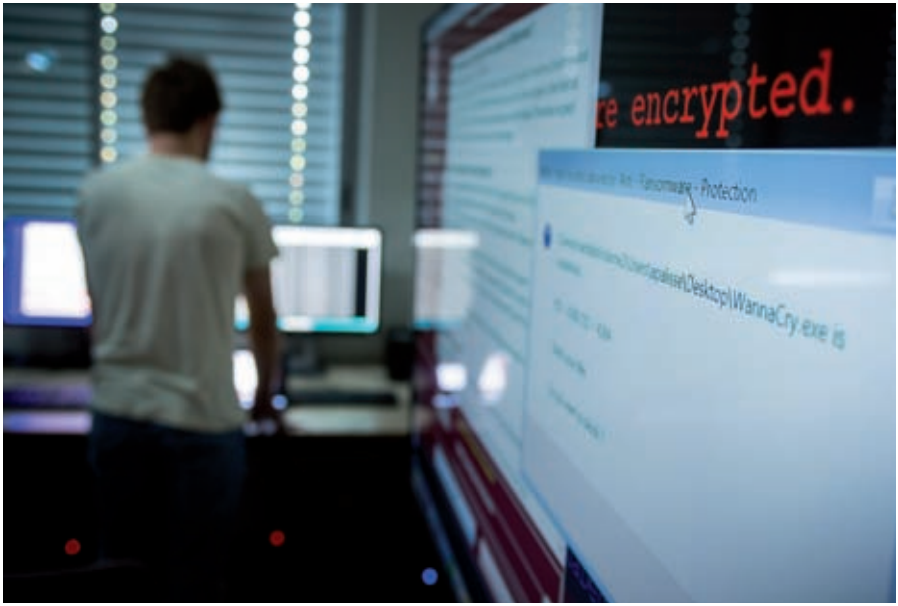
A similar approach applies to security. A security risk analysis (see §2.4) replaces the safety risk analysis. While it is not possible to reason statistically to build an unassailable system in the case of security, it is still useful to ensure that there are no bugs (at least of some kind) in the software, for example by the same formal approach as the one followed for safety, because attacks often build on vulnerabilities that stem from a remaining bug. Monitoring takes the form of dynamic attack detection and recovery mechanisms. This implies a model of the attacker, which should at least cover all known types of attacks, for example in the form of an attack signature base. In this view, the safety-by-design principle becomes the security-by-design principle, meaning that the software must be designed from the foundation to be secure[2]. This has led to design principles such as the OWASP recommendations[3].

However, security and safety remain distinct and different domains, built on different hypotheses, and the protection mechanisms against accidental and intentional threats are usually complementary. In this white book we restrict our attention to security.

1.3 A few examples and lessons learned

Unfortunately, cybersecurity incidents are common, and too often make the headlines. Here we describe a few illustrative examples, in order to highlight the huge diversity of attacks. Some of them target well identified entities, such as TV5 Monde and the Dyn company, although the attacks used totally different techniques. At the other end of the spectrum the Wannacry ransomware targeted a huge population, propagating in a viral manner. Certain electronic voting systems are known to be vulnerable and, on several occasions, security researchers have highlighted their inadequacy through proof-of-concept attacks. The following example illustrates how anonymized databases can sometimes be attacked, revealing physical identities. The last two examples highlight, for the first one, two hardware-targeted software attacks that exploit advanced processor performance optimization techniques, and for the second one, weaknesses of some Internet of Things devices and their exploitation.

2. See Wikipedia https://en.wikipedia.org/wiki/Secure_by_design
3. https://www.owasp.org/index.php/Security_by_Design_Principles



Research on malware – © Inria / Photo C. Morel

The TV5 Monde targeted attack: on April 9th, 2015 the French TV network TV5 Monde was the victim of a major sabotage. Around 9pm, the website and social media channels (Facebook, Twitter, YouTube) were defaced. About an hour later, the network infrastructure was no longer operational and broadcasting was interrupted, resulting in a complete TV blackout—the worst thing that can happen to a TV network. The French National Cybersecurity Agency (ANSSI) later found that the attack was carefully planned [4]. The attackers first connected in January, using a stolen login and password. This allowed them to get access to the internal network, to collect internal documents containing information on the network infrastructure and existing accounts, and to exploit unconfigured services that still relied on default accounts and passwords. Deleting firmware on the network infrastructure (routers and switches) then caused the breakdown, making a simple restart impossible.

Denial of service attacks from the Mirai botnet of home devices: the Mirai malware's goal is to turn vulnerable home devices (such as IP cameras, printers, baby monitors, or home routers) into remotely controlled bots that can later be used to launch large-scale denial of service attacks. This is what happened on October 21st, 2016, when this botnet targeted the name servers of the Dyn company

4. https://static.sstic.org/videos2017/SSTIC_2017-06-09_P09.mp4

(used for instance when translating domain names to IP addresses, see §2.2). This attack resulted in a blockage of many web sites worldwide for several hours.

The Wannacry ransomware: on Friday May 12th, 2017, the WannaCry ransomware propagated throughout the world, infecting more than 230,000 computers in over 150 countries within a single day (source Wikipedia). This ransomware targets computers running the Microsoft Windows operating system, with major consequences for their owners: after infecting a computer, the ransomware encrypts data and displays a note to inform the user, asking for a bitcoin payment in exchange for the decryption key. This ransomware is considered a worm, since it scans for vulnerable systems and then replicates itself on these new targets.

Electronic voting vulnerabilities: in the last few years, several European countries (Estonia, France, Norway, and Switzerland) held legally binding political elections that allowed part of the voters to cast their votes remotely via the Internet. French people living abroad were allowed to vote via the Internet for the June 2012 parliamentary elections. An engineer demonstrated that it was possible to write malware that could change the value of a cast vote, with no way for the voter to know. In the 2011 Estonian parliamentary election, a similar attack was reported by computer scientist Pihelgas, who conducted a real-life experiment with fully aware subjects.

Re-identification in the AOL anonymized database of web search queries: as reported in the New York Times [5], AOL released an anonymized database containing more than 20 million web search queries. Even though the data was anonymized, users could be identified after some investigation, thereby revealing all their personal search queries. More generally, database anonymization is a complex task with pitfalls, that requires finding an appropriate balance between utility and privacy.

The Spectre and Meltdown vulnerabilities: on January 3rd, 2018, two hardware vulnerabilities, Spectre and Meltdown, were simultaneously released. Both vulnerabilities exploit speculative execution (and in particular branch prediction), an optimization technique in modern processors. To avoid idle processor cycles, e.g., while waiting for the result of a memory access, processors may perform out-of-order execution. A branch may then be speculatively executed, while waiting for the evaluation of a conditional. If the branch was wrongly executed, the results are discarded. However, even if the results are discarded, a memory access nevertheless leaves a trace in the cache. The idea of the Spectre and Meltdown attacks is to force a forbidden memory access. Typically, buffer overflows are prevented by checks on the size of the buffer. These checks can be circumvented however by making the branch prediction wrongly predict the test. Then, a cache attack can be used to check which area of the memory has been executed. (Such

5. New York Times, "A Face Is Exposed for AOL Searcher No. 4417749", August 9, 2006 http://www.nytimes.com/2006/08/09/technology/09aol.html

attacks simply measure the time necessary for accessing a particular memory address.) The attacks are particularly severe, because they exploit the design of modern processors, and cannot be simply patched by a software update. Moreover, speculative execution is at the core of modern processor design, and is unlikely to be abandoned by processor manufacturers.

Smart lights causing epilepsy seizures: researchers from the Weizmann Institute of Science have shown that it is possible to hack commonly deployed smart lights, and to strobe them at a frequency that may trigger epileptic seizures [ER16]. The attack is interesting because by turning traditionally unconnected objects (here, light bulbs) into smart objects, they can be misused to create an unexpected attack. This particular attack exploits a combination of several flaws. First, when initializing the smart light controller, the password that allows the controller to connect to the local WiFi is sent unencrypted and can easily be sniffed. Second, the lights accept commands from any devices on the local WiFi without a proper authentication mechanism. Third, the controller does not verify the length of the commands it receives, allowing the concatenation of multiple commands, circumventing the limit on commands that may be sent per second. Finally, the attack is based on undocumented API options, allowing attackers to create a strobe effect.

Lessons learned

These examples highlight several key aspects of security:

→ *Security is an essential cornerstone in a digital world which increasingly pervades every aspect of our daily lives, public and private.* Without security, the world collapses. Attacks such as WannaCry have deeply impacted unprepared citizens, private companies, and organizations, threatening their activities.

→ *All the domains of our digital world are concerned,* including the embedded devices omnipresent in our "smart" homes, and in industrial production controllers (including those for critical infrastructures like power and water supplies). Since all of them are connected to the Internet, security is a serious concern, as demonstrated by the attack on smart lights mentioned above. The Mirai botnet example highlights that all electronic devices need to be secure. Even if this is well understood for computers, it is far from obvious for other objects, in particular embedded devices forming the Internet of Things (IoT)—either because they are autonomous, have a small battery, limited processing power, or are badly connected. Moreover, the inability of IoT devices to apply software updates and patches is a real concern. On the other hand, software updates can themselves be subject to attacks. All of these aspects are still the subject of active research.

[ER16]. E. Ronen, A. Shamir. Extended functionality attacks on iot devices: The case of smart lights. In *IEEE European Symposium on Security and Privacy (EuroS&P'16),* 2016.

→ *Education is essential to security.* The WannaCry attack relied on an operating system exploit that had been fixed in a Windows update two months earlier. This only impacted unprepared end users and system administrators who failed to update their computers in a timely manner, not realizing how important it was. Security is often regarded as complex, mechanically limiting its usage. Usable security, meant to facilitate use of security by end users, is an important and active research domain that is closely related to security education and awareness.

→ *The security of a system is always limited by that of its weakest component.* Even if the core security components (e.g., the cryptographic primitives) are rarely attacked, the same cannot be said of the software implementations of the cryptographic protocols and services. In the case of WannaCry, the attack relied on an exploit of the Windows SMB protocol (the first weak link), which was sufficient to take full control of the computer, no matter what other operating system protections were in use. The second weak link was the users, who by not updating their computers made the attack successful. The TV5 Monde attack was made possible first of all by social engineering, and then by the use of unmodified default login/passwords on various technical equipments.

→ *Obscurity does not increase security.* Sometimes, people believe that hiding the internals of a system or security mechanism will increase security. However, by now we know that, on the contrary, open design principles improve a system's security. In cryptography this fact is known as Kerckhoffs's principle, and goes back to the 19th century: *A cryptosystem should be secure even if everything about the system, except the key, is public knowledge.* This principle should be applied to other systems as well. An open design and well documented system will actually ease security reviews by experts. Attackers are often able to reverse engineer systems, and "security by obscurity" only gives a false sense of security. For instance, the attack on smart lights exploited an undocumented functionality.

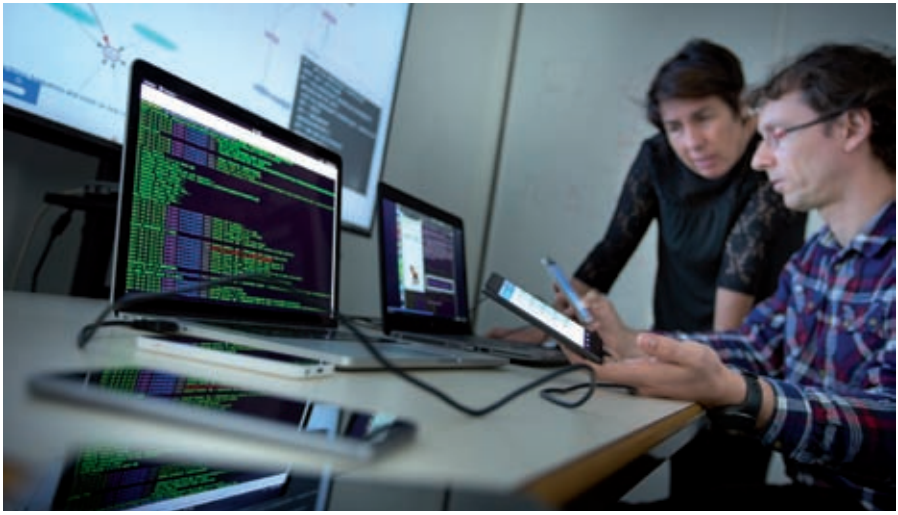
→ *Large, complex systems cannot be totally validated through human inspection: automatic verification tools are needed to find security protocol flaws as well as implementation flaws.* The SMB component targeted by WannaCry has a long history behind it. In spite of that, a security flaw could be identified that made the attack possible. The increasing complexity of each individual component, and the complex composition of components in large interdependent systems, require advanced and automatic security validation tools, which is traditionally a very active research topic.

→ ***Security and privacy are closely related.*** The WannaCry ransomware did not try to exfiltrate user's data, but it could have done so. The attacker had full access to data stored on target computers (e.g., the patient database of a medical center) and could have threatened to disclose this sensitive information. It is therefore essential that security and privacy be considered together at the design stage so that, for instance, malicious intrusions do not put data at risk. Security by design, and more recently privacy by design, have become key principles in security design.

→ ***Diversity of attackers' motivations and the difficulty of attribution.*** Although WannaCry has been classified as ransomware, motivated by the desire to make money, the NotPetya malware that quickly followed it in June 2017 might be a "state-sponsored malware that attempted to disguise [itself] as ransomware in order to muddy [its] attribution and potentially to delay investigations."[6] These examples highlight the diversity of the attackers' motivations and the difficulty—sometimes, the impossibility—of attributing an attack.

→ ***Detection and mitigation of attacks.*** The previous examples show that security is hard to achieve. Since zero risk cannot exist, the early detection and mitigation of attacks is as important as the attempt to reduce the risk of successful attacks.

More generally, there will probably always be vulnerabilities in our systems, despite increasingly efficient preventive security mechanisms. The vulnerabilities appear at all levels of our information systems: applications, OS, firmware, and



Malware analysis and detection – © Inria / Photo C. Morel

6. https://elie.net/blog/security/unmasking-the-ransomware-kingpins/

even hardware, as illustrated recently by the Meltdown and Spectre attacks. Vulnerabilities are sometimes present for a (very) long time in our systems, and we can only hope that they are not exploited before they are discovered. New vulnerabilities are discovered on a daily basis[7], and new forms of attacks can appear at any time. It is mandatory that we detect well-known attacks, but also new forms of attacks, if we are to increase the security level of our systems.

→ *Security comes at a cost.* It is easy to understand that security may be expensive, with additional costs to study, implement, configure, manage, and evolve security tools. But security can also have an operational cost, leading to less efficient systems. For example, mitigating the Spectre or Meltdown attacks may require removing some cache techniques or disabling speculative execution. Such mitigation would entail a significant and possibly unacceptable processor-speed slowdown. Hence, in some cases, one may have to accept a difficult compromise between security and efficiency.

Each of these topics is the subject of active research and is introduced in this document.

[Note] The cybersecurity threat

The cybersecurity threat is *real* and *serious*. We only see the tip of the iceberg: in the vast majority of cases, even the existence of the attack is a critical piece of information for companies or states that is rarely released.

For experts, the question is not whether large scale cyberattacks will eventually succeed, as the answer is definitely positive, but rather: are we sufficiently prepared? This means we should of course reduce the risk of such attacks by better preventive and reactive protection, but also increase our cyber-resilience including pre-established procedures to reduce the catastrophic impacts of successful attacks, and faster recovery to a safe mode of operation after such attacks.

1.4 Security properties, services, and mechanisms

Security Properties

Cybersecurity consists in ensuring three basic and essential properties of information, services, and IT infrastructures well known as the CIA triad: Confidentiality, Integrity, and Availability. Thus, securing an information system means preventing an unauthorized entity (user, process, service, machine) from

7. https://www.cvedetails.com/browse-by-date.php

accessing, altering, or rendering inaccessible computer data, computing services, or the computing infrastructure. Notice that other properties, such as authenticity (proof of the origin of information), privacy, or protection against illegal copying could also be listed. However, these additional properties can also be seen as particular cases of these three basic properties.

[Note] On cybersecurity properties

Confidentiality: assurance that information is disclosed only to authorized persons, entities, or processes.

Integrity: assurance that the system (configuration files, executable files, etc.) or information are modified only by a voluntary and legitimate action, i.e., that the system or information have not been accidentally or deliberately changed.

Availability: assurance that a system or information is accessible in a timely manner to those who need to use it.

Authenticity: assurance that a message is from the source it claims to be from.

Privacy: ability for individuals to control their personal data and decide what to reveal to whom and under what conditions. Privacy can thus be generally defined as the right of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others.

Anonymity: confidentiality of the identity of the user or entity. We note that preventing re-identification through side information is not easy, and that *indistinguishability,* which ensures that an attacker cannot see the difference among a group of entities, is also an important property linked to privacy. Note also that anonymity aims at hiding who performs some action, whereas full privacy may also require hiding which actions are being performed.

Security policy: a set of rules that specify how sensitive and critical resources are protected, i.e., how some or all of the previous properties are guaranteed.

Resilience: initially defined as the ability of a system to return to its original state after an attack, resilience is nowadays seen as the capacity of a system to deliver its services continuously, even while under attack (i.e., capacity to tolerate attacks).

Security Services

Reaching the objectives of cybersecurity requires enforcing *physical, organizational,* and *logical* counter-measures. Even if physical measures (such as guarding or controlling accesses to buildings) and organizational measures (such as precisely defining the mission of an external IT service provider) are crucial, we focus in this document on logical security, i.e., on hardware and software services and mechanisms to ensure the properties of confidentiality, integrity, and availability.

A secure computer system must offer *preventive* services to hinder any violation of these properties, detection services to identify any successful attempt to violate these properties, and reaction services to deploy new or enhanced

counter-measures in case of any successful violation. Indeed, while the goal of cybersecurity is to protect a computer system against attacks, one must also assume that some of the attacks will succeed. Therefore, cybersecurity also deals with intrusion detection and responses to attacks.

Prevention first involves precisely defining which entity may access what information and in which way: permissions, prohibitions, or obligations [8] to read or write information are to be defined. This constitutes a so-called *security policy.* Prevention can even take place before the definition of a policy. Indeed, it is good software engineering to detect early source and binary code vulnerabilities that could be exploited to violate the security properties: this is the *security by design* principle. Even earlier on, we may also prove that a given property is guaranteed by the software: this is *formally proved security.*

The security policy is concretely enforced through *security services.* The following services can be offered, depending on the policy and on the context: entity identification and authentication, control of access to information by these entities, control of information flows within the system, detection of attempts to exploit potential vulnerabilities of the system (intrusion detection, virus detection), and responses to these attempts (reaction).

An even more ambitious objective that could be pursued would be the ability for a computer system to *deliver the intended outcome despite adverse cyber events.* In other words, the computer systems would *tolerate attacks,* a capacity generally called *cyber-resilience* [9]. From a high-level point of view, an entity (state, company, organization, etc.) could be more concerned with cyber-resilience, which is the final objective to achieve, than with cybersecurity, which is a set of deployed techniques that the end-user need not necessarily see.

Cyber-resilience, being the capacity to tolerate attacks, has of course a lot of similarities with fault tolerance, which deals with hazardous hardware failures or software bugs. Even if the hypothesis of safety and security are quite different, since attackers do not follow the rules but rather continuously search for new breaches, the mechanisms proposed to tolerate faults may be adapted to tolerate attacks. Thus, some basic principles of cyber-resilience include replication of data and backups, which have long been well-established in the database community. Besides, replication should be used in the context of a distributed system, to avoid having a single point of failure. While cyber-resilience is of major importance, many techniques to achieve it are reminiscent of other fields (e.g., safety), and will not be detailed in the remainder; others are completely relevant to the security field (e.g., DDoS Mitigation) and will be discussed in the corresponding sections.

8. Obligations correspond to preconditions to fullfil to be permitted to read or write information. For example, a user can be authorized to sign a file if and only if this file has already been previously signed by another given user. Obligations are generally enforced at the application level.
9. Source https://en.wikipedia.org/wiki/Cyber_Resilience

Security Mechanisms

Security services rely on mechanisms implemented at various levels of information systems and infrastructure, including hardware, firmware, operating systems, network layers, hypervisors, and applications. Cryptography is of course a fundamental building block in many cases: the study of *cryptographic primitives* and their use in exchanges between machines (through *cryptographic protocols*) are therefore two essential aspects of digital security.

1.5 Legal aspects

1.5.1 European security regulation

The European Union cybersecurity strategy[10] is built on several instruments, with the goal to improve European cyber-resilience and response while preserving for each nation a level of sovereign capacity to control the main components of their national defense strategy.

The European Union Agency for Network and Information Security (ENISA)[11], set up in 2004, is an important actor of the European cybersecurity landscape. A significant extension of its missions is currently under discussion, aimed at making it the privileged interface with the member states, including support in the implementation and operation of cybersecurity directives.

In mid-2016, the European Union adopted the Directive on security of network and information systems (known as the NIS directive)[12], for an application mid-2018. This directive focuses both on Digital Service Providers (DSP) and Operators of Essential Services (OES). Both DSP and OES are held accountable for reporting security incidents, even if services are run by non-European companies or if the information system management is outsourced to third parties. DSP and OES are also required to provide information that allows for an in-depth security assessment of their information system and policies. Finally, the member states are required to identify agencies in charge of collecting and processing security incidents, in addition to a national competent authority (e.g., ANSSI in France).

There is also a need to promote the development of secure-by-design products and services throughout Europe. In order to achieve this goal, the European Union is putting forward a proposal to set up a European security certification framework, capable of issuing European level security certifications and labels of products and services. This is a daunting task, because even though a very broad set of security certification schemes exist throughout the world, no unified or combined solution exists. Understanding what is required to make things

10. https://ec.europa.eu/digital-single-market/en/cyber-security
11. Initialy called European Network and Information Security Agency, see https://www.enisa.europa.eu/
12. https://ec.europa.eu/digital-single-market/en/network-and-information-security-nis-directive

consistently secure is a complex task. The European Cyber Security Organization (ECSO)[13] liaises with the European Commission for the definition of a European security certification framework proposal. ECSO has issued a state of the art of existing industrial cybersecurity standards for various activity domains, and is working on a so-called meta-scheme approach encompassing many existing certification schemes, evaluating the level of confidence provided by individual schemes and mapping them onto a harmonized set of security levels[14].

Because zero risk does not exist, the European Commission has also issued an official blueprint, called *Commission Recommendation of 13.9.2017 on Coordinated Response to Large Scale Cybersecurity Incidents and Crises*[15]. This recommendation sets out the objectives and modes of cooperation between the member states and European institutions when responding to such incidents and crises.

The European Union is at the forefront of privacy and data protection, with the new GDPR regulation and the ePrivacy regulation that will complement it—see §5.1.2 where these aspects will be discussed.

1.5.2 Forensic analysis

Generally speaking, forensic analysis is related to scientific methods of identifying the authors of a crime by examining objects or substances involved in the crime. In the context of cybersecurity, forensic analysis is related to the explanation of a cybercrime, based on the analysis of information or traces left by the attacker in the computing systems used or attacked.

Concretely, forensic analysis aims to explain the state of a computing system by extracting information and using it to reconstruct the series of actions undertaken by the attacker. A special effort has been made to produce techniques and tools to analyze the content of mobile phones, which are often used to prepare and perform criminal actions in the real world. Here, defense against cybercrime joins defense against crime.

1.5.3 Surveillance and security

With the increase of the terrorist threat, we have witnessed, in several countries, the deployment of mass-surveillance systems intended to help fight terrorism. In France, this took the form of the "Loi relative au renseignement."[16] In particular, this law requires the deployment of black boxes within French Internet Service Providers (ISPs) in order to both collect connection information from specific previously-identified targets in real time, and to analyze connection information

13. https://www.ecs-org.eu
14.Documents produced by ECSO WG1 are available at:
http://www.ecs-org.eu/working-groups/wg1-standardisation-certification-labelling-and-supply-chain-management.
15. http://ec.europa.eu/transparency/regdoc/rep/3/2017/EN/C-2017-6100-F1-EN-MAIN-PART-1.PDF
16. Loi no 2015-912 du 24 juillet 2015 relative au renseignement, Legifrance.
https://www.legifrance.gouv.fr/affichTexte.do?cidTexte=JORFTEXT000030931899

of ISP subscribers in order to identify potential suspects via an automatic process (whose details are not publicly known). The re-identification of the subscriber requires an official decision of the Prime Minister (or a delegate).

These laws highlight the tension between public security and privacy. They have also been criticized because of their economic cost and potential inefficiency, in particular when faced with the "false-positive paradox."[17] The risk is that "more false positives will only overstress technologies, thus causing even more work for signals-intelligence agents, who are already overloaded."

The term "mass dataveillance" has been given to practices where governments or governmental organizations perform surveillance and data collection at a national scale (or larger). This is opposed to "personal dataveillance", which targets an individual of (supposed) interest.

As a reaction to this evolution of surveillance (and in particular to E. Snowden's revelations), the IETF has considered that "pervasive monitoring is an attack" in RFC 7258[18] and that IETF protocols should mitigate it. Encryption by default is among the IETF's initiatives (§2.2).

Surveillance and cyber defense are complex topics by nature. AMNECYS (Alpine Multidisciplinary NEtwork on CYber Security studies)[19] is an example of multi-disciplinary initiative that has gathered several research teams in order to contribute to this complex question.

1.6 Sovereignty issues

Since most critical infrastructure is now controlled by computers, often connected to the Internet, protecting infrastructure requires protecting computer systems and networks. Hence, cybersecurity is a sovereignty issue for states and the EU. Therefore, states and the EU must have the ability to understand the risks and threats. This requires the highest scientific competences, and can only be maintained in the long-term by pursuing advanced research in all domains of cybersecurity. We not only need to have the best experts, but we also must have them in sufficient numbers to cover the increasing needs (see §2.3.3). In addition, we also need experts at intermediate and lower levels to be able to implement the security policies correctly.

States should also have the power to act. This requires sufficient control over digital infrastructure and the whole cybersecurity chain, as the security of the whole depends on the security of the weakest link. This implies control over the software and hardware used in critical infrastructures, so that it can be analyzed and certified free of bugs and backdoors. This also implies control over data storage.

17. https://hal.archives-ouvertes.fr/hal-01157921/document
18. https://www.rfc-editor.org/rfc/pdfrfc/rfc7258.txt.pdf
19. http://amnecys.inria.fr/

Hardware is one of the weakest links, as France and Europe do no longer have the capacity to design and produce their own hardware. Hence, some form of sovereignty has already been given up. Indeed, it is quite possible that hardware devices are rigged with backdoors or hidden functions that allow, for example, a government agency or a company to spy on Internet traffic or to prevent the operation of a particular service.

In fact, the digital and thus dematerialized nature of cybersecurity makes cybersecurity sovereignty different from other forms of sovereignty, such as defense. While the latter is the privilege of states or supranational organizations, the former can be implemented at smaller scales. Many entities (citizens, companies, associations, etc.) can claim some degree of sovereignty over the security of their own data, computer systems, and networks. A consequence of digitalization is the potential transfer of some of the traditional state sovereignties to other entities: blockchain-powered land registration, minting money with digital currencies, or citizen identification services[20], etc. These different levels of sovereignty do not exclude, but rather complement one another, leaving sovereignty of each kind of data at the most appropriate level. This ability of cybersecurity to be decentralized should not endanger the sovereignties of the states. On the contrary, it is a chance that should be exploited, leaving some autonomy to the different entities within certain limits established by incentives, regulations, and laws.

20. For example, SecureIdentity https://secureidentity.co.uk/

Knowing, understanding, and modeling threats

There are many types of attacks against information systems. The threats are therefore numerous. The attacks can target the hardware, the network, the system, or the applications (very often through the malicious actions of a malware), or even the users themselves (social engineering, phishing). The attacker can be an insider, or an outsider.

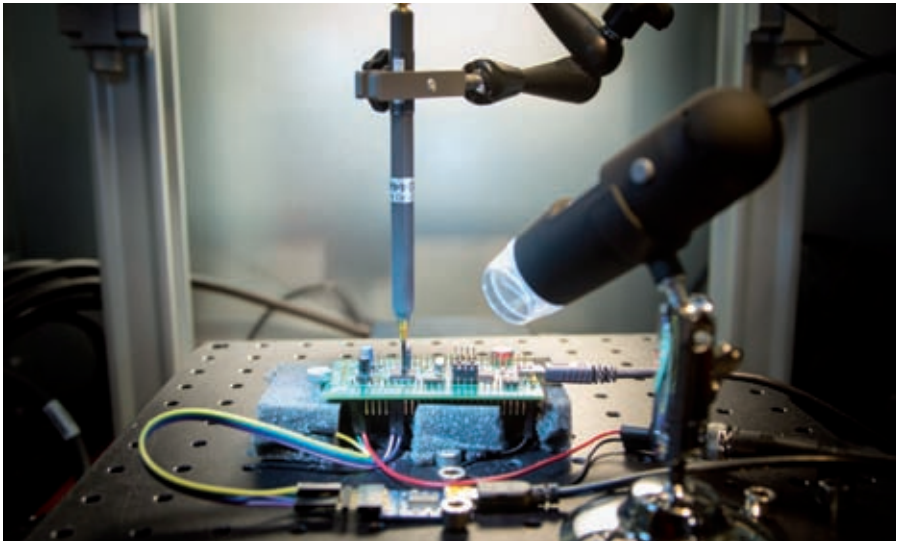
In this chapter, we present bodies of work carried out in order to build a better knowledge of such threats and attacks. Various attack models can be considered in these bodies of work, to define the knowledge of the attacker about the system and the actions the attacker can realize. The section devoted to the human factor also focuses on two crucial aspects of security: usability and education.

Please note that cryptanalysis aims at understanding the threats against existing cryptographic primitives, in order to be ahead of malicious adversaries. Cryptanalysis is thus the foundation of confidence we can have on these primitives, and the knowledge of state-of-the-art cryptanalysis can be seen as the backbone for the design of secure primitives. Cryptanalysis per se is therefore not a threat, but a way to achieve a better security level. It is presented in §3.1.2. Malware analysis and malware detection are presented in §4.5.1.

2.1 Hardware attacks

[Summary]
Physical attacks against the hardware are a real threat, even for implementations of algorithms that have been mathematically proven secure. Attacks by observation and attacks by perturbation are two common classes of attacks that require a physical access to the device. More recently, hardware has also been attacked through the software, which is even more dangerous, as it does not require physical access to the device. Nowadays, a serious but unfortunately possible attack scenario is an hardware attack triggered by a JavaScript application embedded in a web page.

When discussing the security of an algorithm, numerous mathematical tools allow developers to assess its security. Unfortunately, those tools cannot consider the interaction of the computing unit with its physical environment. Physical attacks are a real threat, even for algorithms proven secure mathematically. These attacks can be classified as observation attacks, perturbation attacks, and a new field known as hardware-targeted software attacks. The first two assume the insider attacker model i.e., the device is under the attacker's control, while the last one assumes the outsider model. The outsider model requires fewer hypotheses for the attacker and thus can be considered as more dangerous.



Studying fault attacks – © Inria / Photo C. Morel

Observation attacks

Side-channel analyses (SCA) are physical attacks based on the observation of the circuit behavior during a computation. They exploit the fact that some physical quantities depend on intermediary values of the computation in the device. This is the so-called information leakage. The most classic leakages are timing, power consumption, and electromagnetic emissions (EM).

SCA are threats for all standard cryptosystems such as Data Encryption Standard (DES), Advanced Encryption Standard (AES), RSA cryptosystem, Elliptic Curve Cryptography (ECC) and for critical applications not using cryptography, e.g. PIN verification. SCA can also be used to reverse engineer algorithms.

Perturbation attacks

Fault attacks are now a well-known class of physical attacks where a device undergoes a modification of physical parameters in order to obtain an incorrect behavior. Most classical fault injection means are power glitches, clock glitches, laser pulses, and electromagnetic pulses. Fault attacks have been shown extremely efficient against cryptography, e.g. the Bellcore attack allows any fault, at the correct time, on an RSA-CRT signature to recover the secret. What fault can be achieved and what is the fault model is an active area of research.

This class of attack is chip dependent, i.e. what has been learned of a SoC is not valid for other chips even if they are highly similar (same core). The success of the attack relies essentially on the experiment set up due to the large amount

of parameters (type of the EM probe, distance of the probe on the circuit, form and amplitude of the EM pulse, etc.). Another challenge in this class of attacks is the effects observability. To understand the precise effect, one has to explore the internal state of the chip which is often not available. Most of the counter-measures are related with either temporal or spatial redundancy. The cost of such a redundancy is not affordable for low end devices. Research is focusing on lightweight redundancy to ensure the integrity of the execution.

Hardware-targeted software attacks

In addition to software attacks against software and physical attacks against hardware-targeted software attacks, appeared in the mid-2000s software attacks against hardware components. For example the Rowhammer attack aims at flipping memory bits while reading and writing another cell. The insider attacker model moves to an outsider one when using a JavaScript program executed in a browser to perform this attack remotely. Recently it has been shown to be effective when applied on SSD disks (NAND flash technology).

Perturbation can also be generated in multicore SoC using the Dynamic Voltage and Frequency Scaling (DVFS), i.e., the energy management technique that saves energy by regulating the frequency and voltage of the processor cores. It has been shown that a misconfiguration of these two parameters can be used to induce faults in the hardware. Each core being individually controlled, one core can inject a fault in another core. Even if it has not yet been demonstrated, this attack should be achievable from within a browser.

Software-based attacks against hardware make it possible to circumvent security mechanisms implemented at the software level. In fact, the software protections consider that the hardware is working properly, "simply" executing instructions to produce a result. Of course, this is not so easy and errors that can be exploited by attackers can also occur at the hardware level.

More generally speaking, the traditional approach of computer science and technologies, constantly adding new more and more powerful levels of abstraction naturally leads, when proposing a security mechanism at a given level of abstraction, to consider that the lower layers are correct and safe. This is however not the case; this is why the attackers have had a tendency these last years to target less and less abstract layers, successively attacking by software the applications, the OS, their kernel, the firmware, and now the hardware.

These low-layer attacks typically exploit flaws from optimization mechanisms implemented in modern OS's and processors, such as caches, branch prediction, or speculative execution. Indeed, these optimizations create differences in program execution time, thus revealing secret information. For example, the recent Spectre[1]

1. https://spectreattack.com

attack exploits branch prediction and speculative execution and exfiltrates information through a covert channel based on cache access. To mitigate this attack, one could refresh the cells (read and re-write their values) periodically. This solution would of course come at the price of performance limitations, as other read operations asked by programs would not be possible during the refreshes. More generally, protection against attacks of this type would involve the limitation, if not the complete elimination, of certain optimizations, of course at the cost of lower performance.

The Rowhammer attack evoked above is a software attack that actually exploits a physical property of matter. Each DRAM cell is composed of a capacitor and a transistor that electrically implement a bit of information. By repeatedly accessing cells, the charge of these cells leaks and electrically interacts with the charge of other neighbor cells. It is thus possible to change the value of a cell (and therefore to violate the integrity of this cell) without having ever accessed it. Here, the protection against the attack should be physical: for example, one could consider limiting the reduction of the component's surface, even if the cost would be of course very important.

Notice that these attacks are not easy to detect as they leave no trace at the operating system or application levels.

Finally, it is difficult to know whether attacks of this type have already been used in reality. At the time of this document's writing, it seems much simpler to use much more classical attacks against the software or against users (social engineering).

[Research challenge 1] Hardware-targeted software attacks

Attacks against information systems do not usually involve the hardware layer but exploit a software vulnerability. However, recent attacks, such as Rowhammer, Spectre, or Meltdown, have shown that attacks implemented in software can exploit performance optimizations of the hardware. This new type of attack is especially dangerous as it makes hardware attacks possible at a distance, as opposed to classical side channel attacks. It is not yet completely clear how the current proof of concept attacks can be "industrialized," but they pave the way to a new class of serious attacks. Therefore we need to get a better understanding of how such attacks could possibly be deployed, propose a clear typology of this new kind of attack, and propose countermeasures, both at the hardware and the software levels. This task requires expertise at the hardware, firmware, and operating system levels. The countermeasures can also be difficult to design as they may require to revisit crucial optimizations used for years, such as speculative execution.

[Inria teams] Hardware attacks

↗ The **CAIRN** team works on efficient computing architecture. They do not address explicitly the security issues related to this optimization process, but they know how the SoC architecture can help.

↗ The **CIDRE** team analyses the impact of EM faults on the software stack on modern SoC but also on low end devices. The team also uses circuit observation to optimize the fault injection process. The objective is to evaluate the possibility to move from a high end platform to a low cost platform with the same result. The team characterizes the impact of the fault during the secure boot process to identify potential vulnerabilities. In a collaboration with the **PACAP** team, it verified the robustness of a countermeasure that uses a dedicated compiler to generate fault resistant code. To resist against side channel attacks they evaluated on the fly compilation solutions to increase the number of traces an attacker has to capture to extract the secret. They have also proposed a template attack against PIN verification implementations.

↗ The **PACAP** team evaluates the possibility to eradicate leakage in a code at compile time thanks to code annotation (compiler that generates resistant implementations to side-channel analyses attacks). Their solution is evaluated within the Rennes LHS facilities.

↗ The **TAMIS** team develops new side-channel distinguishers based on machine learning techniques due to their precise knowledge of the underlying leakage measurements and modeling of sensitive information.

2.2 Network threats

[Summary]

At the network level, many examples of attacks exist. Here are two examples targeting the Internet. Finding a path for each packet sent on the Internet, no matter its source and its destination, is a key service known as "routing"[a]: attacking this basic and essential network service can, for instance, isolate a whole country or at the opposite redirect all the traffic of a country through a surveillance point. Another crucial network service, DNS, translates readable hostnames into IP addresses. An attack against this service can redirect a user to a fake banking web site in order to steal the user's credentials. A secure extension to DNS, called DNSSEC, is now available, but its deployment will take time and will not solve all the problems, in particular those related to privacy.

a. See for example this simple introduction: https://interstices.info/internet-le-conglomerat-des-reseaux/

Any type of network may be attacked, taking advantage of its characteristics. We focus here on the Internet and some of its specificities: domain name, routing, and potentially non-encrypted payload.

The Internet is a complex assembly of an extremely large number of devices, from user machines or devices to routers, linked by a huge array of wireless and wired networking technologies. Its operation requires a vast range of information resources, protocols, and services, from low-level routing databases, forwarding policies, low-level MAC address/IP address/host-name mapping protocols (i.e., the ARP/NDP/SEND and DNS/DNSSEC protocols), or link-layer specific technologies (e.g., to create and manage a Virtual LAN) up to high-level services, like Web services. This inherent complexity of the Internet constitutes many facets that are all subject to threats. Hereafter, we consider a small subset of these threats and discuss trends to mitigate them, both from the academic and standardization viewpoints, e.g., as done by the Internet Engineering Task Force (IETF)[2].

Attacks against the Domain Name System (DNS)

The DNS is a hierarchical decentralized naming system for the Internet, with scalability and flexibility as key design goals. DNS is used for address resolution, i.e., hostname to IP mapping (e.g., "www.example.com" resolves to IPv4 address "1.2.3.4"), as well as the inverse mapping. It is also used by such services as email (DNS records enable a search for mail servers) and blacklisted email hosts.

A typical attack against DNS consists in flooding a DNS server by a huge number of queries, leading to a Denial of Service i.e., the server cannot handle the load and thus does not respond to legitimate queries. A more subtle attack consists in poisoning or spoofing a DNS cache. Indeed, when a system queries a DNS server and receives an IP address as a response, it saves this information in a local cache for a given period of time, such that the system can answer a new similar query without having to retrieve the information from the server. If the cache is compromised, then anyone who uses it may be misguided to a fraudulent site.

Being one of the cornerstones of the Internet, the security of DNS is essential and security services have been added under the name "Domain Name System Security Extensions" (or DNSSEC) in 2005. DNSSEC enables any host to be confident in the address resolution results, the IP address (or any information returned by a DNS query like mail servers).

However, DNSSEC does not address confidentiality requirements as will be discussed in §5.3.6.

2. https://ietf.org

Attacks against the Border Gateway Protocol (BGP) interdomain routing

Each IP datagram (or packet) needs to be "routed" through the Internet, from its source to its destination (sometimes multiple destinations). This operation is done hop-by-hop by routers: for each incoming packet a router finds a path and forwards the packet to next router until it reaches its destination. Finding a path within a router is the goal of routing protocols, and these routing protocols leverage a distributed database that contains routing and reachability information. Two types of routing protocols exist: some protocols are meant to operate within Autonomous Systems (or AS), controlled by a single organization (e.g., a university), while others are meant to operate at the interconnection level, between autonomous systems, i.e., at the Internet backbone level. The Border Gateway Protocol (BGP) is the protocol currently in use on the Internet for the exchange of routing and reachability information among autonomous systems. It is therefore of utmost importance since any misbehavior, perhaps caused by an attack, can isolate a whole country from the Internet, or redirect all of a given country's traffic through a surveillance point[3].

BGP has long suffered from security weaknesses. For example, an attacker can forge a false BGP response that will allow him to hijack further traffic. Addressing these weaknesses requires:

- setting up a dedicated public key infrastructure to distribute certificates that can be verified by any BGP router. This is the role of the Resource Public Key Infrastructure (RPKI), managed by the various Internet registries (IANA and Regional Internet Registries);
- leveraging on this RPKI to issue certificates, called Route Origination Authorization (or ROA), that attest that an AS controls certain IP address ranges and is authorized to originate route advertisements for these IP address ranges.

Although these mechanisms are required, an ROA by itself does not prevent an attacker (e.g., a malicious BGP router) to forge or propagate malicious route announcements. The underlying problem of validating an AS path for a specific destination is complex and multidimensional: it requires to check the AS validity, the AS neighborhood, the compliance of the AS path listed in the message with the message propagation itself, and the compliance of the AS path with the actual routing policies of each AS. The BGP security extension BGPsec *(see below)* is meant to provide some guaranties from this point of view.

The Secure Inter-Domain Routing (SIDR) IETF working group has specified the RPKI and BGP security extensions. However, deployment takes time, especially with BGPsec that requires major upgrades, and enjoying a fully secure interdomain routing remains a distant dream. Partial deployment being the rule, several academic works focus on the consequences of attacks that originate from unsecured parts of the Internet and techniques to mitigate them.

3. Isolating a geographic area could also be motivated by the desire to disconnect a certain number of servers involved in distributed services and thereby facilitate an attack (e.g., a blockchain-based service.)

Another aspect of BGP is the geostrategic importance of routing information. Their analysis can help identifying or understanding attacks (e.g., when a subset of the Internet becomes suddenly unreachable) or surveillance practices (e.g., when a subset of the Internet traffic is redirected through a certain domain).

Encryption by default and mass surveillance attacks mitigation

"Peeking is irresistible. If there is information visible in the packet, there is no way to keep an intermediate node from looking at it. So the ultimate defense of the end to end mode is end to end encryption." [CWSR02]. Following this wise advice, research and standardization activities have been carried out in order to facilitate the use of encryption within the Internet and to thwart passive eavesdropping in particular.

This is the case of *TCP encryption,* for instance as developed by the TCP Increased Security (tcpinc) IETF working group [4]. The idea behind TCP encryption is to design TCP extensions that are likely to provide unauthenticated encryption and integrity protection of TCP streams. In this case, the unauthenticated key exchange mechanism enables both ends to encrypt and check the integrity of each TCP packet, very easily, without relying on any external service (e.g., PKI infrastructure), nor user solicitation (as with SSL when connecting to a new host), nor any modification to applications for whom this extension is fully transparent. Such an extension, once sufficiently deployed, will enable all TCP flows to be encrypted by default. However, since there is no end-to-end authentication, there is a risk of Man-in-the-Middle attacks: the attacker impersonates the remote node and proposes its own keying material which allows them to decrypt, peek, and re-encrypt all traffic. For this reason, it is viewed as a "better than nothing" security.

Encryption as a default tends to become the rule, as is the case in the new QUIC transport protocol, now developed within the QUIC IETF working group [5]. Initially proposed by Google as a high-performance replacement for HTTP over TLS/TCP connections, this protocol already represents a significant part of Internet traffic (more than 30% of Google's egress traffic, or 7% of Internet traffic, by the end of 2016). Among many innovations, this protocol provides by default secure communications: QUIC packets are always authenticated and the payload is typically fully encrypted, thereby preventing mass surveillance and other forms of attacks.

[Inria teams] Network threats

↗ The **DATASPHERE** team works on the analysis of BGP data to identify or understand attacks or surveillance practices.

[CWSR02]. D. Clark, J. Wroclawski, K. Sollins, and Braden R. Tussle in cyberspace: Defining tomorrow's internet. In *proceedings of SIGCOMM,* 2002.

4. https://datatracker.ietf.org/wg/tcpinc/about/
5. https://datatracker.ietf.org/wg/quic/about/

2.3 The human factor

As in many other fields, there is a well-known adage about security that says that the main threat lies between the chair and the keyboard. This adage may be exaggerated, and at the very least it deserves further study, but it must be recognized that the users are indeed sometimes a source of security problems. Firstly, they can be the target of the attack (see Section 2.3.1). In addition, They can try to avoid using the available protection mechanisms due to the (real or perceived) excessive complexity of their use (see Section 2.3.2). Finally, their level of education and training is too often insufficient (see Section 2.3.3): they are therefore not aware of the real risks, or on the contrary overestimate them. In either case, they do not know what mechanisms are to be used when.

2.3.1 Attacks against the user: social engineering and phishing

[Summary]
Social engineering aims at convincing the user to perform an action, such as revealing a password, by gaining their trust. Closely related, phishing aims at obtaining information like passwords, credit card numbers, etc. It is often based on massive email campaigns (spam) or messages over other communication media (chats, social networks) to request that people provide sensitive information by either replying to an email or connecting to a website.

Social engineering aims at convincing a person to perform a forbidden or sensitive action by gaining their trust. The attacker can impersonate a person's identity or can pretext a fake urgent matter. Social engineering is not strictly limited to the Internet. However, the Internet allows to scale up phishing attacks, a major representative of social engineering attacks. Phishing usually aims at obtaining information like passwords, credit card numbers etc. It is often based on massive email campaigns (spam) or messages over other communication media (chats, social media) requesting that people provide sensitive information either by replying to the email or by connecting to a website. At first phishing campaigns were quite simple and naive because the same email was massively sent without any customization. Spear phishing is more advanced and leveraging higher social intelligence to make people confident in the legitimacy of the request they received. Hence, the request can be customized regarding the country, the location, or the company of the victims. Actually, this kind of information can be easily found on the Internet.

The FBI estimates that money extorted by phishing was about half a billion dollars a year between 2013 and 2016 [6], based on complaints collected in USA. Even though sophisticated techniques can support phishing, such as DNS poisoning to

6. https://www.ic3.gov/media/2017/170504.aspx#fn3

take over a legitimate website, attackers usually prefer relying on social engineering techniques, that are a much easier way to achieve the same end.

Although massive naive social engineering might be easily detected, it is challenging to detect spear phishing even if carried out at a large scale, due to the automatic personalization for each email sent. Defense occurs at several levels. First, phishing attacks rely on a communication channel to reach the victims. For large campaigns, attackers leverage botnets. Therefore, fighting botnets indirectly limits phishing attacks. Second, an analysis of users' messages, the content itself, and the correlation among them, is another option. Third, most of the time, the victim is redirected to a website. Detecting or preventing such websites to exist or to be accessed is also an effective counter-measure. Both the second and third approaches are relevant to social engineering as, in these cases, the attacker needs some social intelligence to make the victims confident in the legitimacy of the email or the website.

Analyzing message content of an email is problematic due to privacy concerns. In addition, with the use of encryption, only end-users or their email servers may have access to the content, making impossible the correlation of multiple emails sent to different users. Hence, only individual email analysis is possible. That is why blacklisting domain names or URLs is widely used. Moreover, blocking access to a single such website means protecting thousands of users in one go. The main challenge is actually to block early a phishing website or URL and even predict it when possible. Indeed, the attacker needs to setup its infrastructure before effectively starting their campaign. Of course, websites can only be analyzed once a potential URL is known. More active and reactive techniques are thus required to automatically discover and verify potential phishing websites.

[Inria teams] The human factor

↗ The **RESIST** team works on automatic blacklist construction of potential phishing websites. This work includes a reactive approach and a proactive approach.

2.3.2 Improving security mechanism usability

[Summary]

One of the main sources of computer security failures are still human errors. A major reason for this is that user interfaces of many applications or software systems have often not been designed for the purpose of security. A good user interface for security should consider that the user is rarely a security expert; it should always ensure that the user is well aware of the consequences of their actions and be designed to avoid unintentional errors that compromise security. Designing security mechanism with good usability calls for interdisciplinary research with experts in cognitive sciences.

Human errors are a major source of computer security failures. While increasing the users' awareness of security risks is crucial, supposing that users are well-trained is neither realistic, nor sufficient. One major reason for human errors is that user interfaces of software systems are not well designed for the purpose of security, even if design principles that are suitable for other applications are applied. Indeed, security is not the user's main goal unlike browsing the web, or purchasing goods and services. Ideally, users want security to work without them doing any specific actions; often it is easier to ignore or circumvent security mechanisms if this makes achieving their main goal easier. A classic example is a web site with a bad certificate; simply ignoring the warning allows a user to continue browsing, at the risk of visiting a fake, phishing website. Another example is deactivating the whole firewall, to ensure a given port used by a specific application is not blocked. Often users do not understand, or misunderstand, the consequences of their actions. Therefore, security mechanisms must be designed with usability in mind. The user interface should consider that the user is not a security expert (and does not understand, for example, the underlying mechanisms of access control), ensure that the user is well aware of their actions' consequences, and avoid that user make errors that compromise security. Designing usable security mechanism calls for interdisciplinary research with experts in cognitive sciences.

[Inria teams] A usability to be improved

↗ The **CIDRE** team works with ergonomists and psychologists to evaluate how a user is perceiving the importance of a message, to increase the chance that the user is clearly aware of their action.

[Research challenge 2] Security and usability

Very often, when users request a service, they are willing to sacrifice security, and bypass an annoying security mechanism, if it prevents them from using the service. In order to avoid this problem, security must be as transparent as possible. Even though complete transparency is not always possible, security services must be as simple as possible to use. Work is needed to propose interfaces and security mechanisms that are suitable for nonexpert users, that ensure the user is well aware of the consequence of their actions, and that prevent users from making errors that compromise security. Designing such usable security mechanisms calls for interdisciplinary research including typically experts in cognitive sciences.

2.3.3 A lack of education and awareness

[Summary]
Users are often viewed as the weakest link in the security chain, too often being unaware of security issues and therefore easily deceived by even very simple attacks. This is why the education and awareness of "good practices" and "cyber hygiene strategy" of every computer user (at work or in the home) is crucial. Each young citizen should be introduced to the basics of computer science and cybersecurity. Each professional actor should know about the risks related to economic intelligence and cyberattacks, and be informed of the possible defenses. In addition, system and network administrators should follow periodical training to be kept up to date regarding the most recent threats—and the solutions to mitigate them.
The industry needs experts on cybersecurity, although it is facing a shortage of skilled cybersecurity experts at all levels. Even if cybersecurity curricula are now offered by a large number of institutions, efforts must still be made to train more experts.

Users (citizen or professional actor) are impacted more or less consciously by cybersecurity or the lack thereof. Too often a victim of cyberattacks, they are considered by many as the weak link of the chain, being unaware of security issues and therefore easily deceived even by technologically very simple attacks such as phishing or social engineering. Insufficiently informed of the importance of protecting themselves, they may for example use passwords that too weak or too predictable. Finally, a given functionality may be more important for them than security, thus they tend to systematically click on a web link or use an unsigned certificate. Thus, an important challenge is to make security tools more easily usable (see §2.3.2).

More generally, most successful attacks leverage well-known security problems and a vast majority of cyberattacks are the result of poor cyber habits within the victims' organizations.

In this context, the education and awareness of each user is essential. Users must be educated about "good practices" for domestic and professional situations and should be able to enforce a "cyber hygiene strategy" in order to reduce the risks of becoming a victim or spreading an attack.

As a first step, each young citizen should be introduced to basic security concepts and tools, at the same time when being introduced to the basics of computer science. Examples of topics include the importance of updating the software or the operating system to avoid keeping too many vulnerabilities, the importance of an antivirus, the definition of a good password or an electronic signature.

A second step is to educate each professional actor, in each educational program, whatever the domain, introducing additional good practices relative to the professional context. Each professional should be explained the economic

intelligence risks, be taught about the separation of professional and personal data and applications, about the partitioning of networks, and about the importance of backups and business continuity plans. In France, a good example of what could be done is ANSSI's (Agence Nationale de la Sécurité des Systèmes d'Information) CyberEdu project [7] that aims to develop pedagogical material facilitating the integration of digital security in higher education for non-specialists.

A third step targets the operational teams, especially, but not only, within small and medium sized businesses. These teams must be aware of the state of the art in cybersecurity and must therefore follow periodical training on recent risks and threats, on the fine-tuning of systems, on security reinforcement, on maintaining a global safe condition, and on legislation and regulation. Several initiatives are following this path: in France, ANSSI has produced the "40 Essential Measures for a Healthy Network" guide [8] dedicated to people in charge of information systems security. At EU level, the ECSO "Cybersecurity Human Resources Network" is looking at raising the awareness levels through various cyber hygiene initiatives, and the ENISA issued its own document [9].

Finally, the education of cybersecurity professionals is of course a main issue. Today, the industry already faces a shortage of skilled workers in cybersecurity, as proved by the number of job offers published on APEC [10] for cybersecurity positions, which in France has quadrupled from 315 offers to 1,133 offers [11] between 2014 and 2016. Worldwide, the Frost & Sullivan business consulting firm forecasted a lack of 1.5 million professionals by 2020, according to the 2017 Global Information Security Workforce Study [12].

Addressing this problem represents a long term effort in education and training. The inherent difficulty is that cybersecurity requires both an extremely solid background in computer science, together with additional skills relative to the threat landscape, security concepts and tools, and a good understanding of law, human factors and psychology, social sciences, economics, and risk management.

Technical curricula are now proposed by a large number of institutions. In France, ANSSI launched SecNumEdu [13], whose objective is to certify some curricula to provide assurance to students and employers that training in the digital security field meets criteria defined by ANSSI in collaboration with industry and higher education institutions. A first list of programs are already labeled [14]. A similar job

7. https://www.ssi.gouv.fr/entreprise/formations/cyberedu/
8. https://www.ssi.gouv.fr/uploads/IMG/pdf/guide_hygiene_v1-2-1_en.pdf
9. https://www.enisa.europa.eu/publications/cyber-hygiene
10. https://www.apec.fr/
11. http://www.bretagne.bzh/upload/docs/application/pdf/2017-06/etude_apeccybersecuritebretagne.pdf
12. https://iamcybersafe.org/wp-content/uploads/2017/06/Europe-GISWS-Report.pdf
13. https://www.ssi.gouv.fr/entreprise/formations/secnumedu/
14. https://www.ssi.gouv.fr/particulier/formations/secnumedu/formations-labellisees-secnumedu/

has been done by the NATO mainly for US and UK universities[15] and the ENISA for EU universities[16]. One can also cite at the European level, the EIT Digital Activities in Cyber Security that includes Professional[17] and Master schools[18].

The interdisciplinary nature of cybersecurity may require novel teaching methods and strategies. Here, practical (and not just theoretical) learning is crucial. Programs could also take advantage of MOOCs and specific technical professional training such as large scale Cyber Defense Exercises (CDXs). An example of the latter is NATO's Locked Shields 2017[19].

2.3.4 Manipulating users and public opinion

[Summary]
Data and technologies can be used to influence opinions or decisions online, through profiling and well-targeted *fake news.* This requires technical solutions to deter unwanted profiling and enable easy verification of information as well as validation of its source.

A serious issue, actually only loosely connected to security but very often linked to security, at least in the minds of the general public, is that data and technologies are also used to motivate, influence, or shape people's opinions or decisions online. The better understanding of users' behaviors combined with the capacity of building accurate psychological profiles create opportunities to develop techniques that influence users online, by shaping their opinions. These technologies try to impact the user's reasoning and decisions by manipulating their "cognitive biases," e.g., their emotions, memory, or beliefs.

One particular way to manipulate public opinion is fake news. Globally speaking, disinformation through the Internet is seen today as a major issue that requires technical solutions to enable easy verification of information as well as validation of its sources. In the media industry, a number of recent initiatives have started to identify, track, and debunk fake news and wrong claims that circulate over the Internet, for instance creating a shared repository of fake news or lists of trusted and non-trusted information sources *(see, e.g., the Decodex database*[20] *from Le Monde in France).*

15. https://digitalguardian.com/blog/cybersecurity-higher-education-top-cybersecurity-colleges-and-degrees
16. https://resilience.enisa.europa.eu/nis-platform/shared-documents/wg3-documents/cybersecurity-education-snapshot-for-workforce-development-in-the-eu/view
17. https://www.eitdigital.eu/eit-digital-academy/professional-school/
18. https://www.eitdigital.eu/eit-digital-academy/master-school/
19. https://ccdcoe.org/locked-shields-2017.html
20. http://www.lemonde.fr/verification/

[Inria teams] Manipulating users and public opinion

↗ The **PRIVATICS** team investigates how personal data is being used to manipulate people online.

Inria, together with partners in the media industry as well as in the defense sector, studies how to fight against user manipulation and fake news, by leading projects on fake news detection.

↗ At the content analysis level, the **LINKMEDIA** team investigates image retrieval and tracing as well as image forensics, in conjunction with text mining to help in the detection and tracking of fake news.

↗ The **ALMANACH** team applies natural language processing techniques to identify fake news.

↗ At the data and knowledge management level, the **CEDAR** and **GRAPHIK** teams work on easy access to heterogeneous data sources to facilitate information cross-checking and validation in complex environments.

↗ The **DANTE** team focuses on graph analytics to identify the networks of diffusion for fake news and the accounts that legitimate fake news.

↗ Visualization teams such as **ILDA** study the mechanisms needed for human analysts working in a collaborative and dynamic fashion to make the most of data, its related processing, and management algorithms, in an effort to increase information security in the media industry and beyond.

2.4 Modeling threats and attacks with attack trees

[Summary]

To represent all kind of attacks, one can use a graphical representation called "attack trees." In this representation, each leaf of the tree expresses a step that the attacker has to perform in order to carry out their attack. Each non-terminal node contains a label expressing how its children are connected (and, or, sequence). Globally, an attack tree represents the sequences of possible actions that the attacker can perform to reach their objective. Attack trees are largely used during the risk analysis step: risks (threats) are identified, a security policy is designed, and then security mechanisms are chosen to enforce that policy.

In the safety field, so called "fault trees" have been proposed in the early 80's [VGRH81] to graphically represent the safety risks a system faces. In 1999, Bruce

[VGRH81] William E. Vesely, Francine F. Goldberg, Norman H. Roberts, and David F. Haasl. Fault tree handbook. Technical report, Nuclear Regulatory Commission Washington DC, 1981.

Schneier adapted this representation to the security field, introducing "attack trees." [Sch99]

As with any tree, an attack tree is composed of nodes and leaves. A node's label (AND or OR) expresses how the children of the node are connected. Each leaf expresses a step that the attacker has to perform in order to carry out their attack. To be able to express more subtle links between the different steps of an attack, more evolved operators have also been introduced, such as the "Sequential AND" that imposes a time order between the operators of a logical AND.

Common attack trees bring a qualitative representation of an attack. Some research works to enrich attack trees by adding quantitative attributes that include the impact of the attack, the costs of countermeasures, etc.

The semantics of attack trees have been largely extended over the last decade. One of the main extensions considers not only the description of an attack, but also of the actions that the security administrator can carry out to stop its progression. This class of trees are called attack-defense trees [KMRS10]. The originality of this representation relies on the fact that the leaves are heterogeneous and represent both the attacker and the defender perspectives. The link between attacker and defender views has also been investigated in the domain of alert correlation.

Due to the difficulty of building attack trees, various attempts have been made to generate them automatically.

Many other graphical modeling techniques have been proposed, from small variations on the original attack trees to changes of representation, as in attack graphs, Bayesian attack graphs, or Petri Nets. For more detailed information on the various graphical models that are used in security to model an attack and their subtleties, the reader should refer to [KPCS14].

[Inria teams] Modeling threats and attacks with attack trees

↗ The **CIDRE** team proposes a solution to automatically build correlation rules starting from existing attack trees and the description of the monitored systems.

↗ The **DIVERSE** team works on the synthesis of attack trees for supporting computer-aided risk analysis.

↗ The **PRIVATICS** team uses attack trees for the quantification of privacy.

↗ The **TAMIS** team uses attack trees for the quantification of security.

[Sch99] Bruce Schneier. Attack Trees: Modeling security threats. *Dr. Dobb's Journal,* 24(12):21-29, December 1999.
[KMRS10] Barbara Kordy, Sjouke Mauw, Sasa Radomirovic, and Patrick Schweitzer. Foundations of Attack-Defense Trees. In *Formal Aspects of Security and Trust,* Lecture Notes in Computer Science, pages 80-95. Springer, Berlin, Heidelberg, September 2010.
[KPCS14] Barbara Kordy, Ludovic Pietre-Cambacedes, and Patrick Schweitzer. DAG-based attack and defense modeling: Don't miss the forest for the attack trees. *Computer Science Review,* 13(Supplement C):1-38, Novembre 2014.

Cryptographic primitives, schemes and protocols

Cryptography aims at providing techniques and tools to secure communications, even in the presence of an adversary. Historically, the main goal of cryptography was to ensure the confidentiality of messages through *encryption*, i.e., information remains hidden to non-authorized people. Early methods of encryption were generally rather naive, such as Caesar's cipher which consists in shifting each letter by a constant (e.g., replacing 'A' by 'D', 'B', by 'E', and so on) and can be easily broken using techniques such as a frequency analysis. A substantial advance occurred during World War II with the Enigma rotor machine used by Germany. Breaking the Enigma cipher required significant effort and resources. Nowadays, cryptography is based on firm mathematical grounds and aims at guaranteeing many more properties than just confidentiality: cryptography provides tools to protect the integrity and authenticity of messages (avoiding for instance that the amount in a financial transaction is changed), to ensure non-repudiation (the sender cannot deny being the author of a message) and anonymity. Moreover, these tools may be combined to ensure even more complex goals. The aim of *cryptanalysis* is to "break" the cryptographic techniques that are meant to ensure security and is actually used to check their robustness.

Close-up of an Enigma rotor assembly. – TedColes via Wikipédia, CC0

The chapter is structured in three parts:

- cryptographic *primitives* are the most basic building blocks; such primitives allow to encrypt or digitally sign a message;
- cryptographic *schemes* generally build on primitives to provide stronger security goals, guaranteeing the integrity and authenticity of arbitrarily sized messages;
- cryptographic *protocols* rely on schemes to achieve more complex security goals, e.g., establishing a secure communication channel that may be used for confidential and authenticated message exchanges.

3.1 Cryptographic primitives

[Summary]
Cryptographic primitives, such as encryption functions and digital signatures, are the basic building blocks for designing secure systems. They can be divided in two families: symmetric and asymmetric cryptography. Symmetric cryptography supposes that communicating parties privately share a secret key. This kind of cryptography is more efficient than asymmetric cryptography, but requires a preliminary secure key exchange. Asymmetric, or public-key, cryptography does not require sharing a secret key, as the public key (used for encrypting or verifying signatures) does not need to remain secret. The use of these two kinds of primitives is complementary. A typical approach is exchanging a symmetric key using asymmetric cryptography in order to encrypt later communications more efficiently with symmetric cryptography using the exchanged key. Hence, both types of cryptography are required in most applications.

Today we have mature constructions for both symmetric and asymmetric cryptography. Nevertheless, there is still a need for research in both cryptanalysis and design. The aim of the cryptanalyst is to find weaknesses and assess the strength of existing constructions. On the one hand, this work consists in carefully assessing the difficulty of solving the underlying assumed algorithmically hard problems, considering both the evolution of computing power and the improvement of algorithms. On the other hand, this work must also explore new attack methods, such as attacks relying on a possible quantum computer or attacks that exploit side channels. The design of new primitives may be driven by (foreseeable) cryptanalytic breakthroughs (e.g., due to the construction of a quantum computer) or a new demand or need from industry. New demands include lightweight cryptographic schemes that can operate on low-power devices, as well as new functionalities, such as the need for functional or homomorphic encryption that permit computations on encrypted data, typically required when computation on sensitive data is outsourced, which is a current trend.

Cryptographic primitives come in two different flavors. On the one hand, the primitives that rely on a unique secret shared by all the protagonists are known as secret-key cryptography, also called symmetric cryptography. This family also includes cryptographic hash functions: as any hash function, a cryptographic hash function maps an arbitrarily sized value to a fixed size one, but additionally ensures properties such as being a one-way function (it is difficult to find the pre-image of a hash value) or collision resistance (it is difficult to find two values that map to the same hash value). Hash functions do not rely on any secret but are based on similar design principles as symmetric encryption. On the other hand, the primitives that use asymmetric keys (as the signing key and the verification

key in signature primitives) are gathered together under the name public-key cryptography, also called asymmetric cryptography.

Symmetric primitives are very efficient in both software and hardware implementations, and are thus usually well-adapted to constrained environments. Conversely, asymmetric primitives require heavy mathematical tools to be implemented, and their efficiency is several orders of magnitude worse than symmetric cryptographic primitives. However, they do not require a previous agreement between users. Therefore, symmetric and asymmetric primitives offer complementary security features, so they often have to be associated in concrete applications to achieve both appropriate security properties and efficiency requirements. For instance, in most applications, data is encrypted by an efficient symmetric encryption scheme under a secret key. This (short) secret key, which needs to be shared by the users, is either transmitted with an asymmetric encryption scheme or obtained by a key exchange protocol (e.g., Diffie-Hellman key exchange).

3.1.1 Cryptography today

Most of the basic cryptography needs are met today with a relatively small number of standardized primitives, namely AES [DR02] (together with some mode of operation like CTR or GCM) for symmetric encryption and SHA-2 and SHA-3 for cryptographic hashing. For digital signature, key exchange, and public-key encryption, most primitives, like RSA [RSA78] or Diffie-Hellman key exchange [DH76], are based on number theory assumptions (e.g., computational hardness of factoring or computing discrete logarithms).

WHAT ARE THE ISSUES?

In addition to the exploration of new features (e.g., functional encryption), one of the main tasks of cryptographic research is to maintain confidence in the existing primitives and to devise new ones whenever necessary. This confidence stems from the understanding of the threats and their evolution due to new cryptanalytic methods or to technological breakthroughs. Therefore, existent primitives must be continuously scrutinized in order to check that all these threats are adequately addressed.

Cryptanalysis tells us when and why primitives must evolve (e.g., with larger keys, more rounds) or be replaced (e.g., due to algorithmic or technological breakthroughs). It provides the designer with evaluation tools and design criteria

[DR02] J. Daemen and V. Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard.* Information Security and Cryptography. Springer, 2002.
[RSA78] R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM,* 21(2):120-126, 1978.
[DH76] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Trans. Information Theory,* 22(6):644-654, 1976.

that help to adapt existing cryptographic primitives or propose new ones that are resistant against the new attacks.

3.1.2 Cryptanalysis

The aim of academic cryptanalysis is to understand the threats to security of the existing primitives in order to be ahead of malicious adversaries. One difficulty is that the threats may evolve over time with the progress of algorithms, mathematics, or computers (e.g., Moore's law or quantum computing), but the attacker's capabilities evolve as well (e.g., physical access to an implementation, which was not always considered, must now be taken into account in the case of lightweight primitives for IoT). Cryptanalysis is the foundation of confidence we have in these primitives: the more we analyze them, the more we can trust them. It provides an empirical measure of security thanks to a thorough and never-ending scrutiny, searching for possible weaknesses. The knowledge of state-of-the-art cryptanalysis is thus the backbone for the design of secure primitives. In what follows we will distinguish mathematical cryptanalysis, focusing on the design, and implementation cryptanalysis, exploiting particular implementation details. We will also distinguish classical, as opposed to quantum cryptanalysis.

CLASSICAL CRYPTANALYSIS

Cryptanalysing a cryptographic primitive consists in solving a problem that is claimed to be hard in the security model (e.g., recover the secret key, decrypt or forge a signature without the key, find a collision). The attack succeeds if it is more efficient than expected from the security claims. Cryptanalysis usually only succeeds in breaking reduced versions of the primitive (e.g., smaller key, smaller block size, fewer rounds). The security margin of a given primitive is then quantified by how much the reduced versions differ from the original primitive.

The state-of-the-art cryptanalysis is actually the only security criterion to decide at what point a primitive must evolve or be replaced. For instance, after a 10-year cryptanalytic effort by several research teams, a collision has recently been found for the SHA-1 hash function[1], that is now considered unsafe from a cryptographic point of view and is slowly being replaced in applications. Another recent spectacular cryptanalytic breakthrough, strongly affecting the security of some pairing-based cryptosystems, is a quasi-polynomial algorithm for computing discrete logarithms in finite fields of small characteristics.

As modern cryptography advocates for security by knowledge, i.e., having a public design rationale, including justification of the design choices, public cryptanalysis is the only way of being able to trust such primitives. Therefore, any

1. https://www.inria.fr/centre/saclay/actualites/sha-1-les-predictions-d-inria-verifiees

standardization effort is preceded by a long and intensive cryptanalysis phase that is necessary to evaluate the future standard. Moreover, analyzing new proposals often leads to the definition of new attacks that could also be considered as new threats to existing primitives.

[Highlights] Cryptanalytic record computations

Inria has been a key actor in several record computations for cryptanalysis:

↗ In 2010 a 768-bit RSA key has been factored. The computational effort was about 1500 CPU years and 2 calendar years. This record computation was led by EPFL and Inria's **CARAMBA** team. At about the same time, credit card keys were increased from 896 to 960 bits and ANSSI released a recommendation to use 2048-bit RSA keys in 2010.

↗ In 2013, researchers from the **CARAMBA** and **OURAGAN** teams in collaboration with A. Joux designed an efficient algorithm to break discrete logarithms in finite fields of characteristic 2, as well as pairings for algebraic curves based on binary finite fields.

↗ Since 2014, researchers from **CARAMBA** and **GRACE** have solved discrete logarithms in several finite fields of the GF(p^k) for $k \leq 6$.

↗ In 2015, the FREAK attack (**PROSECCO** team) highlights how factorizing 512-bit RSA keys (in combination with a common implementation bug) can be used to break TLS connections affecting about 25% of the web. This work led to fixes in major web browsers and websites.

↗ In 2015, the LogJam attack on TLS (**PROSECCO** and **CARAMBA** teams) highlights that using a pre-computation for a specific 512-bit group, Diffie Hellman keys can be effectively broken and (the still widely used) keys in 768-bit groups are now within reach of academic research teams.

↗ In 2016, the SLOTH attack (**PROSECCO** and **SECRET** teams) highlights how hash collisions in MD5 and SHA-1 can be used to break signature-based authentication in protocols like TLS. This work led to the deprecation of MD5 and SHA-1 in TLS 1.3.

↗ In 2016, researchers from the **CARAMBA** team, in collaboration with colleagues from the University of Pennsylvania, showed how to solve Diffie-Hellman for specially crafted prime numbers. As a result, a few months later, an RFC (Request For Comments) that did not provide details about the generation of the parameters was withdrawn.

QUANTUM CRYPTANALYSIS

After a couple of decades of study mostly limited to the academic world, quantum computing is now at the center of a race between high-tech companies like Google, Microsoft, or IBM. While the prospect of a large universal quantum machine is arguably still many years away, dismissing its potential impact for cryptography has become a rather untenable position. Indeed, the American National Institute of Standards and Technology, NIST, has recently launched a call for quantum safe cryptosystems that would resist attacks implemented with the help of a quantum computer.

Common wisdom in the field says that public key cryptography relying either on the difficulty of factoring or discrete logarithm, e.g., RSA or elliptic curves, is irremediably broken in a quantum world because of Shor's algorithm [Sho]. Symmetric cryptography looks at first much more immune to quantum cryptanalysis since the main applicable speed up seems given by Grover's algorithm in exhaustive search, a task for which quantum computers only provide a quadratic advantage, i.e., the cost of an exhaustive search drops from N to $\sqrt{N}$ where N is the number of keys. In particular, one could naively think that doubling the secret-key size is sufficient to take care of quantum attacks against symmetric cryptography. Unfortunately, this assessment is not backed up by the many years of effort required to gain confidence in the security of cryptosystems through dedicated cryptanalysis. New results have recently appeared in this direction, for instance showing that, in certain models, Simon's quantum algorithm for period finding completely breaks the security of the most widely used modes of operation (such as CBC-MAC or GCM) for authentication and authenticated encryption.

The lesson is that quantum cryptanalysis deserves much more attention than it has received so far. Standards take many years to evolve because of new threats. Therefore, a better understanding of the impact of quantum computers on cryptography is definitely called for now in order to see how quantum attacks should be modeled, to understand how secure post-quantum solutions based on lattices or codes really are, and more generally to integrate quantum techniques in the toolbox of cryptosystems designers.

IMPLEMENTATION CRYPTANALYSIS

The security of a cryptographic primitive relies both on the software and the hardware that implements the primitive and the hardware the code runs on. The attacker of cryptographic primitives may just have access to the software, e.g., by remotely running some other process on the same machine, or may also have access to the hardware and be able to measure some physical quantity during execution, such as the precise timing or voltage, or even perform fault injection. However, the distinction between software and hardware is not so significant for the cryptographers who establish a model of the threats that is usually abstracted over the low-level details and the differences between software and hardware. A more practice-oriented discussion of this kind of attacks is given in §2.1.

When a cryptographic algorithm is implemented on a physical device (a smart card, a cell phone, a computer, ...), an adversary can measure the physical properties of the system during the execution of the cryptographic algorithm. The accuracy of these measures depends on whether the attacker has physical access to the hardware. In this case, they are able to measure the variations of

[Sho] P. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *IEEE FOCS'94*.

the power consumption or of the electromagnetic radiations during execution. They may even be able to perform fault injections. If the attacker just has access to the software, e.g., by remotely running some other process on the same device, they may obtain only some timing information. However, in both cases, these physical measures are correlated to the secret key manipulated by the system and the key can be efficiently extracted unless special countermeasures are used. An important research area thus focuses on evaluating the effectiveness of these attacks and devising effectual countermeasures. In particular, we now have a good understanding of the protection offered by masking, using techniques from multiparty computations to split the secret into shares that are not correlated to the actual secret.

Physical attacks also include fault attacks, where an attacker manipulates the circuit to induce a fault (e.g., by tampering with the power supply or with a laser shot) and exploits the difference between a normal output and a faulty one to recover the secret key. Most implementation attacks require physical access to the system, but attacks are also possible when an adversary can just run code on the same machine as the victim. In particular, cache attacks and fault attacks based on the Rowhammer technique [KDK+14] can be very effective and have been demonstrated from JavaScript code running in a web browser.

3.1.3 Design

New primitives are designed either after some cryptanalytic breakthrough (e.g., the breaking of most standardized hash functions in 2004 and 2005) or to answer some pressing demand from the industry (e.g., lightweight primitives for low-cost devices). In modern cryptography, new primitives come with design rationale and security arguments. Such arguments do not provide an unconditional guarantee of security and may take different forms. In asymmetric cryptography, the argument will typically ascertain that any adversary breaking the primitive with given parameters will solve a problem that is widely believed to be difficult (e.g., factoring a 2048-bit number corresponding to the product of two primes, generic decoding for a linear code, or finding a short vector in a lattice of given parameters). In symmetric cryptography, the arguments are based on properties of the underlying building-blocks that guarantee (or tend to guarantee) their resistance to known classes of attacks (e.g., linear, differential, or algebraic attacks).

[KDK+14] Y. Kim, R. Daly, J. Kim, Ch. Fallin, J.-H. Lee, D. Lee, Ch. Wilkerson, K. Lai, and O. Mutlu. Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors. In *ACM/IEEE ISCA'14,* 2014.

[Highlights] International Cryptographic Competitions

For more than 20 years, the main new cryptographic standards have been specified after open competitions initiated by standardization bodies or international projects. These competitions attract submissions from many countries, from both academia and industry. The candidates are then scrutinized during several years by the whole cryptographic community in a public security evaluation process. Inria has submitted several primitives to these competitions and has contributed to the evaluation process.

↗ Two stream ciphers were submitted by the **SECRET** team to the eSTREAM project launched in 2004 by the ECRYPT Network of Excellence. One of them, Sosemanuk, has been selected (among 34 submissions) in the final portfolio of recommended ciphers for software-oriented environments[a].

↗ The Shabal and the SIMD hash functions, designed by the **SECRET** and **CASCADE** teams together with several partners, are two of the 64 candidates to the SHA-3 competition[b] launched by the NIST in 2007. Both of them were selected among the 11 semi-finalists of the competition.

↗ The **ARIC**, **GRACE**, **POLSYS**, and **SECRET** teams are involved in the design of 10 (out of 68) candidates to post-quantum cryptography standardization process[c] initiated by the NIST in 2017.

a. http://www.ecrypt.eu.org/stream/
b. https://csrc.nist.gov/projects/hash-functions/sha-3-project
c. https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization

Cryptographic properties of components of an encryption system – © Inria / Photo C. Morel

LIGHTWEIGHT CRYPTOGRAPHY

In the last decades, we have seen a huge increase in the number of low-cost smart devices: e.g., contactless cards, key fobs (RKE), sensor networks, home automation, NFC/RFID tags, or medical implants. Most of them transmit sensitive information wirelessly (e.g., contactless cards are often used for payment, access control, or fee collection) and require cryptography to do it securely. Unfortunately, hardware constraints of embedded devices limit their computing abilities and power budget (being powered by a battery or even passively), preventing the use of conventional cryptography. Therefore, many industrial products use weak homemade cryptography (e.g., MIFARE Classic, KeeLoq, Megamos, Hitag2) or no cryptography at all (e.g., medical implants, wireless mice). To fill this gap, secure lightweight ciphers have been designed to run with low consumption of critical resources (energy, power, execution time). These harsh implementation constraints may have downsides, like a smaller block size forcing to renew keys more often or a low latency. Several designs have been introduced and standardized in the last decades, such as KASUMI (UMTS), PRESENT (ISO/IEC 29192-2), or HIGHT (ISO/IEC 18033-3). Lightweight symmetric cryptography is a very active research area, with several new proposals every year and an ongoing standardization effort by NIST[2]. An important effort is also needed for designing lightweight key exchange and asymmetric primitives. One of the specificities of lightweight cryptography is that, in most applications, the low-cost devices are highly vulnerable to physical attacks. Therefore, lightweight primitives should not only have a lightweight specification but also a secure and lightweight implementation.

In such a constrained environment, randomness generation is also an issue. While random coins are required in most of the cryptographic schemes and also for securing their hardware implementation, they are quite hard to generate, especially at a low cost. Pseudo-random generators allow to expand a random seed into a larger stream of random bits, but both the entropy of the seed and the quality of the expansion have to be studied. Any weakness at any point can weaken the whole system. This is even worse if the adversary can have some control on the device, which could allow them to reduce the entropy of the internal state.

POST-QUANTUM CRYPTOGRAPHY

The availability of quantum computing will render obsolete all cryptographic primitives based on number theory that are routinely and almost exclusively used today for securing communications. Even though quantum computers will not appear before one or more decades, the research community needs to line up now and start preparing alternative primitives notably for key exchange mechanisms and digital signatures. In particular, replacements must be ready soon if one

2. https://www.nist.gov/programs-projects/lightweight-cryptography

wishes to insure long-term confidentiality[3]. An action was initiated by NIST[4] to standardize quantum safe cryptographic schemes (public-key encryption, key exchange, digital signature). The five-year public analysis phase within the NIST standardization process is expected to provide a better view on the security level and performance of these techniques.

The oldest quantum safe technique for asymmetric cryptography, the McEliece encryption scheme, was proposed in 1978 and is contemporary to RSA. Its security relates to the hardness of decoding an arbitrary linear code and is the seminal work of code-based cryptography. Later, multivariate cryptography, relying on the hardness of polynomial system solving, and lattice-based cryptography, relying on the hardness of finding short vectors in a Euclidian lattice, were developed. The past decade has been extremely productive, in particular with the emergence of LWE [Reg] and its much more practical cyclic variants (Ring-LWE). Those techniques are reaching maturity and will assert themselves as a practical alternative for asymmetric cryptography in the coming decade. To be complete, let us mention that other techniques are being considered for quantum safe cryptography, one of the most notable being hash-based cryptography (based on the Merkle tree technique) allowing quantum safe digital signature schemes provided the cryptographic hash function it is built upon is collision-resistant.

[Research challenge 3] Post-quantum cryptography

Building a universal (i.e., as opposed to special purpose) quantum computer is widely believed to become feasible in the next decades. Therefore, it is important to think now about quantum-resistant cryptography, as some information that is encrypted today may still be sensitive in, say, 50 years. Most asymmetric cryptography used today is based either on the hardness of factoring or computing discrete logarithms, these problems are both known to be efficiently solvable by a quantum computer. Hence, there is a need for alternatives: lattice-based, code-based, and multivariate-based primitives are the most prominent candidates. It is urgent to perform an in-depth security analysis of these new schemes.

CRYPTOGRAPHY BASED ON THE LAWS OF PHYSICS

Most cryptographic proposals rely on computational assumptions and are therefore vulnerable to algorithmic and hardware breakthroughs. A way to stop relying completely on those computational assumptions is to use quantum

3. Note that the situation is critical for confidentiality since an adversary may store encrypted documents today that need to be protected during decades, even after a quantum computer becomes available.
The situation is different for integrity which can be enforced by resigning documents when the quantum threat becomes real.
4. https://csrc.nist.gov/Projects/Post-Quantum-Cryptography
[Reg] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *In Proceedings of the 37th ACM Symposium on Theory of Computing* (STOC 2005).

cryptography. Using quantum communication, it is possible to construct an unconditionally secure key distribution protocol (known as the BB84 protocol). This means that even an all-powerful (potentially quantum) adversary cannot break the scheme. This offers a very-long-term security, but can only be used for a limited number of applications such as key distribution because of deployment constraints and is thus usually combined with standard or quantum-safe cryptography. In Europe and in Asia, quantum networks are developed in order to be able to perform unconditional quantum key distribution protocols. Currently, these protocols only work on a limited distance of about 50-150 km. In order to create large scale networks, we require trusted nodes, which can be dangerous from a cryptographic point of view, or quantum repeaters. Quantum repeaters are technologically out of reach today, but seem much easier to build than a full quantum computer and could arrive in a near future.

Another solution to obtain unconditional cryptography is to use other laws of physics. Relativistic cryptography [Kan15], for instance, uses space-time constraints between the agents to perform cryptographic protocols with unconditional security. These constraints limit the possible applications, but, unlike quantum cryptography, can be developed on a large scale with today's technology. It is important to follow the advances of unconditional cryptography, which can be seen as a safe backup plan in case we lose our trust in the computational assumptions used in standard cryptography.

3.2 Cryptographic schemes

[Summary] Cryptographic schemes

Whereas cryptographic primitives are the basic building blocks, cryptographic schemes achieve stronger properties with specific modes of operations. Some applications, such as outsourced computation may also require more advanced functionalities than classical encryption. So-called homomorphic and functional encryption schemes permit working on encrypted data, and cryptographic proofs ("proofs of knowledge") may be used to get evidence that outsourced computation was performed correctly. With the increasing complexity of cryptographic schemes and their security proofs, there has been a recent trend, called computer-aided cryptography, which consists in developing tools to check security proofs and achieve higher confidence in the security of some constructions.

[Kan15] J. Kaniewski. *Relativistic quantum cryptography.* PhD thesis, Centre for Quantum Technologies, University of Singapore, 2015. https://arxiv.org/pdf/1512.00602.pdf.

The security of a block cipher, such as AES, or of a trapdoor one-way function, such as the RSA function, does not generally provide secure encryption schemes on their own. For example, these basic functions are not randomized and encrypting twice the same will yield two identical ciphertexts, hence leaking the information that the two plaintexts were identical. Encryption schemes or signature schemes are thus usually defined by a primitive together with a mode of operation that specifies how to use the primitive in order to accommodate arbitrary-length messages and to reach a specific security goal. Examples of these constructions include standardized block-cipher modes for symmetric encryption (e.g., CBC, CTR) or for authenticated encryption (e.g., CCM, GCM), some padding schemes for asymmetric encryption (e.g., OAEP [FOPS01]) or for signatures (e.g., PSS). The security level offered by such constructions is evaluated under the assumption that the underlying primitive has an ideal behavior. The aim is to guarantee that a given scheme is secure as long as no specific weakness has been identified for the primitive. There are two complementary approaches to analyze the security of a mode of operation: the search for generic attacks (i.e., independent of the underlying primitive) provides upper bounds on the security level, while security proofs provide lower bounds.

3.2.1 Provable constructions

The area of provable cryptography, initiated in the seminal work of Goldwasser and Micali [GM84], is rooted in computational complexity theory. Adversaries are modeled as arbitrary probabilistic polynomial time Turing machines, that have access to oracles (for encryption or decryption). Security is then typically expressed as the adversary's inability to distinguish (with probability significantly better than 1/2) whether they have access to an encryption oracle or to a function that always simply returns a random string. Proofs are then performed by reduction: an adversary who can win such an indistinguishability game can be used to efficiently construct an adversary who can invert the underlying one-way function or distinguish the underlying block cipher from a random permutation. Hence, the proof essentially shows that breaking the construction is as hard as solving the underlying problem that is supposed to be hard. Such results have been obtained for classical modes of operation used to achieve authenticated symmetric and asymmetric encryption. However, there is still an important line of research on the design of new efficient modes of operation for symmetric encryption with a high security level. Indeed, most block-cipher modes of operation have their security limited by the so-called birthday-bound: they become insecure if the number of

[FOPS01] E. Fujisaki, T. Okamoto, D. Pointcheval, and J. Stern. RSA-OAEP is secure under the RSA assumption. In *Proceedings of Crypto '01*, volume 2139 of LNCS, 2001.
[GM84] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2), 1984.

calls to the underlying block cipher is close to 2n/2 where n is the block size. This is a major issue for lightweight cryptography since block ciphers operating on 64-bit blocks are preferred in these applications. This implies that the amount of data that can be encrypted under the same key must be much lower than 32 GBytes. This issue has been recently demonstrated on the CBC mode of operation used in HTTP over TLS and OpenVPN in an attack named Sweet32[5] that led the NIST to lower the limit before rekeying 3DES to 8 MBytes. Designing an efficient mode of operation with a higher security level is therefore an important open problem.

While this approach has been widely used for proving the security of public-key encryption schemes and signature schemes under well-known assumptions, it is now required for any new constructions, and namely for the advanced constructions such as fully homomorphic encryption and functional encryption.

3.2.2 Homomorphic and functional encryption

With the development of outsourced storage and computing, classical encryption is not enough: when encrypted, the privacy of the data is guaranteed, but no processing can be performed on the data. In 2009, Gentry [Gen09] proposed the first encryption scheme that allows homomorphic operations: from the encryption of two messages, it is possible to produce the encryption of the sum or of the product, without any secret information. More concretely, it is possible to send encrypted data to the cloud, and let the cloud evaluate a circuit on this encrypted data. The cloud can then send back the encryption of the result, without having learned any information about the result. The owner of the decryption key can then decrypt the result. Whereas the initial construction of fully homomorphic encryption was prohibitive at both the computational and communicational levels, many improvements have recently been proposed, and some small circuits can now be concretely evaluated.

However, the result output by the cloud is still encrypted under the same key as the inputs, thus it can only be shared with those who could already decrypt the inputs. This is the reason why, Boneh, Sahai, and Waters [BSW11] proposed the notion of functional encryption: an authority can distribute functional decryption keys that help to compute the evaluation of a given function on the plaintext. The keys do however not leak the full input, but only the result of the computed function. The authority initially encrypts the data under a master key and derives a functional key kf for a chosen function f. The function f is evaluated on the plaintext data when decrypted with kf. This allows, for example, to do some aggregation on data (statistical analysis) without revealing the data.

5. https://sweet32.info/

[Gen09] C. Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st ACM Symposium on Theory of Computing (STOC'09)*, 2009. http://portal.acm.org/citation.cfm? id=153641.1536440
[BSW11] D. Boneh, A. Sahai, and B. Waters. Functional encryption: Denitions and challenges. In *Proceedings of Theory of Cryptography Conference (TCC'11)*, 2011. http://ia.cr/2010/543..

While classical techniques (discrete logarithm) allow to instantiate functional encryption for simple families of functions, such as the inner-product [ABDP15], it seems that lattice-based techniques will be required for advanced functions. Similarly, ElGamal encryption is (simply) homomorphic, but to achieve fully homomorphic encryption, lattice-based encryption, or some approaches with noise/error look necessary. Hence the huge activity on lattice-based and code-based cryptography.

[Research challenge 4] Computing on encrypted data

The need for computing on encrypted data has emerged, in particular, with the appearance of the cloud and outsourced computation. In cryptography, this problem can be solved using homomorphic or functional encryption. Gentry showed in 2009 in his breakthrough paper that it was indeed possible to construct a fully homomorphic encryption (FHE) scheme. However, this construction remained theoretical and was completely impractical due to its poor performance. Since then, significant progress has been made on FHE schemes, achieving approximately a still very low speed of 50 logical gates per second. Significant progress will have extremely useful applications for privacy preserving cloud computing, where any technical advance may quickly be exploited as an economical advantage.

3.2.3 Proofs of knowledge

For verifiability of outsourced computations performed by the cloud that we do not trust, proofs are required to convince the user of the cloud's honest behavior. There has been a huge activity in zero-knowledge or witness-indistinguishable proofs for privacy concerns. However, in the context of outsourced computing, the soundness and the succinctness of the proof are the most important issue, since there are no privacy concerns with respect to the user, hence the new SNARG (succinct non-interactive argument) primitive that provides succinct proofs for complex statements.

Still, usual zero-knowledge proofs are of major interest for anonymous credentials and any kind of advanced authentication mechanisms that preserve anonymity. New techniques have recently been developed with the Smooth-Projective Hash Functions (SPHFs).

Also, to limit interactions, Non-Interactive Zero-Knowledge Proofs (NIZKs) have become more effective, first with the Groth-Sahai methodology, making it possible to prove many kinds of relations between committed scalars or groups elements, or more recently with the Quasi-Adaptive NIZKs, based on SPHFs that are more specific but more compact and efficient.

[ABDP15] M. Abdalla, F. Bourse, A. De Caro, and D. Pointcheval. Simple functional encryption schemes for inner products. In *Proceedings of Public Key Cryptography (PKC'15)*, 2015

3.2.4 Computer-aided cryptography

Carrying out reduction proofs in a rigorous way is extremely difficult because they manipulate complex probabilistic algorithms. Indeed, the literature contains many examples of subtle errors in proofs, a famous example being the OAEP construction, "proven" in 1994 with an error discovered in 2001. In a position paper, Halevi [Hal05] advocates the use of proof assistants to verify the correctness of the mundane parts of cryptographic proofs automatically. As for reductions in complexity theory, proofs consist of a creative part for finding the reduction and a mundane, but difficult, part which consists in verifying the correctness of the reduction. In recent years, several tools, including CryptoVerif[6], CertiCrypt, and EasyCrypt[7], have shown that these parts are indeed amenable to automated proof verification. The most mature and versatile of these tools is EasyCrypt: the tool consists in a dedicated interactive theorem prover that shows relational properties on schemes modeled as probabilistic programs. While the tool offers solid guarantees it also requires a high level of expertise. For specific applications, such as chosen plaintext or ciphertext security of encryption schemes, full automation has been achieved with the dedicated ZooCrypt tool, resulting in the verification of numerous schemes and in the design of new ones proven secure. The area of computer-aided cryptography is currently expanding by widening its scope, including security of primitives against side channels, constructions based on pairings, applying these ideas to cryptographic protocols rather than schemes (see §4.3) and improving automation.

[Inria teams] Cryptographic primitives and schemes

↗ The **ARIC** team is working on lattice-based cryptography (LBC). Lattice algorithmics is an established research area that is being revived by LBC and by the new tools and concepts that it introduced. Their goal is to contribute to the major technological switch, from conventional to lattice-based cryptography.

↗ The **CARAMBA** team studies mathematical, algorithmic, and high-performance software aspects for asymmetric cryptography based on number theory (RSA and Diffie-Hellman cryptosystems, elliptic curves). Their cryptanalytic work demonstrates the urgent need to increase key sizes for several of these primitives. The team is also involved in the design and cryptanalysis of symmetric cryptographic primitives (in particular in the lightweight context).

↗ The **CASCADE** team focuses on the provable security aspects of the advanced primitives or in advanced settings. They namely study the privacy-preserving primitives (such as FHE, functional encryption, etc.), but they also consider powerful adversaries, with side-channel attacks and quantum computers.

[Hal05] Sh. Halevi. A plausible approach to computer-aided cryptographic proofs. Technical Report 181, IACR Cryptology ePrint Archive, 2005.

6. http://cryptoverif.inria.fr
7. https://www.easycrypt.info/

↗ The **GRACE** team works in algorithmic number theory and the computational issues related to algebraic curves over various fields and arithmetic rings. They also build codes for error correction. Their goal is to provide better cryptosystems and better security assessments for their key sizes.
↗ The **LFANT** team researches algorithms in number theory and arithmetic geometry. They cover all aspects from complexity theory over optimized implementations up to cryptologic applications.
↗ The goal of the **MARELLE** team is to study and use techniques for verifying mathematical proofs on the computer to ensure the correctness of software. They have applied their techniques in the context of cryptographic proofs by contributing to the development of the EasyCrypt special purpose proof assistant.
↗ The **OURAGAN** team works on effective computations of algebraic objects with applications to cryptology.
↗ The **POLSYS** team develops efficient algorithms for computing the complex or real solutions in finite fields. Cryptology is one of the many applications, where it can be used in the emerging topic of algebraic cryptanalysis. This consists in reducing the security of a cryptosystem to the solving of an algebraic system with coefficients in a finite field.
↗ The **SECRET** team works on the design and analysis of symmetric primitives, of asymmetric primitives based on error-correcting codes and on cryptographic schemes based on the law of physics. They have contributed to the design of several primitives (stream ciphers, block ciphers, hash functions, code-based encryption and signature schemes, and key-exchange), and too many cryptanalytic works in these areas. They notably focus on the design quantum-safe primitives and investigate the use of quantum algorithms for attacking both symmetric and asymmetric schemes.

3.3 Cryptographic protocols and services: towards provable security

[Summary]
Security of communications and transactions is nowadays ensured by the means of cryptographic protocols, e.g., TLS. The security of the underlying cryptographic primitives and schemes is however not sufficient to guarantee overall security goals, such as confidentiality, authenticity, or anonymity. Even careful scrutiny of these protocols by experts, cannot guarantee the absence of vulnerabilities: rigorous, possibly computer-aided, security proofs are therefore absolutely needed at the level of specifications and implementations to further increase the confidence level. We can distinguish three approaches in this area. The first one uses proofs by reduction to show that breaking the security of a protocol would imply breaking the underlying cryptographic primitives. These are mathematical, generally handwritten, proofs although a new trend consists in using techniques from theorem proving and program verification to achieve computer checked proofs. The second direction uses automated verification tools to analyze the protocol specifications and find vulnerabilities in the protocol logic, such as man in the middle attacks. These tools are able to analyze complex protocols, but idealize the underlying cryptography. Finally, the third approach aims at producing verified implementations. This approach relies on expressive type systems for special purpose programming languages and requires high expertise, but can result in an end-to-end verified implementation. A major success in this area is a completely verified implementation of TLS.

Strong cryptography is itself not sufficient to ensure security goals at a higher level, e.g. securing communications or web transactions. Correctly using and programming with cryptography is a complicated task and there are many examples of security vulnerabilities that do not require breaking the underlying cryptography (see the Heartbleed, French Passport, and TLS attack examples below). It is therefore important to design and analyze protocol standards and libraries that make use of cryptography to guarantee higher-level properties.

Cryptographic protocols, such as TLS (Transport Layer Security), IKE (Internet Key Exchange), or Kerberos, are in charge of securing our connections and web transactions. They are distributed programs that use cryptography to ensure, for instance, the confidentiality of transmitted data and the authentication of communications and entities. With the increasing diversity of electronic services they are quickly spreading out: for instance, they are the basis for the security of messaging applications and of RFID enabled objects, such as electronic passports; they are also central in security services such as the widespread Single-Sign-On (SSO) or cloud-based services.

Securing data exchange on the internet – © Inria / Photo C. Morel

The design of protocols and security standards requires expertise in several areas of computer science including cryptography, computer networks, and also secure implementation. This task is difficult, even for experts, who may miss attacks due to the high complexity of these protocols. One of the difficulties in correctly designing and implementing cryptographic protocols comes from the fact that security must be guaranteed in the presence of an arbitrary attacker that controls the network and may compromise protocol participants. Vulnerabilities may arise at all levels. For instance, the famous Heartbleed[8] attack is due to an implementation error, allowing a memory overflow, in the popular OpenSSL implementation of TLS: this attack did not reveal an error in the protocol specification, nor did it break the underlying cryptography. An early version of the French electronic passport[9] was shown to be vulnerable to a linkability attack, enabling passport holders to be traced. The vulnerability was due to imprecisions in the protocol specification regarding the error messages: the French e-passport used detailed error messages, effectively allowing to differentiate a particular, previously observed passport, from another one, by using a replay attack. Finally, the FREAK[10] and LogJam[11] attacks on TLS mix vulnerabilities at different levels to downgrade the

8. https://en.wikipedia.org/wiki/Heartbleed
9. Defects in e-passports allow real-time tracking. The Register, 26th January 2010. http://www.theregister.co.uk/2010/01/26/epassport_rfid_weakness/
10. https://en.wikipedia.org/wiki/FREAK
11. https://en.wikipedia.org/wiki/Logjam_(computer_security)

cryptographic key lengths. Dealing correctly with legacy code to guarantee backward compatibility, but avoiding downgrade attacks is extremely tricky.

As illustrated by the above examples, designing protocols and services correctly and guaranteeing their security is a difficult task. Therefore, rigorous security proofs and formal analysis techniques are needed to further improve their security. There have been different complementary approaches and research directions: some analyze protocols at the specification level, whereas others directly analyze the protocol implementation; the analysis may focus on the protocol logic or the underlying cryptography; the degree of automation may also vary, ranging from completely hand-written proofs, to fully automated analysis, as well as interactive computer-checked proofs.

3.3.1 Provable security for cryptographic protocols

The provable security approach, initially introduced to give solid security guarantees for cryptographic schemes, relies on reductionist proofs (inspired by the proofs in computational complexity): typically, one shows that breaking the cryptographic primitive is at least as difficult as breaking an underlying computationally hard problem, such as factoring, computing discrete logarithms, etc. This approach has been applied to protocols, where the reduction shows, for instance, that breaking an expected security property is as hard as breaking an underlying cryptographic primitive. The adversaries that are considered are arbitrary probabilistic polynomial-time Turing machines that may interact in different ways with legitimate protocol participants and compromise some of them.

There have also been proposals for general frameworks for rigorously designing security protocols. Secure multi-party computation (MPC) protocols provide a general model for computing the result of a function, allowing different parties that do not trust each other to provide confidential input. MPC protocols make it possible to implement a variety of protocols, but efficient constructions, secure against strong adversaries, are still an active and challenging research topic. Another family of frameworks, called universally composable, or simulation-based, aim at being highly modular and showing security of components that can be assembled into larger systems. Being able to split the proof of a complex system into proofs of its components can pave the way for creating secure-by-design protocols that can be used as building blocks for larger systems.

In the work described above, proofs are generally carried out by hand, making them error-prone. As for cryptographic schemes, there have been initiatives to automate proofs in these models, e.g., through the CryptoVerif tool, or to use dedicated interactive theorem proving, e.g., EasyCrypt. Improving the scope and the automation of these tools is still a challenging, active research field.

3.3.2 Symbolic automated analysis of cryptographic protocol specifications

Symbolic automated analysis of cryptographic protocols focuses on the protocol logic and its concurrent behavior and can be applied to complex protocol specifications. While the goals are similar to those described in the previous section, the techniques and the underlying models differ. In symbolic models the so-called *Dolev-Yao* [DY81] attacker is supposed to control the communication network completely: an attacker can read any message sent on the network, remove messages, and insert (or modify) messages. The attacker is computationally unbounded, but the cryptographic primitives are idealized: the way an attacker can manipulate messages is explicitly given by a set of rules. Such rules typically specify that when the attacker knows an encryption and the corresponding decryption key the plaintext can be extracted —however, no other unspecified operation (such as cryptanalysis) is allowed. Hence, these models manipulate cryptographic primitives at an abstract, axiomatic level. However, so-called soundness results [AR07] link this approach to the provable security approach (showing that a symbolic proof implies the existence of a computational one), even though their scope is rather limited. Relying on techniques from automated reasoning and concurrency theory, proofs in symbolic models can often be completely automated, exploring all possible attacker behaviors. Mature, automated tools, e.g., ProVerif [12], as well as Tamarin [13] and AVISPA [14] exist nowadays for analyzing many security goals. Major research areas in this field consist in applying these tools to a larger class of security properties (technically this requires showing behavioral program equivalences) that allow us to analyze anonymity and privacy-like properties, to consider protocol executions on partially compromised platforms, and to scale to protocols with a complex underlying control flow.

3.3.3 Verified protocol implementations

As illustrated by the Heartbleed attack, implementation bugs may introduce serious vulnerabilities. Therefore, techniques from the field of programming languages have been used to study the security of implementations directly. This approach primarily relies on the use of expressive type systems to state security theorems directly at the implementation level. One particular successful approach in this area relies on the F* language [15], a strongly-typed higher-order effectful language especially designed for developing proven implementations. For instance, the F*

[DY81] D. Dolev and A. Yao. On the security of public key protocols (extended abstract). In *Proceedings of FOCS'81*, 1981.
[AR07] Martn Abadi and Phillip Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). *J. Cryptology*, 20(3), 2007.
12. http://proverif.inria.fr
13. https://tamarin-prover.github.io/
14. http://www.avispa-project.org/
15. https://www.fstar-lang.org/

language has been used in the miTLS project[16], a collaboration between Microsoft Research and Inria that resulted in a completely proven, reference implementation of the TLS 1.2 protocol and the new candidate protocol for TLS 1.3, going down to the level of the implementation of the cryptographic primitives. Currently, this approach only applies to carefully written implementations with many type annotations. A major challenge in this area is to make it applicable to more general code, which might not have been written by a formal methods expert. Another direction consists in extracting models directly from implementations, either by specifying small language subsets, or in a more dedicated setting by probing the system in a black-box manner. These models can then be automatically analyzed.

3.3.4 Electronic voting over the Internet

We finish this chapter with a particular application of cryptographic protocols: e-voting. Elections are arguably a cornerstone of modern democracies and a security critical process. Estonia has been pioneering this practice since 2005, and using Internet elections even for national parliament elections. Some regions in Switzerland and Australia also offered the use of Internet voting. In France, Internet voting was proposed to French citizens living abroad in the 2012 national election, but in the 2017 elections, this offer was not renewed, because of security concerns.

The main security guarantees an election must offer are vote secrecy and correctness of the result. Vote secrecy should ensure that nobody knows how a given voter voted (unless this can be deduced from the election result, e.g., in case of a unanimous vote). Correctness ensures that the result corresponds to the tally of the votes, as expressed by all eligible voters. In traditional paper elections, at least in France, these properties are ensured through a voting ritual, with a transparent ballot box, a voting booth which provides the necessary privacy to cast a vote secretly, and observers monitoring the box and the tally. The use of computers and machines significantly complicates this task, as computer programs may contain errors and correctness of a system is difficult to ensure. Moreover, the software may be intentionally tampered with, or a malware may alter its functionality. This may both change the votes, as well as leak the individual votes of persons, hence breaking both fundamental properties of an election.

To overcome the above-mentioned problems, cryptographically enforced secret, end-to-end verifiable elections have been proposed. Secrecy is generally achieved by casting an encrypted vote. This vote is then either mixed with other ballots before decryption, so that it can no longer be linked to the voter's identity, or the tally is performed homomorphically, i.e., the tally is computed on the encrypted votes providing an encryption of the result, without a decryption of individual votes. This ensures that even the tallier and server collecting the

16. https://mitls.org/

votes are unable to break vote secrecy. Correctness is achieved by the notion of end-to-end verifiable elections: the voter can verify that their vote has been correctly recorded and that the tally was performed correctly. For this, the system generates cryptographic proofs that the operations were performed correctly. This property actually avoids having to verify the correctness of the software performing the tally, as it generates evidence, i.e., mathematical proofs that can be verified independently of the result correctness.

[Highlights] The Belenios e-voting system

The **PESTO** team has been working on precise definitions of the properties a voting system should guarantee and the formal verification of these properties. This work permitted the discovery of an attack on the popular Helios voting scheme and also clarified the trust assumptions of many protocols. The **PESTO** and **CARAMBA** teams develop the Belenios voting system: a free, open-source voting system that guarantees vote secrecy and verifiability, including verifiability of the casted votes' eligibility, hence avoiding ballot stuffing. Still, this system, as many others, has short-comings: it does not prevent coercion, as a voter can prove how they voted. The system is therefore only recommended for low coercion elections. Moreover, it is vulnerable to malware that may be installed on the machine running the voting client: such a malware could leak the vote, breaking secrecy, or change the vote before it is encrypted (and verifiability only allows to track the encrypted ballot). Solving coercion-resistance and malware-resistance in a satisfactory way are still open research questions.

Moreover, one should note that Internet voting removes the privacy guarantee of a voting booth and requires a means to identify voters remotely. This is actually the case for any remote voting scheme, including paper-based ones, such as postal voting. Finally, cryptographic e-voting schemes also require voters to trust some experts as they rely on advanced mathematical notions, making their understanding difficult. Therefore, these systems seem not yet ready for high stakes, e.g., political, elections.

[Research challenge 5] End-to-end formally verified cryptographic protocols

As the security of cryptographic protocols is extremely difficult to ensure (pencil and paper proofs regularly contain errors), the use of rigorous, formal methods appears increasingly as the only way to achieve the expected security level for this class of systems. Therefore, the area of computer-aided security proofs is an increasingly important topic and must include all aspects from the specification down to the implementation. Recent work, in particular around TLS 1.3, have shown that this is now achievable.

However, such proofs still require carefully crafted code and a very high level of expertise. Leveraging the proof techniques to make them applicable to more general code and usable by a wider audience is now the main challenge. Different protocols often ensure different security properties, but existing tools for verifying certain properties, such as anonymity, do not yet have the same maturity as tools for verifying authentication properties. Yet another challenge is to consider stronger adversary models, e.g., an adversary that may control part of the computer through malware.

[Inria teams]

There are several Inria teams working on formal proofs and analysis of cryptographic protocols:

↗ The **CASCADE** team, in addition to their work on cryptographic primitives, also works on the design of protocols and their security analysis in computational models.

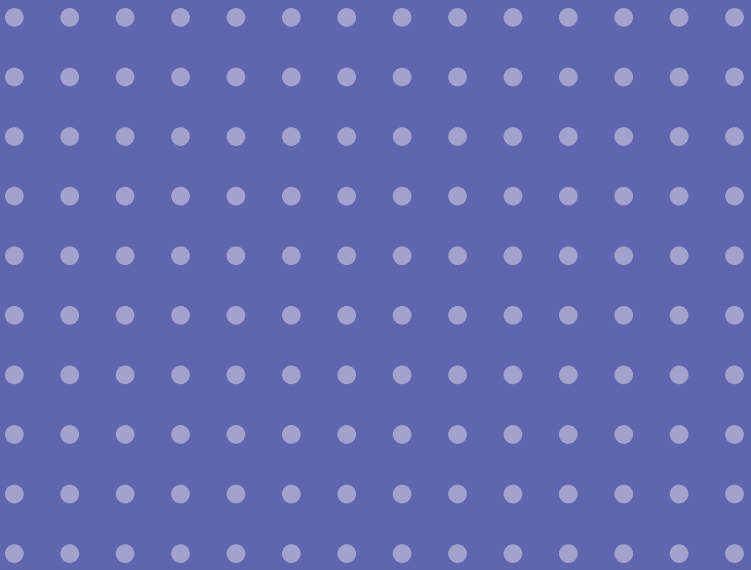
↗ The aim of the **PESTO** team is to build formal models and techniques, for computer-aided analysis and design of security protocols, using techniques from automated reasoning, concurrency theory, and programming languages. They are particularly interested in automated analysis of anonymity properties and e-voting protocols. They contribute to the development of several tools, including AVISPA [a], DEEPSEC [b], and the Tamarin [c] prover.

↗ The **PROSECCO** team conducts formal and practical security research on cryptographic protocols, software security, web security, and hardware protection mechanisms. To this end, they design and implement programming languages, formal verification tools, dynamic monitors, testing frameworks, verified compilers, etc. They develop the automated protocol prover ProVerif [d] and contribute to the design of the F* language that they used to develop a completely verified implementation of TLS. They have also worked on extracting models from implementations of the PKCS#11 key management standard, resulting in the creation of the Cryptosense startup.

a. http://www.avispa-project.org/
b. https://deepsec-prover.github.io/
c. https://tamarin-prover.github.io/
d. http://proverif.inria.fr

4

Security services and mechanisms



When a user starts using a computer system, she first identifies herself (identification), then she proves that she really is who she claims to be (authentication). This proven identity is then used by the system to grant or restrict access to a resource or service only to authorized users or entities (access control), or to avoid a specific information to flow to a given destination (flow control) through the actions of a user. When a user needs to execute a program on an untrusted machine, a hardware-based solution may guarantee isolation and software integrity. All these security services are preventive. Unfortunately, they can sometimes be bypassed by attackers and therefore reactive security is also needed. The user actions are therefore monitored to verify that they do not violate the security policy (intrusion detection and alert correlation). Of course, such a policy violation may have occurred without the user's knowledge, who may have been attacked by a malware acting on her behalf. Malware analysis and detection are thus other security services to offer. Ideally, if ever an intrusion or a malware is detected, the system should react and at least reconfigure itself to avoid another similar attack.

In the remainder of this chapter, we successively present the security services that have been introduced above: identification and authentication (§4.1), access and flow control (§4.2), trusted computing (§4.3), intrusion detection and alert correlation (§4.4), malware analysis and detection (§4.5), and reaction (§4.6).

4.1 Identification and authentication

[Summary]

Identification and authentication are generally the first two security services used when initiating a cyber exchange, either between a human being and a machine or between two machines.

Identification, for a given entity (i.e., a user, service, device, etc.), is the act of stating its identity. Authentication, for this entity, is the act of proving that it is really the entity it previously claimed to be.

Authentication is used to restrict access to a resource or service only to authorized users or entities. Authentication is achieved by presenting an *authenticator* that generally belongs to one of the following three classes: what you *know,* e.g., a password or pin; what you *have,* e.g., an access card; what you *are,* e.g., mechanisms based on biometry. These authenticators are often combined in so-called multi-factor authentication. Despite their numerous drawbacks, passwords remain the most common means of authentication.

Another form of identification is the ownership of a piece of data. This can be achieved through watermarking, which consists in hiding messages in the data. A

good watermarking technique must create a robust link between the piece of media and the hidden message, such that distorting the media does not erase the message. Watermarking is used in copyright management and copy protection, mainly focusing on multimedia content even if the spectrum of applications is larger and it may be used, e.g., to protect source codes, e.g., to protect source codes, or databases.

Identification, for a given entity (i.e., a user, service, device, etc.), is the act of stating its identity. For example, a user would provide their login. As such, identification is not really a security service, as the entity can lie and give the wrong information. That is why the authentication service is needed ; it is the act of proving that it really is the one it claimed to be. For example, a user could reveal the password that is associated with their login.

Notice that some authors consider that authentication is a service related to identification. As such, the word identification can be used for "identity authentication."

4.1.1 User authentication

Authentication is used to restrict access to a resource or service only to authorized users or entities. For example, to access a system a login and password are required[1]. Many on-line services provided by private companies or public administrations also require user authentication. In such cases, identification happens through a network and the means for identification may be coupled with a cryptographic protocol such as TLS (Transport Layer Security).
Authentication is achieved by presenting an *authenticator* that is generally in one of the following three classes:

- what you *know,* e.g., a password or pin;
- what you *have,* e.g., an access card;
- what you *are,* e.g., mechanisms based on biometry.

These authenticators are often combined in so-called multi-factor authentication. For example, cash withdrawal with a credit card requires both possession of the credit card and knowledge of the pin. Creating a new user identity, letting the user manage it, offering a password recovery service, and perhaps multi-factor authentication is however a highly complex task.

Passwords remain the most common means of authentication. Unfortunately, password management is complicated, both for the user who wants to be identified and the system that grants the access. On the system side, passwords need to be stored. However, passwords must not be stored as plaintext[2], as a leak of the stored

1. Data authentication is different from entity authentication and is generally achieved using cryptographic means, e.g., by a key exchange protocol, and the use of a message authentication code (MAC) with the resulting key, see Chapter 3.
2. See, e.g., Recommendation R22 in *ANSSI Note technique - Recommendations pour la securisation des sites web*. [in French] https://www.ssi.gouv.fr/uploads/IMG/pdf/NP_Securite_Web_NoteTech.pdf

list would directly compromise all user accounts and passwords. It is therefore highly recommended to store only the hash value of the passwords, using a one-way hash function. Moreover, as passwords need to be easily remembered by humans, they are vulnerable to dictionary (or guessing) attacks, i.e., brute-force attacks by enumeration. We distinguish online and offline guessing attacks. In online guessing attacks, an attacker tries all possible passwords by executing the identification mechanism for each trial. These attacks can be thwarted by adding an additional delay after an unsuccessful trial or limiting the number of attempts, making this approach infeasible. Offline guessing attacks are possible when a list of hashed passwords is leaked. To avoid the same brute-force attack being applied to all passwords at the same time, a salt value is used: for each user, one stores a random value, called salt, and the hash value of the result of the concatenation of the password with the salt. Some hash functions are also specially designed to be costly in time and memory consumption, in particular on dedicated hardware, to slow down brute force attacks.

Ideally, one would like to have a hash function that is fast when verifying correct passwords and slow on incorrect ones: this idea has been partially realized by the notion of *pepper*. Similar to salt, an additional random value, the pepper, is hashed. This value is however not stored, and must be brute-forced: when the correct password is provided the expected number of hashes is N/2, where N is the number of possible pepper values, while N hashes are needed to discard an incorrect password. From the user's perspective passwords are often complicated to manage: ideally passwords should be difficult to guess and should not be re-used for different services. Some services require the use of digits or special characters in passwords. However, recent recommendations[3] question this practice and recommend longer passwords, also called passphrases. Indeed, studies on real data have shown that without constraints, the word "password" appears as one of the most popular choices. When adding special characters or digits "password" is replaced by "password!" and "password123" in the list of most frequent passwords, hence not improving the security. A good practice is therefore to use a password manager that encrypts all passwords using one master password. Non-technical attacks such as social engineering or phishing are discussed in §2.3.1.

Given the high number of password leaks, there have been efforts to either get completely rid of passwords or couple them with a second authenticator. More widespread, is the use of two-factor authentication. For online payment, the 3D-secure protocol may rely on a confirmation code, sent by SMS to your mobile phone, that needs to be re-entered on the device used for the payment. The goal is to prove both knowledge of the credit card number and possession of the phone. Similarly, the Google 2 Step protocol, can either send a confirmation

3. NIST Special Publication 800-63B. Digital Identity Guidelines – Authentication and Lifecycle Management – https://pages.nist.gov/800-63-3/sp800-63b.html

code, require a tap on the phone, or be configured to require the use of a FIDO U2F USB dongle plugged in your computer. These multi-factor protocols generally offer additional security. However, the increased complexity of both the protocol and the recovery methods, e.g., when a phone is lost or broken, may sometimes increase the attack surface.

Biometry is another means of user identification that relies on sensors to measure biologic characteristics, such as fingerprint, iris, voice, or face specificities. An inherent difference with other authenticators is that biological data is a priori not secret and cannot be modified, or replaced. Therefore, sensors are becoming more sophisticated in order to distinguish the physical presence from a copy (e.g., a fake finger with a copy of the fingerprint or a photo of a face). Although these means are becoming increasingly popular, e.g., laptops and mobile phones can be unlocked using fingerprint or, more recently, face recognition, they still suffer from false negatives and false positives. Therefore, it is recommended that they are only used as a second factor authenticator.

4.1.2 Identification of data owner: watermarking

Data hiding is the art of hiding messages into a cover media. It encompasses two branches, steganography and watermarking, where the word 'hiding' finds two different meanings.

- In steganography, 'hiding' messages means that an adversary, the steganalyzer, cannot statistically detect whether a piece of media contains a secret message. Thus, changing or not changing the value of a pixel of an image that is shared by two or more people is a simple (and actually naive) way of transmitting hidden information between these people. Steganography finds application in Intelligence services with steganalysis in fight against terrorism for example.
- In watermarking, 'hiding' means that secret messages are deeply embedded into media. This secret can be invisible to a human being, making the watermark a special case of steganography. It creates a robust link between a piece of media and the secret such that distorting the media (for example, rotating or cropping an image) does not erase the secret. Watermarking finds application in copyright management and copy protection. For instance, watermarking identifies the owner of a piece of data by embedding in their piece of media their identity as a hidden message.

While literature about watermarking mainly focuses on multimedia content, the spectrum of the cover media natures is very broad: programs (protection of source codes, parameters of Deep Learning classifiers, etc.), databases, maps, 3D objects, or DNA sequence.

Watermarking should not spoil the regular use of the cover media. For multimedia content, the user should not perceive any difference. Watermark embedding makes a big usage of human perceptual models. For programs, a drop in performance (relevance of the output, runtime) should be kept small. For databases, querying should yield similar answers.

The robustness of watermarking is gauged by measuring how the probability of a hidden message decoding error increases as the protected media is more and more distorted. The worst-case attack is defined as the attack maximizing this error probability for a given level of distortion.

Watermarking is nothing more than a secret communication scheme. Embedding and decoding share a secret key which defines how the message to be hidden modulates a given part of the cover media. Security enters the picture when the same secret key is used to protect many pieces of media. It amounts to answering the question as to whether an adversary can estimate the secret key by analyzing these watermarked pieces of media. Once the secret key is disclosed, the adversary may erase the watermark (removing any proof of ownership), modify the watermark or embed their own watermark into any media to usurp ownership.

The actual trend in watermarking moved away from ownership authentication to media consumer identification. This especially concerns confidential and highly valuable documents. Thanks to watermarking, the identification code of a user is embedded into their copy of the media to make it unique. This will disclose the identity of the user who has leaked the content. It does not prevent illegal redistribution per se, but it is a dissuasive weapon to avoid leakage. Identification codes are designed such that even if a collusion of several users mix their copies, the decoding will identify at least one of the traitors.

Another trend is emerging making the connection between data hiding and the generation of adversarial samples deluding deep learning classifiers.

Robust watermarking is a mature technology. Many research results have already been transferred to real life products. The number of research works on the topic has drastically decreased in recent years. The main consumer of watermarking technology is the entertainment industry. However, a robust *and* secure watermarking doesn't exist yet. It appears that the entertainment industry has little interest in security and is satisfied with the level of robustness achieved so far. More research on watermarking as a means to authenticate ownership is thus not recommended. The trend is on a cross-layer design of watermarking and traitor tracing codes, and on the development of protocols between parties (content owners, content distributors and content consumers), who do not trust one and another.

[Inria teams] Identification and authentication

↗ The **LINKMEDIA** team and its spin-off company Lamark work on multimedia (audio/video) protection through watermarking. For example, in the domain of traitor tracing, the team proposes identification codes such that even if a collusion of several users mix their copies, the decoding will identify at least one of the traitors. From the attacker point of view, the team studies how an attacker who has one or several watermarked pieces of data can estimate the secret key used by the watermarking scheme.

↗ The **MULTISPEECH** team studies voice-based authentication and detection of spoofing attacks. It co-organizes the international ASVspoof challenge.

↗ The **PESTO** team applies formal, symbolic verification methods to analyze the security of multi-factor authentication protocols.

↗ The **PROSECCO** team uses symbolic verification in ProVerif to analyze web authentication protocols such as OAuth 2.0 and ACME with respect to a novel web attacker threat model.

4.2 Access control and flow control

[Summary]

Enforcing security first involves precisely defining which previously identified and authenticated entity may have access to what information and in which way. Classically, access permissions (reading or writing) to information are granted if and only if some condition is fulfilled (for example, the user who asks for access is correctly authenticated). A read-write sequence engenders an information flow that may also be controlled in some cases.

Preventive security requires to first define precisely which entity may access what information and in which way. Classically, permissions to read or write information are granted if and only if some condition is fulfilled. For example, the Bell-LaPadula policy states that high level (i.e., secret) information cannot flow into low level (i.e., public) containers.

To implement such a policy, each information container (e.g., a file) is classified (e.g., secret or public) depending on the type of information it contains and each user has a clearance (e.g., secret or public) depending on the type of information they need to know, given their role in the organization: a secret-level user can read both public or secret containers, but can only write into secret containers. This ensures that previously read secret information will never flow to a public container in which a public-level user can read.

Practically speaking, the implementation of a security policy generally builds on access control and flow control mechanisms, implemented at the operating system level.

© WavebreakmediaMicro - Fotolia

4.2.1 Access control

Access control generally refers to regulating the requests to access the resources managed by an information system. This regulation can happen in various places of the information system: at the network level (in firewalls), at the node level (in operating systems), or at the service level (in applications).

At the network level, firewalls allow access to the network resources for authenticated users or legitimate traffic and deny access to unauthenticated users or illegitimate traffic. However, bad configuration of a firewall may cause security breaches. In particular, conflicted filtering rules can lead to blocked legitimate traffic and to accepted unwanted packets.

In operating systems, access control enforces the security policy by granting authorizations to subjects (i.e., authenticated users or processes) on objects (i.e., resources like data, network, computation facilities, etc.). Many access control models have been proposed and some have been widely used for decades, like the DAC and RBAC models. Such models typically represent the authorizations as triples (subject, object, permission) and are frequently employed in operating systems and databases. Many variants of these models have been proposed in the last decade, to capture additional information such as the context of the access (time, location, etc.), the purpose of the applications, or the specificities of the organizations; this gave rise to context-based, purpose-based, and organization-based access control.

Today such solutions are well established and often normalized. In general, these models consider a centralized architecture, i.e., the resources and additional information required to define the access control policies are located on the server side and managed by trusted authorities and administrators.

More recently, cloud computing and the emerging areas of user-centric systems and the Internet of Things (IoT), put a new light on access control research. While the goal is still to define, evaluate, and enforce authorizations, the intrinsic specificities of these contexts induce a deep rethinking of the access control models and their enforcement strategies. In cloud computing, data and computations being outsourced to potentially untrusted entities, access control management must adapt to new trust assumptions. In user-centric systems, the focus is on enabling access control management without resorting to IT experts. The IoT raises the problem of adapting the access control when the Things, who have scarce resources, collect the data.

In addition, Things (e.g., embedded sensors or GPS devices, pedometers, smart meters, connected TVs, toys) track the events occurring in their surroundings and thus generate sensitive data. A myriad of new applications and services are being built by querying this data. Designing access control models for the Internet of Things is thus a critical problem, although difficult because of two conflicting objectives: (i) the access control model should be generic enough to cover the needs of highly diverse applications and (ii) it should be lightweight considering the hardware constraints of the Things they target. So far, the data collected by Things end up on centralized servers where they are analyzed and queried.

In human centric contexts (e.g., the personal cloud), individuals want to manage their personal data themselves, i.e., under their control and not delegate this task to a central administrator. However, designing a well-calibrated access control policy, and enforcing it, confronts the individuals with the difficult choice between delegating the data administration to a qualified third party and giving up their sovereignty or taking charge of it themselves using complex sharing models and tricky security protocols that they probably do not master. Hence the need to design new access control models, simple enough to be managed by individuals.

To conclude, although access control is a long-lasting research issue, the evolution of computing architectures opens important new lines of research. The current trend suggests that the access control, usually thought in a centralized context, should be considered in a more distributed and global context, i.e., regulating data accesses along the whole life cycle of the data, from the Things that collect it to the Cloud that stores it. In terms of enforcement, this also calls to complement access control with security techniques like encryption, data flow control, and trusted computing.

4.2.2 Information flow control

Information flow control consists in monitoring how a piece of information flows through a system or a program. The goal is to ensure either statically or dynamically that a piece of private information is manipulated according to its security property. A typical property is that a private information does not leak towards a public channel.

DYNAMIC INFORMATION FLOW TRACKING

Dynamic Information Flow Tracking (DIFT) is a popular and versatile technique for monitoring the flow of information through a system. Typically, data entering the system is tagged according to its security levels and tags are propagated through the system. This technique can be used to monitor a system and ensure that the flow of information abides the security policy. A basic property is that untrusted user-provided data is sanitized before being processed. DIFT is also a powerful tool for penetration testing and vulnerability analysis: a bug, e.g., a buffer overrun, may be considered a security vulnerability if the data is influenced by an untrusted input.

DIFT analysis can be performed at various levels of granularity. At the system level, the challenge is to instrument the operating system with security tags and get a solution that is correct, precise, and requires some limited maintenance. To improve the scalability of DIFT, several hardware solutions have been proposed. For instance, the CrashSafe project proposes a novel processor design. There are also less intrusive proposals where an independent co-processor is dedicated to managing security tags.

DIFT is often limited to direct information flows that can be enforced by monitoring one execution at a time. Remarkably, DIFT can also enforce hyper-properties, such as non-interference, which are trace set properties. As this monitoring stops the execution in case of violation, they turn an information leak into a termination leak that may be acceptable for some applications.

STATIC INFORMATION FLOW TRACKING

There is a vast literature of type-systems for information flow. Sophisticated type-systems handle rich language features such as exceptions and method dispatch. The JIF system for Java is probably the most impressive implementation of static information flow control. The traditional property that is enforced by type-systems is non-interference, which basically states that public data does not depend on private data. To accommodate programs that leak a controlled amount of information, e.g., cryptographic primitives, practical type-systems need to include a notion of *declassification*. Declassification may take on various forms but requires some kind of user-provided specification that is trusted by the system. In general, it is hard to predict whether the long-term effect of a declassification corresponds to the user-intent.

QUANTITATIVE INFORMATION FLOW

An alternative to declassification is Quantitative Information Flow (QIF) where the goal is to quantify either statically or dynamically the amount of information that is leaked by a system, through some observation by the attacker.

To prevent QIF, typical methods of security such as encryption and access control are not applicable: the only way is to obfuscate the link between the secret and the observable. Ideally, we would like systems to be completely secure, but in practice this goal is often impossible to achieve. Therefore, it is important to have a quantitative theory of leakage, so as to measure the vulnerability of a secret, assess whether a system is better than another, or evaluate the effectiveness of a method to prevent leakage. The quantitative aspects stem from the fact that the knowledge of the adversary is typically of a probabilistic nature and that the best methods to prevent leakage are often randomized methods. The most successful approaches to the foundations of quantitative information flow are based on information theory and on the notion of entropy (in various versions: Shannon, Rényi, or guessing). The entropy measures the vulnerability of the secret and the choice between various notions reflects the different operational adversary models that one is interested in.

A first drawback of QIF is that different definitions of leakage offer different, sometimes incomparable, security guarantees. QIF approach captures a large variety of attacks (approximate guess, multiple guesses, or properties of the secret) and subsumes most of the information-theoretic approaches considered in the literature and their corresponding notions of entropy.

A second problem with QIF is that average probabilistic guarantees may not provide an adequate security guarantee in case of an active attacker, i.e., when the attacker controls some of the input data and tries to use it to trigger security vulnerabilities.

COMPILERS FOR INFORMATION FLOW ENFORCEMENT

Another research trend aims at preserving high-level information flow properties through the compiler chain. The flow of information matters for multitier languages where the compiler has the responsibility to decide which tier stores sensitive information and how to ensure the security of the communications. Information flow is also a stringent matter for critical code, e.g., cryptographic primitives, where any leakage due to the implementation may break the mathematical security guarantee. In this context, the compiler aims at protecting against side-channel attacks, such as timing-attacks or power-analyses.

[Inria teams] Access control and flow control

↗ The **CELTIQUE** team develops certified compiler techniques and analyses for protecting against timing-attacks. Together with the **INDES** team, **CELTIQUE** develops a theory for hybrid monitors that augments a dynamic information flow monitor with a (dynamic) static analysis of non-executed branches computing a symbolic and quantitative form of information flow. Together with the **MARELLE** team, **CELTIQUE** works at providing compiler support for *constant time* programming, a strict programming discipline adopted by cryptographers to limit timing leaks.

↗ The **CIDRE** team has a strong expertise in Dynamic Information Flow monitoring both at the system and hardware levels: the team develops the blare tool, an intrusion detection system (IDS) allowing, on Linux and Android, to dynamically evaluate the legality of information flows. A hardware device has been designed to improve the precision of this evaluation.

↗ The **COMETE** team proposed the theory for grounding security on notions of quantitative information flow and developed the g-leakage framework. They also developed the library Libqif[a], a C++ toolkit implementing a variety of techniques related to g-leakage, quantitative information flow and differential privacy. This team is also investigating an approach based on Game Theory to limit the leakage of information in the presence of an active attacker.

↗ The **FUN** team investigates a way to avoid transmitting all the data collected by the Things to a server. The team uses decentralized storage to avoid reconstructing all data at once. In addition, it identifies malicious relay nodes to bypass them during the data collection process.

↗ The **INDES** team works at defining, comparing, and evaluating the limitations of different information flow policies -including declassification and computational noninterference for cryptography- at the language level. Applications domains mainly include JavaScript, reactive, and multitier languages for IoT. The team also develops static, dynamic, and hybrid sound enforcement mechanisms for information flow security and works on preservation by compilation of information flow guarantees

↗ The **RESIST** team, in collaboration with the **PESTO** team, works on methods for managing firewall configuration files that automatically reveal anomalies and help the administrator to find an adequate position for a newly added filtering rule.

↗ The **PETRUS** team designs access control mechanisms for the personal cloud that do not require the human user to understand the underlying access control mechanisms for enforcing a given security policy. In the IoT context, the team designs control models relying on embedded data management structures and algorithms for Things, so that data dissemination decisions can be evaluated closer to the data, within the Things that collected them.

↗ The **PROSECCO** team aims at designing formally secure compilers for architecture with support for dynamic information flow control. The team also studies how hardware solutions for dynamic information flow tracking can be exploited to ensure that the running code satisfy security policies.
↗ The **VALDA** team studies both foundational and systems aspects of complex data management, especially human-centric data. They proposed a collaborative access control model in the context of the language Webdamlog (distributed datalog). This model allows individuals to declaratively specify powerful policies governing access to their data, dissemination of their data, and delegation of computation.

a. https://github.com/chatziko/libqif

4.3 Trusted computing

[Summary]
Trusted computing builds on assurances provided by secure hardware. Such hardware can ensure platform integrity and secure boot. It is also possible to provide trust at the application level rather than at the level of the whole platform by executing code in isolated, trusted execution environments, called enclaves. Additionally, these enclaves offer the possibility to attest that results were produced by a given code, opening the possibility for outsourced computation of secure data-oriented tasks.

Trusted computing consists in building secure hardware in order to derive global guarantees about computations made on a platform. The first use of this approach aimed at ensuring the platform integrity by using Trusted Platform Module (TPM) and secure boot. A TPM includes a unique cryptographic key and can compute hash functions. This allows it to authenticate hardware devices and to verify that software has not been changed, and thus to certify the integrity of the whole boot sequence. This permits the user to provide guarantees to external parties that their machine is indeed running a specific OS and specific applications. This approach aims at providing full platform integrity under the assumption that the OS and applications are trusted. Following this approach, TPM-based virtual machine monitors (VMM) have been developed, allowing for isolation of multiple guest OS's and thus isolation of attacks on these OS's. This led to the development of microkernels (e.g., SEL4) and unikernels (e.g., MirageOS), with the aim of minimizing the trust required in the OS and applications.

More recently, with the rise of secure areas in main processors (e.g., Intel's Software Guard Extension (SGX) and ARM's Trustzone), one can guarantee that code and data in memory is protected with respect to confidentiality and integrity. It is

thus becoming possible to ensure trust at that application level rather than at the level of the whole platform. Indeed, these isolated execution environments offer the capabilities to execute code in enclaves, whose memory and control flow is protected from the environment (including the OS). Additionally, these enclaves provide attestation capabilities, i.e., means for an external party to check that claimed messages were indeed produced in an enclave running a specific piece of code. This much more versatile approach leads to many interesting research directions for securing computations both in the cloud and on home devices. Typical applications include secure licensing, which leverages guarantees provided by enclaves to ensure that licensed software is not illegally used.

An important challenge related to this disruptive technology is studying its applicability in the case of secure outsourcing of (possibly distributed) data computations. As Trusted Execution Environments (TEE) provide in hardware security guarantees like confidentiality, integrity, and attestation, the code running inside a (local or remote) TEE enclave can be considered as sticking close to a fully honest behavior. This opens up the field to generic, efficient, scalable, secure data-oriented computation tasks. VC3 is a typical preliminary attempt by Microsoft towards trustworthy data analytics based on Intel SGX enclaves in the cloud. Two main research issues must however be overcome. First, the hardware security properties provided by TEE enclaves cannot be considered as unconditionally unbreakable and should lead to investigating slightly different threat models. Second, the most common data-oriented tasks (e.g., private search on the Web, secure data stream processing in the IoT, privacy preserving machine learning) must be efficiently transposed in secure counterparts based on TEE's. The challenge here is to optimize and secure the execution of data-oriented primitives according to the TEE constraints.

These lines of research are currently emerging. The current trend suggests that the availability and diversity of TEE technologies will increase in the near future. New solutions are forthcoming, with multicore platforms in which security/isolation-oriented cores (à la SGX) would cohabit with other all-purpose cores, thus allowing for separation of tasks inside the CPU. This suggests that trusted computing will become ever more prominent in the near future, and Inria will contribute to the research efforts into this direction.

[Inria teams] Trusted computing

↗ The **PETRUS** team works on new properties to reason about secure hardware computation in the context of distributed database computations, e.g., limiting the amount of data accessible to each agent/enclave. The team also studies database processing using secure elements or more advanced TEEs, like Intel SGX.

4.4 Intrusion detection and alert correlation

[Summary]

Nearly any system contains, generally unintentional, flaws. An attacker can exploit them to bypass existing preventive security mechanisms, e.g., access control. Therefore system monitoring is of crucial importance to identify any violation of the security policy. In order to detect intrusions, the main and largely deployed approach consists in defining malicious symptoms and looking for their occurrences in various information sources (network, OS and application logs, etc.). An alternative consists in defining the normal activities of the monitored system and measuring the potential deviations from this normality. In both cases the challenge is to detect all intrusions but only the intrusions. In practice however, detection is far from being so perfect, leading to numerous false positives (false alerts) or false negatives (attacks not detected). This is why several Inria teams explore different ways of producing alerts.

Intrusion detection leads to a huge number of alerts, many of them being false positives. Thus an additional step is needed: alert correlation. This step consists in applying to the alert flow a series of transformations to improve the content of the alerts progressively (for example by adding information on the success of the corresponding attacks, on the origin of the attacks, on the vulnerability that has been exploited, on related alerts, etc.) and thus increase the "situation awareness" of the administrator.

Numerous flaws (i.e., vulnerabilities) are introduced in any system during its design, implementation or configuration. Generally, this introduction is not intentional, but the malicious introduction of vulnerabilities is still a possibility. These vulnerabilities may be exploited to bypass preventive security mechanisms used to enforce the security policy. In addition to preventive security, a second line of defense, namely intrusion detection, is thus mandatory. It consists in monitoring systems in order to detect any violation of the enforced security policy. By "intrusion" we thus mean "violation of the security policy" which breaks confidentiality, integrity, or availability. Using a malware (virus, worm, logic bomb, etc.) is of course a good way to implement such a violation.

Intrusion detection is a reactive security service that consists in collecting information on the operation of the system under surveillance and analyzing these activities to produce alerts if they are considered to be malicious. As the analyzer is very often prone to generate false alerts, a second step of analysis considers the flow of alerts and tries to eliminate the false alerts from this flow. In addition, alerts are correlated in order to identify multi-step intrusion scenarii. Finally, once the remaining alerts can be trusted, reaction to detected intrusions can be considered.

4.4.1 Intrusion detection paradigms

An intrusion detection system (IDS) analyzes data coming from either the network traffic (Network-based IDS, NIDS), the operating system or the applications (Host-Based IDS, HIDS).

The analysis of these two categories of data follows two different detection paradigms: misuse-based and anomaly-based intrusion detection. Anomaly detection consists in defining normal activities of the monitored entity and in identifying any deviation from this normality; while misuse detection consists in modeling malicious activities and detecting occurrences of these activities.

The most classic misuse detection approach, popularized in the late 1990s with the snort NIDS[4], is to search for traces of known attacks in network packets. It involves constantly updating an attack-signature database; zero-day attacks[5] are usually not detected. Furthermore, a compromise has to be made between the selectiveness of the signatures and the risk of false positives and false negatives. Very selective signatures increase the risk of missing attack variants, while more generic signatures may result in high false-alarm rate. In practice, signatures have often been very simple, thus very generic, in order to allow real time analysis of every event, usually leading to a higher rate of false positives.

Anomaly detection is less frequently used than misuse detection. Here, defining the normality through a behavior model is of course the critical point. If the model is too precise, the detector raises a high number of false alarms; if it is too lax, it misses attacks. Finding a good balance is difficult, especially when the model is built using statistics or machine learning, because of the possibility of under training or over training. In addition, a good balance for a given activity may later become unacceptable when the activity evolves. In any case, dealing with legitimate but unplanned behavior is problematic for anomaly detection.

The information gathered in network packets can sometimes be too poor semantically to allow for a good detection process. Therefore, intrusion and anomaly detection must also be addressed at both the application level and the operating system level. The most common anomaly detection mechanisms at application level consist in detecting a deviation of the control-flow of a program. A popular method to detect such anomaly is the use of application sequences of system calls. However, such methods can be bypassed by either mimicry attacks [WS02] or attacks against the integrity of the system call parameters.

4. https://www.snort.org/

5. A zero day vulnerability is one that is unknown to those who would be interested in mitigating the vulnerability.

[WS02] David Wagner and Paolo Soto. Mimicry attacks on host-based intrusion detection systems. In *Proceedings of the 9th ACM conference on Computer and communications security (CCS)*, pages 255-264, New York, NY, USA, 2002. ACM.

[Research challenge 6] Intrusion detection for encrypted networks

Nowadays, intrusion detection is essentially realized at the network level. If, as expected in the near future, the traffic were more systematically encrypted, which would of course be a good practice for security and privacy, the analysis of the network packets would become *de facto* inoperative, apart from the header analysis. Therefore, it becomes important to study and design new mechanisms for monitoring information systems and producing alerts, at the application, middleware, operating system, and even firmware or hardware levels.

4.4.2 Alert correlation

We distinguish implicit and explicit alert correlation approaches. Implicit alert correlation uses data-mining paradigms to fuse or aggregate alerts, simply building on the similarity between alert features (e.g., IP addresses of the victim and the attacker), or using more advanced techniques to extract relevant information from alert groups (mining rules between attributes of alerts). Explicit alert correlation relies on security experts to specify logical and temporal constraints between alerts that correspond to complex attack scenarios, which generally require several steps to achieve their ultimate goal. When a complete or partial intrusion scenario is detected, a higher-level alert is generated.

Additional information about the characteristics of the attacks and about the context in which they occur is also useful to the correlation process. This knowledge allows one to take the context into account while processing the alerts, to identify false positives or to evaluate if a given attack has any chance of being successful given the context in which it occurs.

Unfortunately, it often remains difficult to write correlation rules that correctly exploit all the information available and that correctly translate the system administrator's expertise relative to possible attacks against the system. Automating the rule production is thus a current issue for research.

[Inria teams] Intrusion detection and alert correlation

↗ The **CIDRE** team has extensively studied intrusion detection (application level and network level) and alert correlation. At the application level, the team has proposed an approach to detect the corruption of data items that have an influence on the system calls. This approach consists in automatically building a data-oriented behavior model of an application by static analysis of its source code that is used to build constraints on data manipulated by the program. The application is then instrumented with executable assertions to check these constraints at runtime. At the OS level, the team proposed an anomaly detection approach in which the behavior model is not learned but given in the form of an information flow policy. The basic idea is to define where each information

(for example, the information contained in each file at the initialization of the system) can be stored, potentially mixed with another information. As for correlation, the team introduced an attack description language allowing one to define alerts to be produced by the attack and the logico-temporal rules between these alerts. These rules are used to configure a correlation engine realized by the team. **CIDRE** also proposed with its partners a data model to represent a system under monitoring: this model can be used during the correlation process to bring contextual information, for example for false alert identification. As these correlation rules are sometimes hard to define, the team also proposed, in collaboration with the DGA, a comprehensive process to generate such rules as automatically as possible, starting from attack trees (see §3.4) that have usually already been realized by the administrator to evaluate the threats against its system during the risk analysis phase.

↗ The **LACODAM** team tackles the analysis of large network data to identify potential *advanced persistent threats* (APT) by discovering symptomatic patterns in the metadata of IP-packets. Finding such complex patterns over a large volume of streaming data implies revisiting the existing stream mining algorithms to improve their throughput dramatically, while guaranteeing a manageable false positive rate.

↗ The **MYRIADS** team investigates misuse detection in cloud environment contexts. This is particularly tricky as the information system can be dynamically and automatically reconfigured. Security monitoring mechanisms should then be placed under control of the cloud provider and should follow the dynamics of the environment. In such a context, the team proposed a self-adaptable misuse detection system for IaaS clouds, which monitors changes in the virtual infrastructure of a cloud environment and reconfigures security probes accordingly. In addition, the team proposed a method to enable a cloud customer to verify that a network intrusion detection system located in the operator infrastructure is correctly configured, according to the Service Level Objectives figuring in the Service Level Agreement.

↗ Focusing on network data, the **RESIST** team works at building solutions to characterize and detect unwanted network behavior. The team proposed a clustering and visualization method that allows one to analyze a large number of IP packets in order to make malicious activity patterns easily observable by security analysts. The team also proposed a technique to investigate https (thus encrypted) traffic. The team defined dedicated features for https traffic that are used as input for machine learning algorithms processing full tls sessions. This allows the early identification of encrypted web services in the tls session with a high degree of accuracy, which then enables anomaly detection in the usage of identified services. Another interesting contribution of the team is related to quantification of the number of monitoring nodes required to ensure an acceptable false positive rate for different network topologies. The team has shown that the false positive rate can be reduced by strategic monitoring node placement.

4.5 Malware analysis and detection

[Summary]
Malware (viruses, worms, ransomware, spyware, adware, trojan horses, keyloggers, rootkits, etc.) is of course a major threat to our information systems (OS, applications, and data), especially on the client side (PC's or smartphones).
The goal of *malware analysis* is to obtain full understanding of a suspected malicious code: identify the targets (e.g., a particular end-user, a machine running a particular OS), attack actions (e.g., leakage of information, or encryption and ransom), techniques to bypass security mechanisms, and its own protection mechanisms to avoid detection. To succeed, the analysis must first defeat the anti-analysis protections put in place by the malware creator (obfuscation).
Malware detection is usually based on a the analysis of any information received by a device (a machine, a phone, a firewall) or even on the complete scan of the files contained in a machine. The detection engine compares the data retrieved to a database of known features symptomatic of malware (i.e., malware signatures). The challenge is to maintain an up-to-date signature database since malware authors constantly generate new versions based on the same malicious code in order to escape scrutiny. Therefore, research projects propose detection techniques based on the concrete behavior of the malware, which remains constant across versions.

4.5.1 Malware analysis

Malware analysis intends to dissect any piece of code identified as suspicious and potentially malicious, targeting a full understanding of the malicious code in order to enhance existing security mechanisms or to design new counter-measures.

More precisely, malware analysis aims at identifying the malware's targets (a particular end-user, a company, any machine under a specific operating system, etc.), the actions it intends to perform to attack the targets (sensitive information leakage, encryption and ransom, etc.), the way it succeeds in bypassing the security mechanisms protecting the targets, the way it protects itself against malware detection engines.

Given the potential impact of a malware, it is crucial that the analysis takes place as quickly as possible. With respect to this perspective, scientific contributions on malware analysis are twofold. First, some approaches focus on automatic classification. Their goal is to distinguish benign code from malicious code and then to classify the malicious code into one of the known families. Significant efforts have been made in this area that have permitted to decrease the human workload by reducing the number of samples to be analyzed manually. Remaining samples are generally either issued from unknown malware or too protected to be automatically processed. Other approaches thus aim to help the experts to reverse engineer and understand the malicious code.

An analysis can be performed statically, without executing the malware. It offers several benefits. Firstly, static analysis is safe for the host-architecture as the malicious code is not executed. Secondly, static analysis brings insights about all possible executions of the code and thus of about all possible behaviors of the malware.

Unfortunately, since malware authors are aware that their code will probably be confronted with a static analysis, they use various techniques to make static analysis and reverse engineering much harder. They use obfuscation techniques like packing, control flow flattening, or opaque predicates. All these techniques make the control flow graph computed from the malicious code irrelevant for most static analyzing tools.

An analysis can also be performed dynamically, which means that the malicious code is executed as far as possible. The main goal of this kind of analysis is to observe the real and concrete effects of the attack on its target. A difficulty here is to trigger the execution of malicious part of the code.

4.5.2 Malware detection

We distinguish here two kinds of malware. Regular commercial-grade malware intends to cause damage to an important population of end users as quickly as possible and can be found on popular platforms such as Windows or Android. By contrast, target-specific malware intends to cause damage to specific platforms, such as the Stuxnet worm that targeted programmable logic controllers of Iranian nuclear plants. Target-specific malware are built with great care and are therefore especially hard to detect [6]. In addition, being more stealthy, they are often only tardily identified, when external symptoms reveal their existence: Stuxnet was revealed in 2010 but it is thought to be have been deployed in 2005 or even earlier.

Regular, commercial-grade pieces of malware were initially and mostly developed to target the Windows operating systems and applications, as it was the most widely used operating system. Nevertheless, during the last ten years, Android has also become a popular target for malware authors. A piece of malware may infect its target from different entry points, such as user-downloaded applications, visited websites, or email attachments, and can often spread through different networks. The detection of malware usually occurs during the scan of either network traffic (files attached to mail, for example) or machines. In both cases, the detection engine compares the data to analyze with a database of malware signatures (known features that are symptomatic of malware). The main challenge consists in building and maintaining an up-to-date signature

6. The former Carte team lead a deep analysis of Duqu (https://en.wikipedia.org/wiki/Duqu), one of the most sophisticated malware in 2011, and developed a malware detector based on morphological analysis, a technique based on control flow graph comparison. The activity on malware analysis is carried on in the Carbone team of LORIA.

Detecting unwanted network behavior – © Inria / Photo C. Morel

database. This is a cat and mouse game: on the one hand, malware authors try to avoid detection as long as possible while minimizing the production code effort, and, on the other hand, defenders have to produce malware signatures that are as accurate and complete as possible by deep analysis of known malware, as explained in the previous section.

Malware authors usually build on malware kits and additional armoring techniques to generate new executable files based on the same malicious code. This way, a huge number of variants can be generated from a single, original sample. For instance, new variants of Cerber are generated every 15 seconds. The detection of a new malware or a new variant of a known malware remains an open problem. A recent advance is to use behavioral approaches, instead of static signatures, by monitoring any deviation from a model of the normal behavior, for instance by modeling information flow at the operating system level.

Starting from around 2012, scams relying on ransomware have grown internationally. A ransomware is a specific type of malware that restricts access to a computer system or to its hosted data (classically by enciphering the data). A ransomware requires the user to pay a ransom to the attacker to remove the restriction. Emergence of ransomware are probably linked to the Bitcoin system, which permitted anonymous payoffs and made massive attacks profitable.

[Inria teams] Malware analysis and detection

↗ The **CIDRE** team works on malware analysis, malware detection and malware de-obfuscation. Regarding malware analysis they target the Android environment. The idea is to use information flow tracking to analyze the malware behavior and, potentially, generate a signature of the malware based on its real activities (i.e., engendered information flows). Like all dynamic analysis, this work is relevant only if the malicious part of the code is really executed. The team thus focuses on the automatic triggering of malicious code. The malware detection tool monitors the usage of filesystem data to check for data deviations with respect to their normal use. Moreover, a postmortem action using an unclassified Machine Learning algorithm provides clues to clearly identify the detected malware. The work on malware de-obfuscation aims at circumventing protections of the malware itself against static analysis.
The LHS Rennes collects ransomware (Malware'O'Matic platform). As this kind of malware has a short lifespan, this requires to periodically renew the database and verify which ransomware are still alive.

4.6 Reaction to detected attacks

[Summary]
Ideally, intrusion and malware detection, as well as alert correlation should lead to the detection of all attacks without false alerts. Therefore, the next obvious step is to respond (possibly automatically) to the detected attacks through appropriate actions: modification of the security policy, new configurations of existing security mechanisms, implementation of new security mechanisms, patch deployment, etc. Of course, it is important to prevent the countermeasures from having similar or even worse consequences than those of the attack itself.

Ideally, alert correlation should lead to the detection of all attacks without false alerts. Therefore, the next obvious step should naturally be to respond (potentially automatically) to the detected attacks. Automatic response is also made necessary by the speed with which an attack may propagate, and the considerable damage it could make before manual response. Considering that the security policy has been violated although preventive mechanisms have been used to enforce this policy, two levels of reaction can be considered: (1) the attack may have succeeded because the policy was incorrect, in which case the policy must be amended, and new configurations of existing security mechanisms or even new security mechanisms must consequently be put in place; (2) the attack may also have succeeded because the enforcement of the policy was incorrect,

in which case configuration errors of the security mechanism must be identified and corrected. Using formal methods can help in guaranteeing that the security properties requested by the policy are effectively insured at the policy level and at the enforcement level. Of course, it is important that the countermeasures (modification of the policy, new configurations, new security mechanisms, patching, etc.) are not similar or worse than those of the attack itself. For example, when trying to stop a DDoS attack, legitimate packets may also be discarded: the service is then unavailable to legitimate users, which was the very objective of the attack.

[Inria teams] Reaction to detected attacks

↗ The **CTRL-A** team targets the relatively little-studied topic of models and control techniques for the automated reaction to attacks. The team uses detection information to identify the appropriate defense and repair actions, so that the system can remain operational, entirely or in a degraded mode. In terms of Autonomic Computing, this capacity is referred to as self-protection.

↗ In reaction to DDoS attack, legitimate packets may be discarded. The service may then be unavailable to legitimate users. This is why the **RESIST** team proposes scattering the DDoS traffic in time and space: by introducing voluntary delays and longer routes, whereas performance can be degraded, service is not discontinued.

5

Privacy and personal data protection

Privacy is often defined as the ability for individuals to control their personal data and decide what to reveal, to whom, and under what condition. There is however no single definition since the notion of privacy remains intimately linked with our cultural roots. For instance, the notion of personal data, the cornerstone of privacy, has received several definitions, not always compatible, depending on countries. In France, *personal data* is defined by the "Loi Informatique et Libertés" from 1978, as information that can be directly or indirectly linked to a person, by the data controller (in charge of data processing), or any third party, using any type of means. Therefore, it has a very broad scope, irrespectively of the nature of data. Additionally, *Sensitive Personal Information* is personal data related to such domains as health, politics, religion, or sexual orientation, that cannot be collected and processed except under well-defined situations. These notions and the associated obligations constitute the cornerstone of the French and European regulations (e.g., through the European General Data Protection Regulation, GDPR), as will be discussed in this chapter. The notion of *Personally Identifiable Information* (or PII), widely used in the U.S.[1], is close to that of personal data as defined above, but is not equivalent.

Privacy considerations have become a central topic in our connected world. Several domains, as yet unaffected by this trend, will soon generate huge amounts of personal and sometimes sensitive data, without leaving users any option to opt out. Privacy is therefore a key question, as important as security.

Being naturally multi-faceted, the work on privacy encompasses several dimensions:

- part of it is *legal:* harmonized rules are needed, that apply in the largest geographic area possible, in order to favor good practices and ban the others. In order to be applicable, guidance may be needed to implement these harmonized rules and this can be really challenging;
- part of it is *technical:* advanced privacy tools are needed in both theoretical and applied areas. They can help analyzing and improving existing systems or help designing privacy preserving systems from scratch;
- part of it is *economic:* understanding the underlying ecosystem is essential, since it often determines the practices in terms of personal data collection and processing. A sustainable ecosystem respectful of European regulation in terms of data protection is needed;
- part of it is *cultural:* the peoples of different geographical areas can have different approaches to privacy, due to their cultural roots, and these differences impact the local regulation in terms of personal data;
- finally, part of it is *sociological:* the end user is often inclined to declare herself concerned by privacy while at the same time behaving in an opposite manner.

1. http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-122.pdf

This well-known "privacy paradox" highlights the need for sociological studies to better understand human behaviors in this domain and potentially improve awareness and practices.

This chapter essentially covers the technical aspects and the legal dimensions up to a certain point, since the other aspects are out of Inria's research scope. We first present the high-level principles and regulations around privacy. We then present tools and privacy enhancing technologies. Finally, we discuss privacy leaks on existing systems, where personal data is either deliberately communicated by the user or collected without the user's knowledge.

5.1 Privacy principles and regulation

[Summary]

To take into account the major changes that took place during the last decade in terms of collection and use of personal data, the European Union adopted the General Data Protection Regulation ("GDPR") that came into effect in May 2018. The main change is the emphasis put on the responsibility of the data controllers, i.e., the organizations processing personal data, as well as their sub-contractors, if there are any. Any data controller must conduct data protection impact assessments, implement privacy by design and be accountable. If the impact assessment indicates that the processing is likely to severely impact the rights and freedom of physical persons, the measures taken will have to be strengthened. The rights of a data subject are also strengthened with better information and control over her data, following the user empowerment philosophy.

However, the GDPR provides very little guidance about the effective implementation of these concepts. An interdisciplinary work is needed to reduce this gap between legal and technical instruments, for instance by defining rigorous privacy risk analysis and privacy by design methods, or by defining techniques to strengthen accountability, transparency and enhance the user's control over her personal data. In any case, privacy comes with a price since tensions exist with several other considerations and privacy is sometimes regarded as a limiting factor.

A certain number of fundamental principles and legal considerations govern privacy. This section discusses them, in particular those associated to the new European regulation (GDPR) and other legal texts (e.g., the ePrivacy regulation that particularizes and complements the GDPR).

5.1.1 Tensions between privacy and other considerations

Privacy comes with a price. In a context where many commercial services depend on personal data (e.g., many free services are supported by targeted

advertising), where big data can offer highly valuable services (e.g., the study of the flu virus propagation through the analysis of medical acts), where several countries deploy mass-surveillance systems meant to help fight terrorism, where a certain form of user traceability is needed in order to enable respectful relationships between citizens, privacy can be regarded as a limiting factor. This is the sign of a fundamental tension between privacy, that requires to minimize personal data collection, and other considerations like utility, security, or accountability, where the higher the volume and accuracy of data, the better. A compromise is therefore needed and the idea of finding an appropriate balance is central for instance to data protection regulations. This balance of course heavily depends on cultural aspects, hence the importance of having a European regulation in the domain in order to preserve our sovereignty. The following sections will address and illustrate this tension according to several angles.

5.1.2 Evolution of the regulatory framework

The notion of privacy is complex and multifaceted. In addition, its perception evolves through time and space and is affected by the adoption of new technologies. To take into account the major changes that have taken place during the last decade in terms of collection and use of personal data, the European Union adopted in 2016 the "General Data Protection Regulation" (or GDPR) taking effect in May 2018, throughout the State Members in a uniform manner.

The biggest shift introduced by the GDPR is the emphasis put on the responsibility of the data controller (i.e., the private or public organization or association processing personal data) as well as its sub-contractor (if there are any). Any data controller must:

- conduct data protection impact assessments;
- implement privacy by design;
- and comply with the accountability principle.

If the impact assessment indicates that the processing is likely to impact the rights and freedom of physical persons severely, the measures taken will have to be strengthened. The rights of the data subjects are also strengthened in order to improve their information and control over their personal data, following a user empowerment approach. For instance, the data controller must be in position to prove they have obtained explicit user consent, users have the "right to be forgotten" on search services provided in Europe, a new data portability right is added in order to enable a user to switch to a different platform while reusing their data, and children under the age of 16 are better protected.

However, the GDPR provides very little guidance about the effective implementation of these new provisions and some of them raise a number of technical challenges. Before describing research on privacy enhancing technologies in

§5.2, we provide an overview of the challenges raised by legal requirements and interdisciplinary works to reduce the gap between legal and technical instruments.

5.1.3 Data Protection Impact Assessment (DPIA)

Data Protection Impact Assessments (DPIA), or simply Privacy Impact Assessments (PIA), are used by organizations to assess any privacy issue that might arise when developing new products or services that involve the processing of personal data. Conducting a DPIA is made mandatory in Europe by the GDPR for certain categories of personal data processing. Beyond legal requirements, conducting a DPIA is in the interest of any organization to ensure that privacy risks are properly understood and addressed before deploying any new product or service. There is already a large body of contributions on DPIAs and a number of DPIAs for specific products have been published. The Commission Nationale Informatique et Libertés (CNIL), the French data protection agency, has also recently released a tool to help data controllers prepare a DPIA[2].

All these contributions are very useful to define the DPIA process (including planning, stakeholders consultation, resource allocation, or audits) and its main goal (evaluating the likelihood and severity of privacy threats). However, they do not define very precisely how the technical part of the DPIA, the privacy risk analysis, should be performed.

There are of course a number of commonalities between privacy and security risk analyses. However, privacy is a more complex and multifaceted concept aiming at the protection of people (i.e., individuals, groups, and society as a whole) rather than resources or organizations. These dimensions must be considered in a privacy risk analysis, especially the notion of privacy harm that has been extensively discussed by lawyers.

Even if privacy risk analysis frameworks are being defined, work remains to be done. First, dedicated privacy risk analysis frameworks for specific application areas would be very useful in practice. On the theoretical side, it would also be interesting to establish formal links with privacy metrics. Last but not least, beyond legal compliance, a great benefit of a privacy risk analysis should be to provide guidance for the design of a new product, following the privacy by design approach.

5.1.4 Privacy by Design (PbD)

The philosophy of privacy by design, made mandatory by the GDPR, is that privacy should not be treated as an afterthought but as a first-class requirement in the design of any system. However, from a technical standpoint, it remains a challenging endeavor:

2. https://www.cnil.fr/fr/PIA-privacy-impact-assessment

- privacy includes a variety of dimensions (such as collection limitation, data quality, purpose specification, use limitation, or security) which are generally not defined very precisely;
- then, these requirements may seem to be in tension with other requirements such as functional requirements, ease of use, performance, or economic viability of the product or service.

In order to implement privacy by design, a wide array of Privacy Enhancing Technologies (PETs) are available, as discussed below. Each PET provides different guarantees based on different assumptions and therefore is suitable in different contexts. As a result, it is quite complex for a software engineer to make informed choices among all these possibilities and to find the most appropriate combination of techniques to solve their own requirements. Solutions have been proposed in different application domains such as smart metering, pay-as-you-drive, or location-based systems but the next challenge in this area is to go beyond individual cases and to establish sound foundations and methodologies for privacy by design.

A formal framework, for example based on epistemic logic, can be helpful to express data minimization requirements as properties defining for each stakeholder the information that she is allowed to know or not. But conflicting requirements often have to be met simultaneously, for instance guarantees about the correctness of a computation's result. In fact, the tension between data minimization and correctness is one of the delicate issues to be solved in many systems involving personal data. A formal framework is also useful for reasoning about architectures. Its axiomatization can be used to prove that a given architecture meets the expected privacy and integrity requirements.

5.1.5 Accountability

In line with previous regulations (e.g., the 1980 OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data), the GDPR integrates accountability as a principle that requires that organizations put in place appropriate technical and organizational measures to demonstrate their compliance with the regulation.

A distinction can be made between three main types of accountability:

- Accountability of policy can be seen as the first level of accountability: the organization should be able to demonstrate that it has defined a clear and properly documented privacy policy;
- Accountability of procedures, which refers to the demonstration of organizational mechanisms such as documented processes to cope with user consent or to address complaints or personal data requests;
- Accountability of practice is the *a posteriori* demonstration of the effectiveness of the procedures' accountability. This is a proof that the privacy policies have been effectively met.

The first type of accountability is purely declarative and provides at best a form of legal guarantee (binding commitment). The second type adds guarantees at the organizational level but only the third type can deliver the full promises of accountability. However, focus has been placed on the first and second types of accountability so far, resulting in superficial guarantees.

For accountability to contribute effectively to the enhancement of privacy protection, it is necessary to be able to translate its general principles into practical measures and to consider its various dimensions. For instance, collecting and keeping detailed processing logs can contradict privacy if those logs contain personal data. An appropriate balance between these requirements must be found.

5.1.6 User empowerment through control and transparency

A service that is compliant with the legislation and for which the data controller is accountable, is not necessarily in line with user expectations. Empowering the user is required in order to gain user acceptance.

EMPOWERMENT THROUGH USER CONTROL

Privacy is increasingly seen as the ability for a user to control their personal data. However, even if this notion is predominant in the privacy literature and plays a central role in the GDPR, clear definitions are still missing. The word "control" is usually used in a very vague way in this context, both by lawyers and by computer scientists. For example, such notions as "access control" or "usage control" in computer science do not really encapsulate the intuition underlying the notion of control over personal data. This lack of precision may lead to misunderstandings and makes it difficult to check compliance. A multidisciplinary study of the notion of control as used by lawyers and computer scientists has led to identify three dimensions, corresponding to the capacities for an individual:

- to perform actions on their personal data;
- to prevent others from performing actions on their personal data;
- to be informed of actions performed by others on their personal data.

On the practical side, two main conditions must be met to make it possible for data subjects to exercise control over their personal data:

- the user must be properly informed about the collection of their data, its purpose, the entity collecting the data, and for instance the retention delay (transparency);
- the user must be able to express their choice to have their data collected or not for a given purpose and have guarantees that this choice is actually followed (consent).

In fact, the explicit consent of the data subject, which is a cornerstone of most data protection regulations, is a typical example of a legal requirement that is very difficult to put into practice. This is the case with Internet of Things (IoT)

where many data communications occur without user notice, or social networks for instance, and several proposals have been made to help a user to exercise their control.

EMPOWERMENT THROUGH TRANSPARENCY

Transparency is an essential concept to design privacy preserving systems and services. The data controller should provide clear and complete information on what information is collected, how often, for what purpose, how data is processed, how data is stored (where, how long, with what security), and whether data is likely to be communicated to third parties. The same questions recursively apply to third parties to whom data may be transferred.

Transparency is a challenge in various domains where data collections happen in an invisible manner. This can be the case with Internet of Things (IoT) systems that measure and collect continuously personal data. The information provided to the data subjects should be as visible and intelligible as possible (which excludes simple signs on walls that generally remain unnoticed). A new design space for effective privacy notices is required, which is an active research topic in the privacy area.

Beyond the transparency of data collection, a new major challenge is the transparency of algorithms. This need for more transparency is another illustration of a legal requirement (in the GDPR and also in the law for a digital Republic[3] adopted in France in October 2016) which raises many technical challenges and can be a source of interdisciplinary research topics.

Algorithm transparency is central in many automated systems with societal impacts (e.g., with an automated assignment system, once all candidates have posted ordered preferences): the question of a possible algorithmic bias, whether intentional or not (e.g., bug), is unavoidable. Transparency of algorithms is necessary to enable third parties (including citizens) to analyze their internal behavior. This is the role of the TransAlgo[4] French initiative led by Inria.

Since transparency requires detailed information to be communicated to the user, the way it is achieved is non-trivial. The usual approach, through the General Terms and Conditions of Use, is usually far from satisfying, being non-user-friendly: this legal document is often meant to protect the company rather than informing the user. Research is in progress on analyzing practices, on trying to have a standardized format understandable by everybody (like Creative Commons did in a different domain) and perhaps amenable to mechanized treatment, and on the impacts of transparency on user behavior.

3. Loi pour une Republique numerique
https://www.economie.gouv.fr/republique-numerique.
4. https://www.inria.fr/en/news/news-from-inria/transalgo

[Research challenge 7] Understanding privacy and deriving practical tools

Understanding privacy principles and regulations is the foundation of any activity in privacy. Although this is not a new domain (e.g., the "Loi Informatique et Libertés" was adopted in 1978), this area has recently experienced major evolutions with the new GDPR European regulation and at the same time new opportunities to collect personal data. As a consequence, understanding the concepts and the regulation is a first necessity. Being able to derive practical tools is another one: even though the GDPR promotes several concepts and goals, it provides little guidance about the effective implementation of these new regulatory provisions.

In particular the GDPR introduced the right to data portability whereby a user can retrieve their data in a human readable and machine portable format. This right opens new research areas around individualized management and control over one's personal data. The goal is to empower citizens to leverage their personal data for their own good, which calls for secure, extensible, and sovereign personal cloud platforms, three conflicting goals that open new research challenges *(see e.g., §5.2.4)*.

[Inria teams] Privacy principles and regulation

↗ The **CIDRE** team works on privacy policies and the right to be forgotten in collaboration with lawyers.

↗ The **INDES** team works on the relationships between web tracking and the ePrivacy Regulation, in relationships with lawyers. In particular the team evaluates the privacy impacts when the Regulation doesn't require user consent for tracking and builds tools that detect violations of the Regulation.

↗ The **PETRUS** team works on control over personal data and data minimization and has strong collaborations with research groups in other disciplines such as economy, law and social sciences.

The team also contributes to enhance individuals' control over their personal data from an architectural point of view. This is the case of PlugDB, a secure personal server that allows individuals to exercise control over their personal data, while preserving durability, availability, and sharing.

↗ The **PRIVATICS** team works on most subjects, with a strong emphasis on interdisciplinarity through collaborations with lawyers and economists. For instance, the team has contributed to a framework and methodology for conducting a privacy risk analysis in a rigorous and systematic way, compatible with most DPIA recommendations. Concerning privacy by design, the team has proposed a formal framework based on epistemic logic that enables one to express data minimization requirements. This framework has been used to formally compare different architectures for biometric access control.

Concerning accountability, the team has defined a set of practical measures to be taken at each phase of the personal data life cycle, from collection to deletion, including storage, usage, and forwarding to third parties. On the formal side, the team has proposed a framework based on privacy friendly logs, showing that compliance can be checked based on logs that do not contain personal data.
Concerning user control, after a multidisciplinary study of this notion as used by lawyers and computer scientists, a formal model has been derived to formally characterize each type of control.
On the particular case of IoT and user control, the team has proposed an architecture based on "Privacy Agents" implementing the choices of the data subject, expressed in a restricted natural language that can be easily understood by non-experts.

5.2 Privacy tools

[Summary]
Tools for different types of publics, from a data protection officer down to an end-user, are needed in order to transition from the high-level principles of GDPR to privacy compliant products and services. Some of them focus on such principles as data protection impact assessment, privacy by design, and accountability.
Other tools are meant to anonymize a database before releasing it for open data access. However, this is a complex task that requires an appropriate trade off to be found: increasing privacy usually reduces the utility of an anonymized database. From this point of view, the differential privacy concept turned out to be a key tool in order to provide provable privacy guaranties.
The advent of personal clouds, meant to give users a full control over their own data, is another major tool for user empowerment.
Finally, techniques have been designed to provide unlinkability, i.e., the guarantee that no one can link several uses of a service by a given user. This is particularly important for systems that embed an RFID token, like electronic passports. Concerning communications, the Tor system tries to guarantee anonymous communications (no one can identify the source of a packet), although in practice this is less usable.

5.2.1 Tools related to DPIA, privacy by design, and accountability

Several tools have been designed in order to address the needs for data protection impact assessment, privacy by design, and accountability. Some of them are respectively discussed in §5.1.3, §5.1.4 and §5.1.5.

5.2.2 Database anonymization: a necessity for open-data and big-data

A strict regulation applies to any data controller in charge of a database containing personal data. Evading this regulation is possible by anonymizing this database, since the resulting database no longer contains any personal data. For instance, this is a prerequisite for releasing a public dataset in the context of an open-data initiative.

Data anonymization consists in altering the dataset in order to remove any information that could be used to re-identify any participant of the dataset or to infer personal attributes. This is not an easy task. A first reason is that anonymization is intrinsically complex and domain dependent: there is no universal solution. Certain types of data, such as mobility traces, are highly unique and therefore identifying. For instance, the knowledge of four spatiotemporal points may be enough to uniquely identify 95% of the individuals in a large cell phone operator mobility database [dMHVB13].

A second reason is that anonymizing a database is one thing, preventing re-identification through side information (the "inference" problem) is another one. The AOL de-anonymization of pseudonym No. 4417749 [BZ06] is famous from this point of view. By analyzing the detailed records of searches she made, her identity has quickly been found. In general, replacing names by pseudonyms is now known to be highly vulnerable to re-identification attacks.

Last but not least, an appropriate trade-off between privacy and utility is needed (see §5.1.1). Guaranteeing meaningful privacy requires the distortion of the original dataset which mechanically yields imprecise, coarse-grained knowledge even about the population as a whole.

Data anonymization can be achieved through various types of approaches. In 2014 the G29 group, gathering the European data protection agencies, published a technical document on the topic [Art14]. This document discusses the effectiveness and limits of various techniques: permutation, differential privacy (see §5.2.3), aggregation, k-anonymity, l-diversity, and t-closeness. In terms of privacy guaranties, these techniques can be classified into two categories: syntactic and semantic privacy models. Syntactic models focus on syntactic requirements of the anonymized data (e.g., with k-anonymity each record should appear at least k times in the anonymized dataset), without any guarantee on what sensitive information the adversary can exactly learn about individuals. Semantic models, on the other hand, are concerned with the private information that can be inferred about individuals using the anonymized data as well as perhaps some prior (or background) knowledge about them. A higher level of privacy can be expected from the semantic models. This is the case of differential privacy that is discussed in the following section.

[dMHVB13] Yves-Alexandre de Montjoye, Cesar A. Hidalgo, Michel Verleysen, and Vincent D. Blondel. Unique in the crowd: The privacy bounds of human mobility. *Scientific Reports*, 3:1376, 03 2013.
[BZ06] M. Barbaro and T. Zeller. A face is exposed for aol searcher no 4417749. New York Times, August 2006.
[Art14] Opinion 05/2014 on anonymisation techniques, April 2014.-

5.2.3 Differential privacy

Differential privacy [DMNS06] was originally proposed in the area of statistical databases and it is nowadays one of the most successful approaches to privacy. The goal is to protect an individual's data while publishing aggregate information about the database. This is obtained by adding controlled noise to the query outcome in such a way that the data of a single individual will have a negligible impact on the reported answer. Differential privacy has several advantages: (1) it is independent from the side-information of the adversary, meaning one does not need to take into account the context in which the system will operate; (2) it is compositional, i.e., if we combine the information that we obtain by querying two differentially-private mechanisms, the resulting mechanism is also differentially-private; and (3) differentially-private mechanisms usually provide a good trade-off between utility and privacy.

A successful variant, called local differential privacy, has the advantage of requiring no trusted third party: users obfuscate their personal data, adding noise by themselves, before sending it to the data collector. Local differential privacy is particularly suitable when data is collected for statistical purposes, as it usually achieves a good trade-off between privacy and utility. Local differential privacy has the same advantages as differential privacy (independence from side-knowledge and compositionality). It had a considerable impact after large companies like Apple and Google adopted it for privacy preserving data collection systems (e.g., Google uses a particular implementation in the RAPPOR crowdsourcing technology).

There are many cases in which the data domain features a notion of distance (e.g., location, energy consumption in smart meters, or age and weight in medical records). Then the privacy/utility trade-off can be greatly improved by exploiting the concept of approximation intrinsic in the notion of distance. The idea is to allow two values to become more and more distinguishable as their distance increases so that more accurate statistics can be made.

[Research challenge 8] Open data and anonymization

Open data initiatives may sometimes mean releasing databases that contain sensitive, personal information. To ensure privacy of the individuals, data need to be anonymized. Robust anonymization, that effectively resists de-anonymization attacks, is an active and hot research topic. If differential privacy has become a key scientific tool to achieve provable anonymization guarantees, challenges remain on its application, for instance in order to improve the privacy/utility trade-off.

[DMNS06] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography (TCC'06)*, 2006.
https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2014/wp216_en.pdf

5.2.4 Empowering users with personal clouds

The time of individualized management and control over one's personal data is upon us. Thanks to smart disclosure initiatives and the right to data portability in the GDPR, we can access our personal data from the companies or government agencies that collected them, thus allowing the creation of cross-domain personal data repositories. Concurrently, Personal Cloud solutions, also called Personal Information Management System (PIMS) or Personal Data Server (PDS), are flourishing (see, for instance, the French startup Cozy Cloud[5]). Their goal is to empower us to leverage our personal data for our own good, opening the way to new value-added services by crossing data issued from different data silos or by sharing our data for social/societal benefits (e.g., contribute to an epidemiological study, compute queries based on shared data within communities of users).

However, managing our own personal data constitutes a considerable burden. We must ensure the security of the data we gather and manage the disclosed data and control its usage. We inherit the combined responsibility of an information security expert and a database administrator. Therefore, personal cloud providers propose solutions to manage personal data on behalf of their customers, thus creating genuine honeypots (centralizing huge amounts of personal data belonging to millions of individuals). Thus, paradoxically by empowering users, smart disclosure and personal clouds push them towards even more delegation over even more data, thus exposing them to a larger risk than ever.

Dedicated personal cloud prototype – © Inria / Photo C. Morel

5. https://cozy.io/en/

To escape this paradox and truly achieve the empowerment of the user, a set of expectations should be fulfilled: (1) sovereignty: the system must offer the user the ability to exercise her data disclosure decisions under her own authority and without any form of delegation; (2) security: the system must provide tangible guarantees about the enforcement of these decisions whatever the type of attacks or misuse the system could face not only to the personal cloud holder but also to the other users and to third parties; and (3) extensibility: the system must not impede the development of new services using the data of a single individual ('Personal Big Data') or of large groups of individuals ('Big Personal Data'). However, these statements introduce two forms of tension, a first one between sovereignty and security: to reconcile the absence of delegation to central IT experts or administrators with high security guarantees; and another one between security and extensibility: to reconcile security, which calls for closed systems, with the need for extensibility to support new data services, which by definition are not fully trusted. Deriving an architecture to fulfill these three requirements altogether is an intrinsically difficult issue.

Another important problem linked to users' empowerment is to allow for novel big personal-data applications (e.g., participatory sensing, epidemiological studies, and personalized recommendation systems). Distributed personal-data processing is not a new issue. However, the personal cloud is distributed at the individual level and is expected to scale up to nationwide populations. In this context, the first issue is to prevent or minimize the effect of personal data leaks during computations performed at such scale. Also, decentralization emphasizes the central role of the individual in the architecture. This calls for the emergence of new forms of decentralized computations where individual profiles and individual privacy settings are integrated by construction.

Beyond the privacy and security perspective, many important research problems around personal clouds remain to be addressed: foundational and systems aspects of complex data management, especially with human-centric data, and orchestration of queries to the various services.

5.2.5 Privacy preserving protocols and communication technologies

UNLINKABILITY PROTOCOLS

According to the ISO/IEC 15408-2 standard, unlinkability, also referred to as untraceability, *"ensures that a user may make multiple uses of resources or services without others being able to link these uses together."* Of course, this notion of unlinkability applies both in the physical world (i.e., tracing a person) and in the virtual world (i.e., linking transactions of a person).

Unlinkability is getting increasingly important with the widespread use of RFID tokens for authentication. If the authentication protocol guarantees unlinkability, it should indeed be impossible to place an unauthorized RFID reader that can decide whether a given token is the same as a previously seen token. For instance, since RFID tokens are used in electronic passports, an attacker who eavesdrops a first legitimate protocol execution for a target, say Alice, should not be able to setup a reader that distinguishes Alice's passport from others. Similarly, the authentication protocols implemented in 3G/4G/5G mobile telephone networks prevents an eavesdropper, other than the operator, to link different communications made by the same device.

Similarly, digital transactions may require unlinkability. A typical example is a blockchain based electronic currency where the ledger is completely public. Such systems generally provide pseudonymity (i.e., use of a pseudonym rather than a real identity). However, by linking several transactions made by the same user, user profiling is trivial and then reidentification may be possible with side information. That said, Zerocash [6] is an example of an electronic currency that provides unlinkability guarantees.

More generally, authentication and authorizations are sometimes required while guaranteeing anonymity and unlinkability. In that case, Direct Anonymous Authentication (DAA) protocols provide anonymous authentication tokens, allowing remote anonymous authentication.

Finally, Fingerprinting is a direct threat to unlinkability. Browser fingerprinting is discussed in §5.3.4. However, even devices may be fingerprinted: for example, the length of the encrypted message containing personal data, in particular the JPEG photo of the electronic passport can be effectively used for tracking [7].

ANONYMIZED COMMUNICATIONS

Tor is an anonymizing system that aims at preserving the anonymity of Internet users. This is achieved thanks to so-called "onion routing": instead of direct connections from a host to a destination, an application-level virtual circuit is created that goes through a certain number of relaying nodes spread throughout the world and chosen randomly. Traffic is encrypted by the client as many times as there are relays in this virtual circuit. Then each node decrypts the outer part of the packet received, thereby revealing the IP address of the next relay, and forwards the packet to this relay. Thereby, Tor prevents any eavesdropper to identify the source and destination hosts by looking at the packet source and destination IP addresses.

However, even if theoretically interesting, this solution is severely flawed for the general public because of the amount of knowledge required to correctly operate on top of Tor. Indeed, several types of de-anonymization attacks against Tor are regularly found.

6. http://zerocash-project.org/
7. https://www.inria.fr/centre/nancy/actualites/securite-des-donnees-les-passeports-biometriques

[Inria teams] Privacy tools

↗ The **COMETE** team works on differential privacy and its variants, in particular when a notion of distance exists, as well as techniques to measure the utility of the result. For instance, the team has implemented d^x-privacy and works on a tool to retrieve as accurately as possible the original distribution from the noisy one. The team also develops a tool to measure the result's utility in terms of the quality of the true distribution's approximation. Finally, the team also works on the unlinkability definition and its formal verification.

↗ The **DIANA** team has worked on de-anonymization attacks on Tor.

↗ The **PESTO** team works on the unlinkability definition and its formal verification.

↗ The **PETRUS** team works on solutions and architectures for personal cloud systems and on trusted environments for privacy preserving decentralized computation. In particular the team works on a new reference architecture where potentially complex manipulations of personal data rely on Trusted Execution Environments (TEE). The objective is to limit the side effects in terms of data leaks since only expected results are declassified to third parties, without any direct access the raw data.

↗ The **PRIVATICS** team works on various tools for privacy risk analysis, privacy by design, accountability, and user control. The team also works on different aspects and models for data anonymization. The team developed several anonymization schemes for mobile traces and set-value datasets under the k-anonymity or differential privacy models. The team also worked on a novel technique for privately releasing high-dimensional datasets using generative neural networks. The idea is to produce a synthetic dataset that resembles the original training data as much as possible while complying with privacy requirements. Differential privacy has been applied to protect the user's privacy in several domains, including smart meters, sequential data publication, generative neural networks, and data storage in Bloom filters.

↗ The **VALDA** team studies both foundational and systems aspects of complex data management, especially human-centric data, in the context of personal clouds. They focus on personal clouds under the angle of data integration and service orchestration.

↗ The **WIDE** team works on the use of gossip protocols for privacy preserving decentralized computations and seeks to extend this work to the protection of privacy for decentralized machine learning systems.

5.3 Privacy analysis of existing systems

[Summary]

Our connected world is at the origin of many privacy leaks, and as time passes, domains that are today leak-free will also be impacted. Some of the leaks are deliberate. This is the case with social networks where users massively share (sometimes overshare) personal data. The consequences are numerous for the user's privacy. For instance, the profiling it makes possible enabled the rise of dedicated companies that commercialized services meant to influence users (e.g., their votes).

Geolocation information is another type of information that is shared with either the tacit or the informed consent of the user. However, the record of a user's location over time is particular in the sense that a lot can be inferred, from home and work locations to sensitive personal information (e.g., religion if she regularly goes to a place of worship). In order to benefit from geolocalized services without leaking too much information, several solutions have been designed. For instance with spatial obfuscation, the position precision is reduced by reporting a zone. Geo-indistinguishability, which leverages on differential privacy, is a promising technique to achieve spatial indistinguishability with additional good properties.

The user can also be invited to provide biometric features, such as fingerprints, in order to permit their identification and authentication, in case of access control systems or secure national identity documents. However, these highly discriminating and stable features (they cannot be changed) create major security and privacy risks. Biometric systems must be carefully designed and, in order to investigate design options, various frameworks have been developed to define privacy architectures, to formally reason about them and to justify design choices in terms of trust assumptions.

But very often, privacy leaks happen without the user's consent. This is the case when browsing the web. Every visit to a web site can trigger a wide variety of hidden data exchanges across multiple tracking companies. Information can then be used for targeted advertising, but also to discriminate users (e.g., through customized prices) or worse. Several privacy solutions have been proposed, from protective regulations (e.g., with the GDPR) to client-side mechanisms like advertisement and tracker blocking tools. However, the domain is in constant evolution, new techniques appearing to better protect users and, simultaneously, to improve tracking within a browser or across devices.

With the advent of smartphones and IoT, privacy leaks have reached an unprecedented level in volume and precision, both within the digital and physical worlds, and often without the user's knowledge. This trend will be reinforced in the coming years, encompassing new domains. The goals of research in this domain are to analyze these systems, to give transparent information of hidden behaviors, to highlight good and bad practices, to propose methods likely to improve transparency and user control, and to encourage certain stakeholders to change practices.

As a common denominator of our connected world the Internet can also be the source of hidden privacy leaks. A first example is the wireless access network used by most devices. We have seen the rapid growth of cyberphysical tracking systems that analyze the Wi-Fi frames sent by a smartphone looking for a known access point. Research is needed to analyze these technologies and propose privacy preserving versions whenever possible. Another example is the DNS service used to map host names to IP addresses. A motivated attacker could eavesdrop DNS traffic (under certain circumstances) and potentially identify websites of interest. Unfortunately, the road is long before a privacy friendly version of DNS is standardized and deployed.
Finally, the safe browsing techniques used to detect blacklisted URLs (e.g., known to contain a malware) are potentially intrusive: storing the whole list of blacklisted URLs on the host is impossible and relying on an external service too intrusive since this latter would collect all the visited URLs. Intermediate solutions are used that need to be carefully analyzed from a privacy point of view.

This section focuses on privacy leaks in various existing systems. It starts with leaks for which there is either a tacit or informed consent of the user and then presents data collections that take place without the user knowledge.

5.3.1 The visible side: the case of social networks

Social networks are places where users are incited to share personal data massively. Very often, personal information, and even sensitive personal information, is shared to a group of people that is significantly larger than what the user would think. Several reasons can account for this situation. First of all, the "Privacy Paradox" is a well-known phenomenon whereby users explain they are concerned by their online privacy but at the same time behave in the opposite way. Many users may not realize that the risks apply to them, particularly the youngest, for whom the risks are either theoretical or for those "who have something to hide". Or perhaps they consider that the reward obtained by sharing personal information is superior to the risks. Second, users may be too confident in the protection offered by the social network, omitting the small detail that significantly enlarges the group of people with whom information is shared. Third, the default settings and behaviors with respect to third parties may also be much more permissive than one could expect (e.g., Facebook changed default audience to "Friends only" in 2014, before that posts were "Public" by default). And finally, mistakes are always possible (e.g., the audience setup in Facebook posts is sticky, meaning that if a post is tagged for public audience, so will all the following ones until the user changes the audience back to the previous setting).

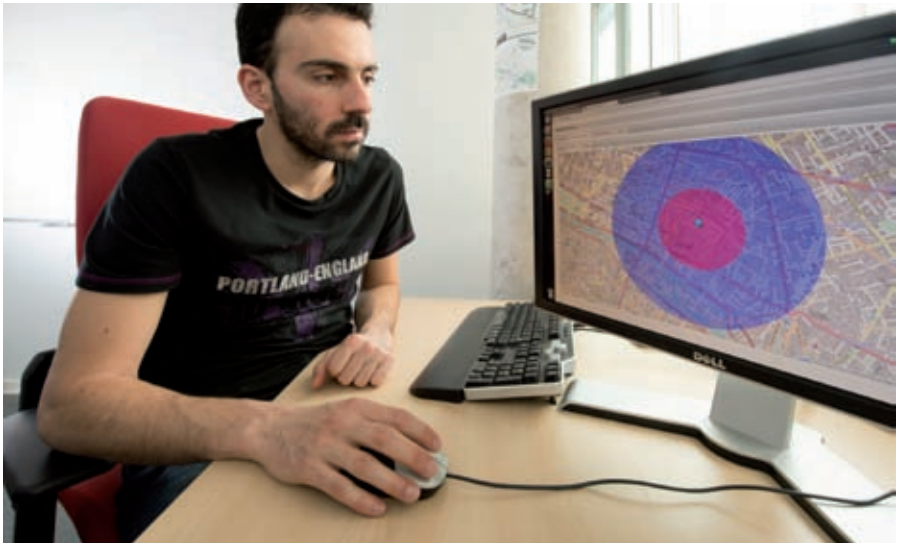
The consequences are numerous: identity theft, social pressure on users (in particular teenagers), sexual predation, unexpected consequences of shared

information (e.g., to a potential employer), targeted surveillance, or at the other extreme massive surveillance of citizens. Users are generally unaware of the precision of the story they tell about themselves. On 2009, *Le Tigre* published the life of Marc L*[8], the story of several years of this person's life, solely collected from various social networks. More recently, information extracted from social networks has been used by the *Cambridge Analytica* company to profile users and influence their vote via personalized messages[9].

Research is needed to better understand the tendencies, but also protect the users, in a context where data mining and machine learning have become extremely powerful tools.

5.3.2 The visible side: the case of geolocation information

Geolocation information is another type of information that is shared with either tacit or informed consent of the user. However, collecting the geolocation of a person over time raises major risks: additional personal data can easily be inferred (e.g., their home and work locations, or their habits), but also Sensitive Personal Information (e.g., health concerns if they go to an hospital or religion if they regularly go to a place of worship). Protection is therefore needed, for instance in order to benefit from geolocalized services without leaking too much information.



Geolocalisation anonymization – © Inria / Photo H. Raguet

8. http://www.le-tigre.net/Marc-L.html
9. https://ca-political.com/casestudies/casestudydonaldjtrumpforpresident2016

The main methods for the privacy protection of location data are divided in two classes: spatial cloaking and spatial obfuscation. In spatial cloaking, the goal is trace-anonymity in order to prevent the identification of an individual. Most of the methods of this class are based on group-anonymity, a very popular approach in the anonymity literature. The general idea is to make the traces of an individual indistinguishable from those of other individuals; this is typically achieved by reporting a cloaked area that is large enough to contain the group size necessary to meet the intended anonymity constraint. In order to limit the size of the cloaked area, some proposals have combined spatial cloaking with temporal cloaking as well. When the system uses pseudonyms, a necessity for certain applications, there is the risk of linkability between points belonging to the same user's trajectory. To solve this problem researchers have proposed the so-called mix-zones, which are zones in which many users meet and can get their pseudonym renewed, without the danger of being traced. All the above measures related to spatial cloaking need, of course, the intervention of a trusted party that acts as an *anonymity server*.

In the second class, spatial obfuscation, the goal is to address the problem of identifying the user's position. In general, privacy is preserved by reducing the position precision. This is done by reducing the granularity of the location information: the user reports a zone rather than the exact coordinates. An important advantage of this approach is that it can be done without the intervention of a trusted third party. This method however is not very robust, being subject to trilateration attacks: a user sending two consecutive signals from two different zones reveals that they are close to the border between them and with three consecutive signals from different zones would reveal their position quite accurately. People have therefore investigated more effective solutions for spatial obfuscation.

Geo-indistinguishability [ABCP13] is one of them. It extends local differential privacy to arbitrary metrics with the idea that the protection of the user's location increases exponentially as the distance from the real location decreases. Therefore, an attacker can determine that the user is in Paris rather than London, and be reasonably confident that they are in the Quartier Latin, but cannot tell where exactly in the Quartier Latin. Geo-indistinguishability inherits appealing properties: it is independent from the side knowledge of the adversary, it is robust with respect to composition, and it does not rely on any trusted third party. It can be implemented at the user's end simply by adding noise to the real location. A low-cost planar Laplace function can be chosen for that, which enables its use in computationally limited devices such as smartphones. Thanks to the above properties, geo-indistinguishability via the Laplace mechanism has been adopted in several tools for location privacy (e.g., LP-Guardian, LP-Doctor, and SpatialVision QGIS plugin).

[ABCP13] Miguel E. Andres, Nicolas E. Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer; Communications Security,* CCS '13, pages 901-914, New York, NY, USA, 2013.

5.3.3 The visible side: the case of biometry

Biometrics is a powerful technology to identify or to authenticate a person. Biometric features, such as fingerprints or iris, are stable over time and highly discriminating; these are key advantages for applications such as security or access control. However, from a privacy point of view, these advantages turn into drawbacks: because of their stability over time and because an individual cannot easily change their biometrics, the leak of biometric traits to a malicious entity gives rise to serious privacy risks, including tracking and identity theft.

Many techniques (like encryption, homomorphic encryption, or secure multi-party computation) and architectures have been proposed to take into account privacy requirements in the implementation of privacy preserving biometric systems. Some solutions involve dedicated cryptographic primitives such as secure sketches and fuzzy vaults, others rely on adaptations of existing cryptographic tools or the use of secure hardware solutions. The choice of particular techniques and the role of the components (like the central server, a secure module, terminal, or smart card) in the architecture have a strong impact on the privacy guarantees provided.

Considering the variety of options available and the complexity of these techniques, general frameworks have been proposed to define privacy architectures, to specify different options, to reason about them in a formal way and to justify their design in terms of trust assumptions.

Biometric systems are also increasingly used by states to secure identity documents. For example, the French government authorized in October 2016 the creation of a centralized file of "secure electronic documents" (TES). The main motivation put forward was the fight against identity fraud. However, the decree also authorized access to the database by a variety of police and officers. Several criticisms have been voiced concerning the risks that such a centralized file could pose to individual freedom and privacy. The strengthening of the means to fight fraud and criminality, and the requirement to protect privacy should not necessarily be incompatible. However, in order to be able to reach a decision on the benefits and weaknesses of a secure electronic document system, it is necessary to [CM17]:

- clearly define the desired functionalities and the advantages that can be expected from them;
- describe the technical solutions in a sufficiently precise way to enable their analysis;
- and rigorously analyze the risks of privacy breaches with regard to the expected benefits.

[CM17] C. Castelluccia and D. Le Metayer. Titres electroniques securises : la centralisation des données biométriques est-elle vraiment inévitable ? Note d'analyse Inria, Février 2017.

5.3.4 Hidden privacy leaks: the case of web tracking

The massive deployment of the Internet has quickly been accompanied with personal data leaks. Every visit to a web site can trigger a wide variety of hidden data exchanges across multiple tracking companies that each collect large amounts of data on users' preferences and habits. Information can then be used for targeted advertising, but also to discriminate against users (e.g., through customized prices) or surveillance.

Web tracking has been made possible thanks to the discreet addition of small components, called trackers, to webpages. Each tracker is owned by a third party, usually distinct from the web site owner, and it enables this third party to recognize users across the different websites that embed it. These technologies are roughly divided into stateful and stateless types. Stateful tracking techniques store information on the user's computer that can be retrieved later to recognize them. Third-party cookies are the most prevalent stateful online tracking technique. Advanced features, such as the ability to respawn cookies deleted by a user or the synchronization of cookies among different third parties, are often implemented in order to map and exchange user's profiles.

On the other hand, stateless tracking techniques allow third parties to recognize users , just by *fingerprinting* without storing anything. Collecting various pieces of information about the user's browser and operating system is sufficient to uniquely identify each browser. In 2010, Eckersley first demonstrated that the technology was highly effective through the Panopticlick project[10]. Recent works have demonstrated that today fingerprinting is as effective on mobile devices as it is on computers, in particular thanks to recent web technologies (e.g., HTML5 brought highly discriminating attributes) or websites where a user is connected.

These practices, source of major privacy leaks, raise two complementary and complex questions: how to detect them and how to protect oneself?

With stateful web tracking, explicitly blocking third-party cookies within the browser configuration or adding ad blocker extensions help a lot. Although not perfect, these techniques correctly detect and disable the most common and intrusive trackers [MHB+17].

Concerning protection against stateless web tracking (fingerprinting), disabling JavaScript is efficient but not practical. The automatic tracker detection question is complex and there is no precise methodology today. Most detection methodologies ignore many trackers because they only check access to fingerprinting-related APIs in the browser or apply very basic static analysis. Language-based security techniques can be used to go further: a monitoring is put in place to analyze how

10. https://panopticlick.eff.org/

[MHB+17] Georg Merzdovnik, Markus Huber, Damjan Buhov, Nick Nikiforakis, Sebastian Neuner, Martin Schmiedecker, and Edgar R. Weippl. Block me if you can: A large-scale study of tracker-blocking tools. In *2017 IEEE European Symposium on Security and Privacy, EuroS&P 2017, Paris, France, April 26-28, 2017,* pages 319{333, 2017.

much information is actually leaked from fingerprinting-related APIs and whether this information is sufficient to uniquely identify a user. Another approach consists in adding browser diversification at the virtual machine and API levels: by introducing enough noise during the fingerprinting process, trackers can be fooled.

The previous protection techniques were at the client initiative. Website owners may also be interested in protecting their users from web tracking (e.g., the GDPR makes website owners liable for the third-party tracking present on their websites). This is the goal of newly proposed architectures where additional servers automatically intercept and modify web requests thereby preventing third-party tracking.

Finally, it should be noted that the business model of web sites is often based on advertisements and blindly blocking everything harms content providers. Hence the question of user control and responsible choice: rather than trying to block all trackers, authorize some of them, for instance those present on web pages considered as less sensitive.

5.3.5 Hidden privacy leaks: the smart world

Beyond web tracking, the advent of intelligent and connected devices considerably expended the opportunities to collect personal data, both in volume and precision, encompassing domains that were up to now out of reach.

Smartphones have played a key role from this point of view. These personal assistants, easily personalized with applications, always connected, equipped with a large variety of high precision sensors, contain and generate a lot of information about our activities and centers of interest, both in the cyber world (Internet) and the physical world. Smartphones therefore have become ideal targets. The smartphone ecosystem consists of many actors, from application developers to advertisers. However, personal data collection is mainly orchestrated by the Advertising and Analytics (A&A) companies[11], also known as "third parties." In a world where free applications represent a large part of the offer, the A&A companies have developed small tracking software that developers are invited to integrate within their applications in order to monetize them. These trackers collect personal data from several billions application sessions every day and send them to the A&A companies in order to create user profiles whose accuracy keeps increasing every day.

Arguably, collecting personal data in exchange for free applications or services could be acceptable if this collection was documented in a privacy policy notice, respectful of the laws of the country where the user resides, and if it was implemented in a "privacy by design" manner with data minimization and accountability guarantees. But many studies report that the opposite is taking place. A&A companies tend to collect as much personal information as

11. http://www.mobyaffiliates.com/guides/mobile-advertising-companies/

technically possible[12]. However, major differences exist between the various operating systems (OS), depending on the decisions taken by the OS editor with respect to user tracking possibilities and user control (e.g., in 2013 Apple has been the first company to ban the access to stable identifiers and to replace them by a dedicated Advertising Identifier, under full user control).

Today the craze for "quantified self" wearable devices, smart home appliances, smart cities, and connected cars, or more generally "Connected Devices," enables the collection of personal data in new domains[13]. This is all the more worrisome as part of data being collected is Sensitive Personal Information and many connected devices remain highly vulnerable.

If the business model of companies selling connected devices is different from those of smartphone application developers, little is known about the actual practices in terms of confidentiality and data exchanges. Also, the privacy notice that should be provided with each connected device in order to inform the user is often missing or is unreadable for non-lawyers. The end-user is therefore prisoner of a highly asymmetric system.

Academic work is needed in these domains to understand the various facets of the problem, including the underlying business models. By giving transparent information of hidden behaviors, by highlighting good and bad practices, by proposing methods to improve transparency and user control, the ultimate goals of these works are to reduce the information asymmetry of the system, to empower users, and hopefully to encourage certain stakeholders to change practices.

5.3.6 Hidden privacy leaks: the case of the Internet

The Internet, as a complex assembly of diverse technologies, also features privacy risks for the users. This section discusses risks associated to the access network, to the Domain Name System (DNS) that is central to almost every activity on Internet, and finally to phishing protection services that could be turned into privacy intrusive tools.

WIRELESS ACCESS NETWORKS

In wireless access networks the traffic carried over the wireless link is in general protected by security mechanisms like WPA in IEEE 802.11 (Wi-Fi) networks. However, the headers and the content of management frames is not protected and is thus available to snoopers. The exposure of this information poses serious privacy threats that are made critical by the massive adoption of portable devices and the development of wireless networks.

12. For instance the InMobi company has been condemned in 2016 by FTC (https://www.ftc.gov/news-events/press-releases/2016/06/mobile-advertising-network-inmobi-settles-ftc-charges-it-tracked) because of their unfair misuse of the ACCESS WIFI STATE Android permission to track the users' geolocation.
13. https://www.cnil.fr/fr/enceintes-intelligentes-des-assistants-vocaux-connectes-votre-vie-privee

Picking up IoT waves – *© Inria / Photo C. Morel*

More precisely, Wi-Fi enabled devices scan for nearby access points by sending probe requests. These probe requests can include the name (SSID) of the network to which the device has been associated in the past. In that case, the SSIDs emitted by a device reveal a lot of personal data, like travel history and identity, and one can also infer social links between users. Another aspect with 802.11 frames is the MAC address, a globally unique identifier tied to the device. Using this identifier, it is possible to detect the presence of people and track them in the physical world.

The opportunity of tracking smartphones users has been seized by corporations, leading to the rapid growth of cyberphysical tracking systems. For instance, they are deployed in commercial centers to measure and analyze client movements and habits, and attempts have been made to deploy such technologies within public equipment in order to display targeted advertising. Measures taken by these corporations supposedly to anonymize data and reduce privacy risks have proven to have a limited impact.

Solutions to enable privacy preserving analytics for this kind of applications are thus needed. A promising approach is to use probabilistic data structures based on Bloom-Filters that include a perturbation mechanism to enforce strong privacy guarantees while allowing the accurate estimation of the number of detected device identifiers. In response to tracking issues, Wi-Fi vendors started

implementing MAC address randomization, a technique in which a random and temporary pseudonym replaces the real MAC address in the IEEE 802.11 frames. Despite the adoption of this Privacy Enhancing Technology, studies have shown that it may still be possible to track users through active attacks that force devices to reveal their real MAC address and through fingerprinting attacks based on content and probe request timings that could be used to single-out devices.

A lot remains to be done for Wi-Fi and similar wireless technologies in order to reduce the privacy risks associated with their use; this remains an active research topic.

CORE INTERNET SERVICES

Beyond the access networks, the DNS protocol, central to the Internet (see §2.2), does feature privacy threats that remain unsolved. The progressive transition to the "DNS Security Extensions" (DNSSEC) version solves some of the related security threats, however it does not address confidentiality requirements. For instance, even in the presence of DNSSEC, an eavesdropper present between a client and its DNS resolver will be able to analyze DNS traffic and identify most of the visited web sites, even in case of encrypted HTTPS web traffic [14]. This leaked information is considered as personal data (it is associated to a physical person) and tells a lot on the user centers of interest. Other risks are discussed in [Bor15].

Several directions are under consideration in the context of the "DNS PRIVate Exchange" (DPRIVE) IETF working group [15], leveraging on encrypted communications (e.g. over TLS or DTLS). Research is still needed as practical considerations make the real-world situation more complex than it may seem.

MALICIOUS ACTIVITIES DETECTION SYSTEMS

Browsing the web can lead users to visit malicious websites. The Safe Browsing techniques have been setup in order to detect blacklisted URLs (e.g**,** websites known to be involved in phishing attacks or to contain malware). Although very useful to the end-user, these services are also potentially dangerous from a privacy point of view. Storing the whole list of blacklisted URLs on the host is impossible, and relying on an external service to check visited URLs is too intrusive (the service provider would be in position to collect all the URLs visited by a user). Therefore, intermediate solutions are used in practice, and there is a need to carefully analyze them from a privacy view point.

14. 4DNS traffic between clients and their DNS servers traditionally goes through UDP communications, incompatible with TLS protection.

[Bor15] Stephane Bortzmeyer. DNS privacy considerations. *RFC*, 7626, 2015. https://tools.ietf.org/html/rfc7626

15. https://datatracker.ietf.org/wg/dprive/about/

The Google Safe Browsing service, in turn reused by most web browsers, has been studied from the privacy perspective. Although the company probably did their best to anonymize data and be privacy compliant, this Google service lacks transparency and accountability. More generally opening and discussing their technology with independent, trusted third parties would be highly beneficial.

[Research challenge 9] Towards a privacy preserving smart connected world

Our connected world experiences an unprecedented growth in terms of personal data collection, with practices that are increasingly intrusive for the citizen's intimacy. Surfing the web, using smartphones and other smart devices, driving a connected —and soon to be autonomous— car are activities that generate personal data leaks. The lack of transparency (many services and devices behave as black boxes) and lack of user control (how to express consent or opposition when there is no information, nor user interface) are major issues.

Identification of such hidden behavior is hindered by the number and complexity of underlying technologies specific to each domain. For instance, identification of tracking practices in a web page requires advanced JavaScript execution analyzes, while monitoring of smartphone applications needs dedicated frameworks, and monitoring of certain wireless communication technologies remains mostly unsolved. The analysis of these data flows is required to assess potential privacy leaks, e.g., in a smart home. Such challenging and diverse research activities are essential to bring transparency, highlight good and bad practices, and enable regulators to enforce data protection laws. As such, this research directly helps in the shaping of our future smart connected world.

[Inria teams] Privacy analysis of existing systems

↗ The **CIDRE** team works on user control in the context of social networks. The team has proposed a classification of existing social networks based on the type of implementation (centralized versus distributed) of their functionalities (e.g., communication, search, or storage).

The team also worked on location privacy, producing the GEPETO tool[a] whose goal is to enable a user to design, tune, experiment, and evaluate various sanitization algorithms and inference attacks and evaluate the resulting trade-offs between privacy and utility.

↗ the **COMETE** team works on location privacy, proposing the notion of geo-indistinguishability [ABCP13], that extends differential privacy to distance metrics: the precise location of a device is protected through the addition of controlled noise to the reported position. The team has also developed a tool based on geo-indistinguishability, called Location Guard[b], a browser extension that allows to protect the user's location while accessing location-aware websites, by adding controlled noise to it.

↗ the **DIVERSE** team works on web browser fingerprinting. The *Am I unique* website[c] demonstrates how the situation evolved with the most recent web technologies (e.g., HTML5 brought highly discriminating attributes) and shows that fingerprinting is today as effective on mobile devices as it is on computers, albeit for different reasons. In order to protect against fingerprinting, the team also works on browser diversification, with the FPRandom proof of concept[d].

↗ the **INDES** team works on analyzing various forms of web tracking in web applications. In a joint work with **PRIVATICS**, the team also demonstrated that web browsers can be fingerprinted through the extensions[e] that the user installs and the websites where they are logged. Finally, the team proposes a server-side technique to protect against web tracking: two additional servers respectively rewrite and redirect the original web application requests to prevent third-party tracking automatically[f].

↗ the **PESTO** team works on privacy in online social networks, with the goal to inform users of latent information that can be inferred from their published information.

↗ the **PRIVATICS** team works on privacy threats introduced by the information society, in particular for biometry, web tracking, smartphones, smart world (IoT), wireless access networks, or malicious website detection systems. The goals are to understand the situation, analyze the threats, and if applicable design privacy preserving solutions to prevent or mitigate them.

For instance, the team proposed a general framework for the specification and formal reasoning of biometric architectures, and applied it to several architectures for biometric access control. The team also contributed in [CM17] to the secure electronic document systems (TES) debate. Through the MyTrackingChoices project, the team proposed a new type of ad blocker that lets the user choose which website categories can track them or not. The team also worked on the subject of smartphone and smart world (IoT) privacy, following a multidisciplinary approach with lawyers and economists, and on the questions of transparency and user control inside a smart city. Finally, the team pioneered the problem of privacy leaks within wireless access networks and developed the Wifiscanner[g] and Wombat[h] Wi-Fi scanning/tracking tools.

↗ the **SPIRALS** team works on analyzing web browser fingerprints, in particular its inconsistency and evolution over time.

a. https://gforge.inria.fr/projects/gepeto/
b. https://github.com/chatziko/location-guard
c. https://amiunique.org/
d. https://github.com/plaperdr/fprandom
e. https://extensions.inrialpes.fr/
f. https://www-sop.inria.fr/members/Doliere.Some/essos/
g. https://github.com/cunchem/gtk-wifiscanner
h. https://github.com/Perdu/wombat

Critical infrastructures, systems, and applications: use-cases for security

In this chapter we look at cybersecurity under the angle of applications. We focus on a few selected domains where cybersecurity is likely to have a strong impact. We selected in particular security sensitive domains:

- infrastructures: the Cloud, Software-Defined Networks (SDN), as well as the blockchain;
- critical and cyber-physical systems: the Internet of Things (IoT) and industrial systems;
- application areas: medicine, robotics (including connected autonomous vehicles), and machine learning (ML).

This list is of course not exhaustive, but it illustrates well that many systems and application areas are security critical and that security considerations must necessarily be included in the design process. The categorization into critical infrastructures, critical systems, and critical applications areas is also not absolute — for instance, medical implants are cyber-physical systems — and critical applications may of course build on critical infrastructures or systems.

6.1 Critical infrastructures

Communication infrastructures are primarily designed to offer a service, often with usability and efficiency as their primary goals. However, security is also crucial as these infrastructures may be used for storing and manipulating sensitive data. Loss of availability (or simply efficiency) and integrity of data may also have strong economic impact. We will focus on three examples of such critical infrastructures: the Cloud, software-defined networks, and the blockchain.

6.1.1 Security and privacy in the Cloud

[Summary]
Security remains an obstacle to a wider adoption of Cloud services by organizations for their critical services. Privacy has also become an important issue for Cloud users in the IoT era. Hence, a major challenge for Cloud providers is to provide their users with both security services with associated guarantees in SLAs and privacy preserving computing and data storage services. Security and privacy issues are exacerbated in the context of distributed Clouds, AKA. edge and fog computing. Trust and other security assumptions change over time owing to changes in the technology and the discovery of new vulnerabilities. These evolutions have led necessarily to the appropriate evolution of the threat models in Cloud security considering not only attacks on the tenant operating system but also attacks on the hypervisor. Securing hypervisors and operating systems and proving security properties on them appear to be timely and important research topics.

Autonomous vehicles – © Inria / Photo G. Scagnelli

Security is one of the main concerns in the adoption of the Cloud model[1]. The hardware and low-level software infrastructure is owned and governed by the Cloud provider while customers outsourcing their information system have the control on the services and the OS running in their virtualized infrastructures. Server virtualization enables the execution of different OS's and/or applications from different tenants on the same server in a datacenter. This multi-tenancy situation entails specific security and privacy threats.

Cloud environments face multiple security threats originating from different privilege levels (application, network, and operating system levels) in the infrastructure. In an IaaS (Infrastructure as a Service) Cloud environment, the attack surface is expanded with the addition of the highly privileged hypervisor, as the building block of a Cloud infrastructure, as well as its web-exposed management API. In such a context, security concerns two different actors: tenants, and providers.

Tenants are concerned with the security of their outsourced assets, especially if they are exposed to the Internet. Attacks targeting traditional information systems could also target applications running inside virtual machines in an outsourced infrastructure.

The provider is also concerned about the security of the underlying infrastructure especially since it has no insight regarding the hosted applications and their workload. In a Cloud environment, security threats originating from corrupted tenants against other legitimate tenants and their resources, threats against the

1. http://www.infosecbuddy.com/download-cloud-security-report

provider's infrastructure as well as threats towards the provider's API should be considered. Cloud providers need to assess security risks taking into consideration the hosted virtualized infrastructures and vulnerabilities in virtualization technologies. They need to know which virtual machines can interact via network protocols.

In IaaS Clouds, the Cloud provider manages the Cloud's security monitoring infrastructure while Cloud tenants manage their outsourced information system. Security monitoring is essential in Clouds. A specific challenge is to ensure that the security monitoring infrastructure is automatically reconfigured when Cloud dynamic events (e.g., VM migration) happen. Tenants are incited to trust the provider's claim (e.g., infrastructure availability) thanks to the assurance given by Service Level Agreements (SLA's), making a trade-off between tenant's disclosed private information and the monitoring service offered.

Although multi-tenancy maximizes efficiency for the Cloud provider's resources, it also offers the possibility that a tenant's VM can be located in the same physical machine as a malicious VM. This in turn engenders a new threat: breaking the resource isolation provided by the hypervisor and the hardware and gaining access to unauthorized data or disturbing the operation of legitimate VMs. One of the most prominent attacks that illustrates this threat is the side channel attack where an adversary with a collocated VM gains access to information belonging to other VMs (e.g. passwords, cryptographic keys). For example, the attackers could use shared CPU caches as side channels in order to extract sensitive information from a collocated VM.

Cloud providers are usually assumed to be "honest but curious": given their privileged position with respect to their tenants either a malicious provider or a provider whose system becomes compromised could threaten their tenants' privacy. Virtual machine introspection mechanisms can be used by the Cloud provider to monitor the tenants' virtual infrastructure. Outsourcing data in the Cloud means delegating control over data to the provider. Cloud customers have little to no control on where or for how long data is stored and to which third parties it is forwarded. Moreover, the Cloud provider may provision resources located in different countries with different data regulations resulting in data protection depending in where data is stored and this is often transparent to Cloud customers.

[Inria teams] Security and privacy in the Cloud

↗ The **AVALON** team is interested in the modeling of application security properties and their automatic enforcement for applications executed in Clouds. They proposed a specification-driven approach where the security is expressed as properties in a mechanism agnostic language to facilitate the expression of the user's requirements for both the application and its security resulting in the Sam4C toolbox. They also developed mechanisms to provide automatic application deployment and automatic enforcement of its security, with proven properties. .

↗ The **CASCADE** team works on privacy for the Cloud, designing a new generation of secure multi-party computation protocols enabling computation on encrypted data.

↗ The **CIDRE** team investigates security assessment in Clouds considering virtualization technologies, hypervisors and SDN, in connectivity extraction in Cloud infrastructures and Cloud-specific vulnerabilities in attack graph generation.

↗ The **MYRIADS** team aims at integrating security monitoring terms in IaaS Clouds' SLA's and designing a self-adaptable Cloud security monitoring infrastructure. The team has designed one such framework that is able to alter the configuration of its components and adapt the amount of computational resources available to them depending on the type of dynamic event that occurs in a Cloud infrastructure.

↗ The **STACK** team has proposed a compositional approach to the declarative and correct composition of privacy-preserving applications in the Cloud. The proposed approach provides language support for the composition of three techniques: symmetric cipher, vertical data fragmentation, and client-side computations to make Cloud applications' privacy preserving. The team also studies isolation threats in containerized environments investigating side-channel attacks in the context of edge Clouds.

6.1.2 Security of Software-Defined Networks (SDN)

[Summary]

The softwarization of networks is the ongoing evolution in the networking field. The goal is to enhance flexibility and reduce costs. However, these evolutions centralize the control of a network, providing a single point of failure for the attacker to target. Moreover, the network softwarization enables a better coupling between the network and applications thanks to diverse API's. While previously restricted to a limited number of actors, these API's significantly ease the development of applications, but also create a new attack surface.

Major evolutions have recently occurred in networking technologies. A shift towards the softwarization of networks deeply changed the network architectures and operations. The most notable emerging paradigms are SDN (Software-Defined

Networking) and NFV (Network Function Virtualization). SDN advocates a logically centralized and powerful controller to replace distributed algorithms (the former panacea) leaving only forwarding purposes to network devices. NFV enables any kind of network function to be run as a virtual machine, supposing some central controller or management plane to orchestrate the function's deployment. It is thus well aligned with the cloudification of everything. Although core networks will continue to rely on high-end specific network hardware, the softwarization of networks naturally relies on standard servers in a datacenter-like architecture. The final objective is to enhance flexibility in network operations and to reduce the costs substantially. However, there are inherent security-oriented challenges related to this paradigm shift.

Centralizing the control of a network reduces the attacker's focus and increases vulnerability: the attacker may efficiently disrupt a network by compromising a central controller, while distributed algorithms are more robust against individual attacks. Targeted denial-of-service against a single controller may provoke a severe impact. New technologies, new frameworks, and new protocols come along with these new paradigms to integrate as many functionalities as possible, thus increasing the attack surface. Guaranteeing the security and detecting the misuse of these protocols is of paramount importance.

In addition, the softwarization of networks will enable a better coupling between the network and applications thanks to diverse API's. Development of applications will no longer be restrained to a very limited number of actors as is currently the case with hardware appliances produced by very few manufacturers. Strict quality control will need to meet the market's usability and flexibility demands. Hence, runtime application behavior can mistakenly or intendedly lead to open security breaches in network. Moreover, such breaches may be the consequences of a set of applications that are only problematic when used co-jointly. Verifying policies and detecting configuration anomalies in softwarized networks must take into account that configurations are dynamic, using only static analysis is thus not appropriate. Service Function Chaining (SFC) consists in chaining multiple virtualized network functions (VNF) to deliver innovative services, including for instance security services (firewall, intrusion detection, deep-packet inspection, proxy, etc.); several chains could even share some VNF's.

NFV enables a higher flexibility and should make network functions more elastic. However, network functions can be collocated on the same physical machines. As the core idea is to take network functions off-the-shelf, knowing or predicting exact behavior is challenging. Moreover, expected short development and deployment cycles can be the source of dangerous behaviors, mistakenly or intendedly introduced. Checking and evaluating the output of a network function from a security perspective is essential. Several approaches can be leveraged: static code analysis if possible, dynamic analysis, or testing. Furthermore, isolation

between collocated network functions is primordial. Whereas VM isolation has been largely studied in the past in order to render sustainable the Cloud model, the game changes when it comes to virtualizing network functions. Indeed, achieving line-rate operations leads to relaxing strong isolation properties and thus can lead to security problems. A good trade-off between security and performance needs to be found and might be adjusted depending on the criticality of the VNFs. Detecting and preventing a misbehaving VNF (Virtual Network Function) that impacts the operation of a collocated VNF is necessary. Similarly, the VNF deployment may integrate security constraints rather than just being seen as a pure resource allocation problem.

[Inria teams] Security of Software-Defined Networks

↗ The **COATI** team focuses on optimizing resources used by forwarding devices in SDN. The goal is to be able to store more forwarding rules in a compressed manner while maintaining acceptable performance and limited control overhead.

↗ The **DIANA** team contributes to the definition of new solutions for the SDN and NFV paradigms that take into account security considerations. In addition, the team evaluates the performance of related technologies and so, indirectly, their ability to resist to heavy loads created by denial-of-service attacks.

↗ The **RESIST** team works on the definition and testing of new network programming abstractions. Runtime and in-switch execution of VNF's need a new data plane abstraction. Since isolation between VNF's may not be guaranteed, it becomes necessary to introduce safeguards and methods to predict execution time and dependencies.

↗ The **RESIST** and **VERIDIS** teams both aim at checking SFC's by translating complex policies into a formal representation. Verification is needed to detect potential inconsistencies due to the distributed deployment of multiple SFCs.

6.1.3 Blockchain

[Summary]

By proposing a trusted, append-only and immutable ledger with various models for writing and managing it (fully decentralized, permissionless, permissioned, consortium, etc.), blockchains enable many applications relying on this new security feature and its related infrastructures. Yet, their real security and level of trust need to be properly asserted with analysis both from the cryptography and distributed systems communities. Also, as with any Internet system, blockchains can suffer for instance from low-level network attacks, software bugs, or failures, that can appear when blockchains are used for higher-level applications (smart contracts). Additional features such as strong privacy and anonymity may also conflict with security requirements from public and legal bodies.

A SECURITY PROMISE

The quite trendy and emerging blockchain technology, first put in place by the Bitcoin protocol, is a security promise. A blockchain, in any of its variants, implements a secure electronic ledger, a functionality similar to existing ledgers managed by banks, notaries, states, etc., providing the main security property that data registered in the ledger cannot be removed or modified. Indeed, the blockchain provides integrity of past history, relying on the notion of cryptographic hash functions: the hash of the last trusted block certifies the integrity of the whole ledger since its inception.

SECURITY OF THE LEDGER

Such a ledger can be guaranteed by a single, centralized entity, trusted for certifying the last block of data on a regular basis [HS91] and for regularly publishing chained integrity checksums of the ledger. Decentralizing the role of certifying the blockchain is the main innovation of Bitcoin. The decentralization could follow a peer-to-peer model, with unregistered, unknown, and untrusted nodes (i.e., permissionless as in Bitcoin or Ethereum). The decentralization could also follow a permissioned model, where either some participants are known to have the role of signing the last block while controlling each other (e.g., Hyperledger, industrial blockchains), or participants are unknown stakeholders that are believed to certify the ledger honestly to protect their stake.

As a consequence, the security analysis of a blockchain depends on the social and political model underlying the blockchain: the Bitcoin blockchain can be attacked by a powerful adversary controlling 51% of the computing power of the Bitcoin network (Namecoin has exhibited this weakness [ANSF16]); consortium blockchains can be completely and quickly rewritten if signing keys are compromised; and blockchain with stakes encounter the risk a non rational stakeholder acting against their own financial interests.

DYNAMICS OF THE PROTOCOL

Besides the integrity properties of the ledger, the problem of dynamically determining the current data block to be appended to the ledger brings into focus the domain of distributed algorithms, where issues of liveliness and correctness are central. Since the blockchain is replicated between participants, it is crucial that all have the same view of the blockchain: failing to do so allows an attacker to profit from the divergent views of the blockchain. While most blockchain protocols establish that the views are the same after a certain period of time, there is always a period of instability before the agreement is reached. Attacks

[HS91] Stuart Haber and W. Scott Stornetta. How to time-stamp a digital document. *Journal of Cryptology,* 3(2):99-111, January 1991.

[ANSF16] Muneeb Ali, Jude Nelson, Ryan Shea, and Michael J. Freedman. Blockstack: A global naming and storage system secured by blockchains. In *2016 USENIX Annual Technical Conference (USENIX ATC 16),* pages 181-194, Denver, CO, 2016. USENIX Association.

can thus arise (for instance double spending) and non-cryptographic issues from distributed algorithms need to be addressed.

NETWORK ATTACKS

Network issues are often not considered, but blockchains can suffer from low-level network attacks. Their security cannot be solely guaranteed by the cryptographic protocol itself. Recently, it has been proven that the mining power of bitcoin networks is very unbalanced and that a network-level attack is possible through the partitioning the overlay network (BGP hijacking) [AZV16]. No exhaustive assessment of network-level attacks and their impact on the blockchain security and performances has as of yet been realized. A threat assessment model needs to be established and adapted to each blockchain technology. Security can be improved by adding either security mechanisms by design (network-level requirements, such as topology constraints) or counter-measures at runtime (coupling between the network and the blockchain). First, a monitoring scheme is needed to assess the security of a blockchain. Second, dynamic mechanisms such as blacklisting nodes or strengthening the consensus model can be dynamically applied (programmable networks, with flexible configurations will be helpful). Blockchain-specific network policies could be defined, deployed, and verified automatically, for example assuming cooperation between autonomous systems exchanging BGP route information.

SOFTWARE FAILURES

A blockchain system has to be implemented in software, as must be applications using blockchains. These implementations can encounter the same problems as any system, such as bugs, attacks, undefined behavior, and so on. These issues are much more critical for blockchains, however, because their applications handle currency and securities. This means any issue will have direct financial impact, giving cyber-criminals strong economic incentive to find or create such issues. Furthermore, applications must cope with notoriously error-prone features such as concurrency, distribution, authority, and secrecy. While there have been no major issues with the Bitcoin or Ethereum systems per se, both clients (the MtGox exchange) and applications (the DAO) have suffered devastating breaches.

BLOCKCHAINS AS BUILDING BRICKS FOR HIGHER LEVEL PROTOCOLS

A blockchain can be used as a building block providing a ledger, on which higher level programs and protocols can be implemented. For instance, Ethereum envisions the blockchain as storage with integrity for data and programs, where programs can be activated for writing and updating data on the blockchain, while not erasing past versions. Similarly, Bitcoin has a weak, yet useful, notion

[AZV16] Maria Apostolaki, Aviv Zohar, and Laurent Vanbever. Hijacking bitcoin: Large-scale network attacks on cryptocurrencies. *CoRR*, abs/1605.07524, 2016.

of programmable money that allows for instance off-chain transactions and cross-chains atomic swaps. Even when considering the underlying blockchain as a perfect cryptographic ledger primitive, developing higher level programs and protocols has the same pitfalls as developing high level cryptographic protocols given simple basic blocks. Because privacy is lacking by default in permissionless blockchain, some efforts have been made to ensure privacy using zero-knowledge proofs and other advanced cryptography (Monero, Zcash). At this point, monitoring requirements or legal constraints, e.g., KYC (Know You Customer), are conflicting with the privacy provided by strong cryptography, but this conflict can sometimes be dealt with, depending on the blockchain model.

[Inria teams] Blockchain

↗ The **ANTIQUE** team is conducting research on the semantics, abstraction, formal proof, and automatic analysis of systems, together with verification of security properties, on distributed data-structures and consensus protocols.

↗ The **AVIZ** team develops visualization tools and exploration methods for the interactive analysis of complex and large data sets, in particular the Bitcoin blockchain.

↗ The **CASCADE** team is doing cryptography with rigorous security proofs and has an expertise in protocols for e-cash (centralized and decentralized "cryptocurrencies") and privacy tools for blockchains.

↗ The **CIDRE** team is designing distributed agreement algorithms to improve security, performance, and scalability of permissionless blockchains by focusing on both the underlying peer-to-peer networks and the structuration of the information in blockchains.

↗ The **COAST** team is doing research on safe and secure peer to peer data sharing and service composition for collaborative environments without a central authority.

↗ The **DELYS** team does research on distributed systems, from theory to algorithms to implementations, including blockchain as a core mechanism where they work to improve its performance and scalability by reordering operations without compromising safety and by interoperation between blockchains.

↗ The **GRACE** team studies privacy and secure multiparty protocols and how they can be used in the context of blockchains, as well as coding theory for distributed storage.

↗ The **PROSECCO** team works on the analysis, design and implementation of security protocols based on blockchain technology, e.g., smart contracts, through the use of formal methods, software languages and tools.

↗ The **RESIST** team is designing novel approaches to monitor and configure the blockchain infrastructure (network, Clouds, etc.) in order to enhance or guarantee performance and security. In addition, the team also investigates blockchain-based management to address the evolution of the Internet ecosystem.

↗ The **SPECFUN** team is doings formal proofs of mathematical theories, as well as formal analysis of Ethereum environment and other blockchain enabled execution models.

↗ The **TOCCATA** team promotes formal specification and computer-assisted proof in the

development of software that requires a high assurance of its safety and correctness. ↗ The **VERIDIS** team studies mechanized verification techniques applied to concurrent and distributed algorithms and systems. It is interested in precise formulations of the algorithmic problems that arise in the context of blockchains and corresponding verification problems.

6.2 Critical and cyber-physical systems

Critical systems are systems that require very high reliability, because failure could have extremely harmful consequences, such as endangering life, causing severe damage to an infrastructure, or cause important economic loss. However, most safety-critical systems are also security critical and must resist a cyberattack.

A related concept is that of Cyber Physical System (or CPS for short). Wikipedia defines a CPS as *a mechanism that is controlled or monitored by computer-based algorithms, tightly integrated with the Internet and its users. In cyber-physical systems, physical and software components are deeply intertwined, each operating on different spatial and temporal scales, exhibiting multiple and distinct behavioral modalities, and interacting with each other in a myriad of ways that change with context.* CPS's are often part of critical systems, which must be secured. Composed of many interconnected subsystems, running different

Security of embedded systems (fuzzin) – © Inria / photo C. Morel

protocols for communication at different scales, they offer a large attack surface. Besides, some of them are exposed targets because a successful attack, such as shutting down part of a state infrastructure, may have a huge economic and political impact. Hence, cybersecurity is now often an essential issue for CPS.

While cybersecurity of CPS may reuse traditional approaches, methods, and techniques for securing systems and networks, new approaches will also have to be developed to cope with dynamicity and to analyze and ensure their safety and security in this context. Interestingly, these might be inspired by reactive security, where monitoring and reaction to abnormal situations will take a more important place – and machine learning is likely to take an increasing important role that will, of course, introduce its own security issues.

[Note] *CPS vs. embedded* **systems**

CPS generalizes the much older concept of *embedded systems* in many ways. An embedded system also implies an interface between the digital and the physical world and emphasizes the temporal issues, but it focuses on individual components rather than the whole system and the interactions between its many components—and, often, humans as well. For example, the ABS (Anti-lock Braking System) is an embedded system—and thus also a CPS, while a power plant controller is a CPS, involving many other subsystems. A large majority of computer devices in use today are actually parts of embedded systems or CPS.

6.2.1 Security of Internet of Things (IoT)

[Summary]

The IoT revolution is extending the Internet at a rapid speed and is changing the way the world interacts with physical devices. Security and privacy considerations put this revolution at stake. IoT applications are not just yet another instantiation of previously known distributed systems: their specificities, in particular the resources-constrained nature of devices, present new challenges for security that need to be addressed and classical security solutions are not always applicable. Several research directions need to be explored, in particular: secure operating systems and firmware update capabilities, lightweight and hardware-assisted cryptography, wireless technologies security and privacy, IoT dedicated security and privacy policies, IoT intrusion detection methods, secure programming languages and compilers for IoT applications, and dedicated authentication protocols.

The Internet of Things (IoT) is the network of embedded physical devices, equipped with sensors, actuators, processing, storage, and connectivity that allows these things to connect and exchange data (Wikipedia). IoT applications belong to a wide variety of domains, from patient health (e.g., smart pacemakers),

quantified-self devices (e.g., activity tracking watches), to home appliances (e.g., smart thermostats, plugs, or bulbs), cars (e.g., vehicle-to-vehicle emergency systems), and industrial systems (e.g., SCADA). Because IoT enables a more direct integration of the physical world into computer-based systems, the IoT has been labeled [2] the Fourth Industrial Revolution.

However, concerns about IoT security, or the lack thereof, and privacy, since personal and sometimes sensitive personal data is exchanged, put this revolution at stake. For instance, the Mirai botnet (§1.3) involved a large number of consumer devices (including vulnerable IP cameras) to launch massive Distributed DoS attacks. Similarly, users' privacy is at risk when companies treat security as an afterthought (as was the case with some Internet connected sex toys [3]).

A first IoT specificity is the resources-constrained nature of devices. Microcontrollers in IoT devices are architecturally very different from a typical PC in terms of storage and CPU capabilities [4]. Certain IoT devices or components will be stringently constrained not only by low energy consumption but also limited resources. These ultra-low-power and ultra-low-cost devices operate on batteries with limited capacity or even harvest energy from the surrounding environment (e.g., light, heat, or vibration).

These specificities led to the design of dedicated operating systems to handle different embedded architectures. Popular IoT operating systems include Contiki [5] (initiated at the Swedish Institute of Computer Science), Mbed OS [6] (ARM), RIOT [7] (initiated by Inria, Freie University of Berlin, and Hamburg University of Applied Sciences), and Zephyr [8] (Wind River Systems, Linux Foundation, Intel, NXP Semiconductors, et al.).

Another specificity is the heterogeneous nature a global IoT application. Covering the spectrum from microcontrollers to the Cloud, an IoT application potentially includes code that runs in web clients, on servers, as well as in embedded devices. IoT applications must handle a wide variety of asynchronous events: queries to distant services, answers, timeouts, and errors. In turn, each event may launch calculations that trigger a cascade of new events.

Bringing security and privacy to IoT is challenging, the attack surface being important and classical security solutions not always applicable. Among the research directions to be addressed, we can mention the following:

2. Source: the World Economic Forum
3. https://blog.trendmicro.com/penetration-testing-researchers-successfully-hack-a-vibrator/
4. For example, RAM and Flash storage are 106 times smaller in an Arduino Uno microcontroller and its processing power is around 16 MIPS (Million Instructions per Second) to be compared with a few 100,000s MIPS for recent desktop processors.
5. http://contiki-os.org/
6. https://www.mbed.com/en/platform/mbed-os/
7. https://www.riot-os.org/
8. https://www.zephyrproject.org/

SECURE SOFTWARE UPDATE CAPABILITIES

The secure deployment of software updates on IoT devices, and in particular the firmware itself, is a mandatory feature of any secure operating system. Unfortunately, this key feature is often compromised by the desire to keep device complexity and energy consumption as low as possible, or simply by bad practices. In addition to academic work, the Software Updates for Internet of Things (SUIT)[9] IETF working group investigates this question, focusing on an architecture that is agnostic to the communication technologies and protocols (e.g., CoAP or HTTP).

LIGHTWEIGHT CRYPTOGRAPHY AND HARDWARE-ASSISTED CRYPTOGRAPHY

Regarding the cryptographic building blocks, embedded device constraints can require the use of lightweight cryptographic primitives. This topic is discussed in §3.1.3.

IoT devices will also benefit from the design of specific hardware architectures and software optimizations of lightweight cryptography, including protection against side-channel attacks and fault injection. These objectives require an efficient platform based on:

- hardware accelerators (crypto-processors) for security functions (e.g., symmetric and asymmetric cryptography, hashing, authentication, signature, or random number generators), with a specific focus on energy efficiency and ultra-low-power;
- specialized crypto-processor designs, such as randomization, that protect against attacks;
- compiler optimizations targeting resource-constrained crypto-processors;
- hardware accelerated dynamic binary translation (DBT) as a mean to enhance software protection;
- and new techniques for efficient hardware protections against side-channel attacks and fault injection, both in software and hardware.

The global aim is to propose an energy-efficient crypto-processor with an instruction-set architecture that can be configured at runtime.

WIRELESS TECHNOLOGIES SECURITY

IoT communications leverage both on standard wireless technologies (e.g., Wi-Fi and Bluetooth) and on low power technologies (e.g., Bluetooth Low Energy (BLE) and Zigbee), as well as low-power wide area networks in unlicensed bands (e.g., LoRa[10] and SigFox[11]) and in licensed bands (the promising 5G that also aims at connecting IoT devices). Work is in progress with respect to the security and privacy risks (§5.3.6) associated to these communication technologies.

9. https://datatracker.ietf.org/wg/suit
10. https://en.wikipedia.org/wiki/LoRa
11. https://en.wikipedia.org/wiki/Sigfox

IOT DEDICATED SECURITY AND PRIVACY POLICIES

Another challenge is the design of IoT dedicated security policies that prevent attacks exploiting the exposure of the system to physical hazards or resulting in physical consequences.

IOT INTRUSION DETECTION METHODS

Another challenge for IoT security is the design of IoT intrusion detection techniques to stop malware and other attacks, for instance based on the device activities in IoT networks (§4.4).

SECURE PROGRAMMING LANGUAGES AND COMPILERS FOR IOT APPLICATIONS

Programming an IoT application is complex because of the mix of technologies involved and the languages used, and taking security and privacy into account adds another level of complexity. Several frameworks meant to simplify IoT development exist, however they are not always designed with security in mind and can present serious security risks. Therefore, there is a need to develop a secure and privacy-preserving programming language and framework for the IoT. The formal verification of IoT software, starting with its cryptographic library, is another possible research direction.

DEDICATED AUTHENTICATION PROTOCOLS

Finally, the combination of resource-constrained devices and IoT heterogeneity requires novel authentication protocols to provide a simple, secure, and privacy-friendly way of seamlessly authenticating individuals and their smart devices online.

[Inria teams] Security of Internet of Things

↗ The **ANTIQUE** team is working towards analyses based on abstract interpretation for IoT programs.

↗ The **CAIRN** team tackles the question of improving energy efficiency through the use of flexible hardware accelerators. It proposes to rely on morphable hardware, whose structure can be efficiently reconfigured at runtime, for example to specialize the hardware to the application it executes, adapt the accelerator parallelism for performance or resource usage. The team also explores hardware accelerated dynamic binary translation.

↗ The **CELTIQUE** team is working on the formalization of the Hop.js semantics in Coq as well as on formal verification of program analyses for the IoT.

↗ The **FUN** team is working on the verification of the Contiki system as well as on intrusion detection techniques for the IoT.

↗ The **GRACE** team participated in the design of several targeted cryptographic primitives for microcontrollers, notably qDSA, a signature scheme for IoT, now included in the RIOT operating system.
↗ The **INDES** team is working on the development of a secure compiler and on the enforcement of privacy policies for Hop.js, a programming language for the IoT.
↗ The **INFINE** team leads the RIOT[a] open-source IoT operating system development along with the Freie University of Berlin and Hamburg Univ. of Applied Sciences. Work is under progress on supporting secure firmware update facilities for RIOT.
↗ The **KAIROS** team is working on formal model-based co-design techniques for real-time and security aspects of the IoT.
↗ The **PETRUS** team is working on adapted strategies to stock data in IoT objects instead of sending the data to the servers.
↗ The **PRIVATICS** team is working on privacy considerations and privacy preserving systems for the IoT.
↗ The **PROSECCO** team is building HACL*, a verified cryptographic library that is already included in RIOT and work is underway on a lightweight version for resource-constrained IoT devices.

a. https://www.riot-os.org/

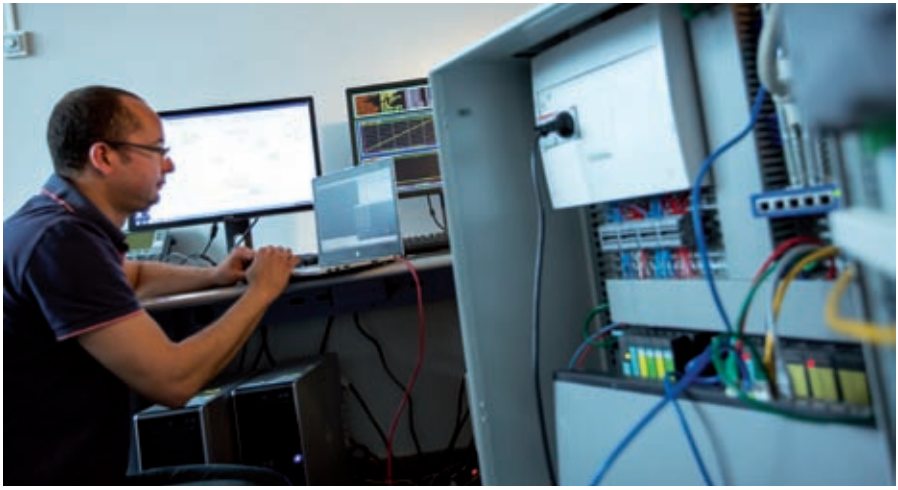
[Research challenge 10] Securing the Internet of Things (IoT)

Security in IoT is a major challenge: attacks are still relatively easy (many devices have not been designed with security in mind), invasive (e.g., pervasive in our lives), and potentially with a major impact due to the multiplication factor made possible by the large number of devices available and to the direct implications some of them have in the physical world (e.g., connected cars). The research directions are manifold, with for instance the ability to securely update embedded devices' software, the design of lightweight cryptographic primitives adapted to limited resources, the analysis of the security of new low-power wide area wireless technologies, the detection and mitigation of intrusions or misbehaving devices, and the need for secure-by-design frameworks, protocols, and operating systems.

6.2.2 Security of industrial systems

[Summary]
Cybersecurity of Industrial Systems is an emerging topic and recent cyberattacks on industrial systems show that the problem is open. One of the main difficulties in dealing with industrial systems is that they fail the security-by-design principle: since they were not intended to be exposed on Internet at the origin, the protocols used are not secure; sometimes specifications are not publicly available; general-usage firewalls and intrusion detection devices are not handling industrial protocols. End devices are built with slow processors unable to use standard cryptographic protocols and thus require dedicated ones.

Industrial Control Systems (ICS) or Industrial Automation and Control Systems (IACS) include a large variety of industrial digital data acquisition, control, and monitoring systems together with their underling communication networks. Two system architectures are currently used: DCS (Distributed Control System) and SCADA (Supervisory Control and Data Acquisition), although SCADA tends to supersede DCS.



Understanding attacks on industrial control systems – © Inria / Photo C. Morel

Cybersecurity of Industrial Control Systems is a relatively recent application field. Communications in traditional industrial systems were achieved through small size networks using proprietary protocols. They were submitted to hard real-time constraints and not interconnected with IT systems or the Internet. Therefore, traditional cybersecurity in industrial systems was achieved by

isolation and obscurity, using proprietary protocols. Modern and large-scale industrial systems such as power grids, nuclear plants, or hydraulic dams have global control optimization needs that require interconnexion with supervisory control applications, distributed data-base systems, and, consequently, long-range communication, standard data-exchange formats and interoperability. The paramount communication paradigm in the modern industrial systems is the convergence of the Information Technology (IT) and Operation Technology (OT), or said otherwise, the interfacing between TCP/IP networks and real-time proprietary protocols. The counterpart of IT pervasion in the OT area is that the OT is now exposed to cyberattacks in the same way as the IT. Moreover, as the OT still heavily relies on legacy protocols they are even more vulnerable. However, for almost a decade (1995-2003), the cybersecurity of the industrial systems was largely ignored.

The initiating event of the industrial systems cybersecurity research was the Northeast blackout of America in 2003. Although not the result of a cyberattack (for memory, the blackout was due to a wrongly calibrated sensor and a chain of IT bugs), this event demonstrated that a false data injection may exploit several vulnerabilities to shutdown 256 nuclear plants and blackout 55 million people in the US and Canada. Little progress was made until 2010 while the reality of the threat was not formally proved. After the occurrence of Stuxnet in 2010-2011 important research programs were started. Further events like the string of attacks on the Ukrainian power grid in 2015-2016 increased the importance of industrial cybersecurity programs.

The IT cybersecurity solutions do not apply directly, due to the specificities of the industrial systems. Below is a list of soft spots of the ICS communication.

- *Insecure and unsecurable protocols:* legacy protocols were not designed for security; moreover, they are also intended to be used with very low speed processors, which rules out the use of cryptographic protocols.
- *Unavailable specifications:* some legacy protocols are still proprietary and the full specifications are not disclosed.
- *Many versions:* some legacy protocols are intended to be extended by constructors with their own messages.
- *Large attack surface:* the end devices (e.g., programmable logic controllers) often act as network gateways between several legacy networks.
- *Process-oriented attacks:* the legality of a network packet depends on the state of the underlying physical system or the frequency of the packet. For example, opening a feeding tap of a tank will be harmless if the tank is at an intermediate level but may damage the plant if the tank is full. Similarly, Stuxnet only modified the value of an otherwise legal control command. Indeed, repeatedly starting and stopping an actuator may eventually damage it.

Process-oriented intrusion detection is an important topic in the ICS cybersecurity area.

[Inria teams] Security of industrial systems

↗ The **CIDRE** and **CTRL-A** teams developed an approach based on monitoring the process specifications. Process specification are safety properties expressed in Linear Temporal Logic (LTL). They are automatically mined from execution traces, then monitored by runtime verification techniques. Alert correlation between process monitors, a white list network Intrusion Detection System (IDS) and a pattern IDS are used to reduce the number of the false positives. The approach is validated on an ICS testbed.

↗ The **CTRL-A** team has also developed a testbed and demonstrators of attack/defense in industrial systems and smart-grids. On the protocol vulnerability research side, the **CTRL-A** team worked on a vulnerability of the real-time protocol in IEC 61850 networks, demonstrating the attack, and proposed an IDS module in BRO.

[Research challenge 11] Secure Industrial Systems

Industrial systems rely more and more on software mechanisms that can be attacked. Their security has thus become a major issue, especially since the consequences of an attack against such systems can be dramatic. Although traditional security approaches seem applicable to the case of industrial systems, their specificities require reviewing the traditional security mechanisms to adapt them to this new context. In particular, the communication protocols used in this context cannot be modified overnight. There must be a transition during which legacy communications should be embedded in secure protocols. In addition, real-time control of the system is usually required. Security must thus also be applicable in real time. Finally, it is often impossible to modify industrial devices. Therefore, preventive security mechanisms cannot be used. It is then mandatory to use reactive security and thus extremely important to study how effective attack detection mechanisms could be deployed in this context.

6.3 Critical application areas

Some application areas are particularly security sensitive. Medicine is one example, as medical data is extremely sensitive and attacks on medical apparatus and implants could endanger life.

Finally, machine learning has recently become extremely popular in many different areas. Even though it is not an application area in itself, it is an interesting case study for security: machine learning techniques may be misled by so-called adversarial examples and present privacy risks when trained on sensitive data.

6.3.1 Medicine

[Summary]
Medicine is being significantly transformed by the digital revolution, and is therefore also increasingly exposed to cybersecurity threats. One threat is the leakage of sensitive, medical data. As this data is extremely useful for research, the compromise between privacy and utility is of particular importance. Medical apparatus, such as surgical assistant robots, are increasingly connected. Similarly, medical implants offer wireless connections to avoid unnecessary surgeries, increasing the attack surface in a similar way as for CPS and IoT.

Medicine, as many other domains, is being significantly transformed by the digital revolution and is therefore also increasingly exposed to cybersecurity threats. While cybersecurity in medicine is not much different from cybersecurity in other domains, many aspects of cybersecurity are actually more sensitive in medicine, because medicine ultimately involves human lives, for which there is no price and thus no acceptable risk. This of course applies when an attack could endanger our lives, but also when it could leak our medical records that belong to the class of Sensitive Personal Information and thus are subject to very protective regulations (see chapter 5).

Therefore, looking at medicine as an application domain is instructive. Although privacy has long been a serious concern in medicine, it is still a challenging issue; other aspects of cybersecurity that could endanger our lives due to malicious actions are often underestimated.

In fact, while safety has always been part of the medicine culture, where all possible precautions are taken to save our lives, security is usually not much of a concern, because hospital and care centers are normally considered a sanctuary where there is no place for crime. This may be changing with the digitization of medicine, as attacks may be conducted remotely, from outside of the hospital.

There are mainly three kinds of threats: privacy leaks, malicious actions on medical apparatus, and attacks on implants.

PRIVACY LEAKS

Our concern —and thus our awareness— for privacy leaks is much higher than for malicious actions, probably because even before digitization, we have always treated health information as very private and personal. However, medical data is being more and more centralized and stored for very long periods, hence, despite our awareness, this remains a critical issue that has not yet found an acceptable compromise.

Although medical data is highly sensitive and should not be leaked, the increasing amount of medical data over a person's lifetime and on a very large

population constitutes precious information for medical research. Indeed, a medical study based on a few tens or hundreds of patients is far from statistically sufficient. We will soon have the possibility of analyzing not only thousands or even millions of records, but also records containing extremely rich information for which machine learning techniques could reveal interesting correlations between pathologies and physio-biological factors. Even though keeping medical data private and decentralized would be a good answer to privacy preservation, it could be a hindrance to medical progress, given the considerable potential impact access to this data could have for medical discoveries.

MEDICAL APPARATUS

Health care is an important user of digital equipment. If it has long been the case with imaging equipment, it has recently spread to all kinds of monitoring, as well as robotic surgical assistants. All these systems are of course interconnected, which increases the attack surface. There are no particularities to mention about the apparatus except that in general cybersecurity threats and sometimes even the safety of these devices are not given sufficient attention.

IMPLANTS

More worrying are medical implants (running the gamut from pacemakers, insulin implants, or audio implants to artificial hearts) that nowadays can be controlled remotely, allowing for the adjustment of parameters without a surgical intervention. This automatically exposes them to cybersecurity attacks as for Cyber Physical Systems (see §6.2). However, these implants are usually under important size and energy-consumption constraints, often making normal encryption techniques inapplicable, as is often the case in IoT devices. Manufacturers then choose ad hoc solutions that are likely to be insufficiently secure.

[Inria teams] Medicine

↗ The work of the **PETRUS** team on private Cloud (§4.2.2) was originally motivated as a way to decentralize medical data, allowing everyone to hold their personal medical records and keep control on what, how, and by whom pieces of information can be accessed when necessary.

↗ The **PRIVATICS** and **TYREX** teams are jointly working on better compromises between privacy and utility (§5.1.1), anonymization techniques (§5.2.2), and the privacy of machine learning (§6.3.3).

6.3.2 Robotics and connected autonomous vehicles

[Summary]
We can partition robotics into four major application areas: self-operating robots (e.g., factory robots, robots to assist the elderly), remotely operated robots (e.g., surgical robots, drones), robotic-based transportation (e.g., terrestrial vehicles with or without people inside), and large systems of robotic systems (e.g., smart cities). At Inria most research around cybersecurity and robotics focuses on connected autonomous vehicles (CAV's). Connected cars are communicating with each other and also with external objects. Due to legacy issues, architectures do not isolate critical parts from non-critical ones. This makes CAV's vulnerable to both internal and external threats. Internal threats are generally due to software bugs or unsecured channels. External ones are due to remote attacks exploiting communication channels. Current directions for avoiding these attacks tend to isolate the critical subsystems from non-critical ones.

Systems of connected robots are instantiations of the cyberphysical systems. Early examples can be found in factories and hazardous sites (e.g., nuclear or chemical plants). The closeness of humans and robots is getting tighter every day. Schematically, one may distinguish four application areas, which intersect partially: self-operating robots (e.g., factory robots, robots to assist the elderly), remotely operated robots (e.g., surgical robots, drones), robotics-based transportation (e.g., terrestrial vehicles with or without people inside), and large systems of robotic systems (e.g., Smart cities). Of course, there is a non-empty intersection between systems of connected robots and IoT. Cyberthreats may lead to undesired, possibly catastrophic and irreversible, outcomes in the cyberspace and in the physical space.

A recent study[12] from TrendMicro and Polytechnic University of Milan showed that industrial robots connected to the Internet are vulnerable and not secured. This security problem is also known for tele-operated surgical robots[13]. Roboticists are paying increased attention to cybersecurity issues, due to emerging low-cost hardware and SoC's that are changing the economic side of the equation. The possibility of having numerous affordable "smart nodes" (having low energy and computationally efficient cybersecurity services implemented on them) within networks of robots or IoT opens up new perspectives.

Currently, within Inria, the robotics-related domain where cybersecurity issues are addressed is connected autonomous vehicles (CAV's), in particular cars—see the

12. https://www.trendmicro.com/vinfo/us/security/news/internet-of-things/rogue-robots-testing-industrial-robot-security
13. http://www.washington.edu/news/2015/05/07/uw-researchers-hack-a-teleoperated-surgical-robot-to-reveal-security

dedicated white book on this subject[14]. In addition to onboard robotics, CAV's will be equipped with radio devices enabling the exchange of data with other vehicles (Vehicle-to-Vehicle, or V2V, communications) as well as mobile access to Internet-based services (Vehicle-to-everything, or V2X, communications). Furthermore, CAV's will be equipped with internal sensors able to monitor passengers, in particular their drowsiness. However, in current vehicles, all devices share the same communication bus and most actuators (e.g., steering, brakes) are accessible via the onboard diagnosis port. Moreover, current onboard systems are monolithic. Thus, existing CAV's are vulnerable to both internal and external cyberthreats.

INTERNAL CYBERSECURITY ISSUES

Internal issues are due to native attackers (faulty software, back doors) and to intrusions via unsecured channels (viruses, malware). Intrusions may lead an otherwise honest vehicle to behave maliciously. This has been illustrated with the takeover of a Jeep Cherokee using a Trojan horse placed in an MP3 CD.

Cameras and diverse sensors installed within CAV's raise specific privacy concerns. Who owns the collected data? What happens if the data is stolen? Also the underlying rationale where a passenger should be able to retake control whenever necessary is not widely acknowledged, nor accepted, since human reaction delays are usually considered to be too high.

EXTERNAL CYBERSECURITY ISSUES

External attacks can be performed remotely via V2X communications. Cybersecurity issues raised with V2X technologies are similar to those arising in any network services. And since V2X communications depend on intermediate relays (e.g., road-side units and telecommunication network nodes), without specific protection, they favor man-in-the-middle attacks and hence, message falsification, suppression, or spoofing.

In France, state authorities and insurance companies require authentication and non-repudiation for proper identification of responsibilities (e.g., in the event of an accident). Thus a state-issued certificate is stored in each vehicle in a tamper-proof device. For improving VAC safety, periodic beaconing is being considered. This implies the broadcasting of unencrypted position, speed, and direction several times per second. The real identities would be obfuscated using pseudonymity, i.e., reversible anonymity based on asymmetric cryptography[15]. Unfortunately, it has been shown that frequent changes of pseudonyms do not prohibit linkability of CAV paths. This is a real privacy concern and beaconing has come under question.

14. Véhicules autonomes et connectés — les défis actuels et les voies de recherche. Available at https://www.inria.fr/institut/strategie/vehicules-autonomes-et-connectes
15. https://research.utwente.nl/en/publications/pseudonym-schemes-in-vehicular-networks-a-survey

To avoid the above attacks scientists and manufacturers are defining partitioned architectures comprised of two subsystems. A Safety Critical subsystem is in charge of critical V2V messaging, processing, and cooperative driving. A non-safety critical subsystem, isolated from the critical part, is in charge of services based on V2X communications (e.g., infotainment, traffic and road conditions, free parking places, personal messaging). The safety critical subsystem will be endowed with the capability of inspecting V2X messages received by the non-safety critical subsystem, prior to importing them for further processing.

[Inria teams] Connected autonomous vehicles

↗ The **CIDRE** team currently works on the definition of an intrusion detection system (IDS) that could be integrated on the next generation vehicle architecture. Several paradigms (network vs host-based IDS, anomaly vs detection) should be explored in the CAV context.

↗ The **RITS** team is conducting research on safety critical subsystems and protocols for vehicular communications. The goal is to enable the prevention of attack, their immediate detection, and that render eavesdropping and tracking unfeasible and useless. In addition to pseudonymity, anonymity (non-reversible obfuscation of identities) of message senders has been shown to be feasible by forming trusted ad hoc vehicular sub-networks spontaneously. Onboard systems come with a "stealth mode" option (no sending of V2X messages), for increased privacy.

6.3.3 Machine learning based technologies

[Summary]

Machine learning is used in an increasing number of applications, including e-commerce and recommendation systems, advanced language translation mechanisms, spam or parental filtering tools, as well as self-driving cars and cyber physical systems in general. The deepest impact so far is certainly in speech and image recognition, but other areas are also significantly impacted. Machine learning techniques suffer from two main threats in relation to cybersecurity. The first one, called adversarial machine learning, consists in adding carefully designed noise (barely visible to human eye) to an image, leading to misclassification. The second one is related to privacy and consists in extracting information about training data from a trained network.

Machine learning (ML) based technologies now form the backbone of a quickly growing number of organizations and services. These include e-commerce and recommendation systems, advanced language translation mechanisms, spam or parental filtering tools, as well as self-driving cars. Millions of users interact on a daily basis with such systems, transparently, without even noticing.

There is one domain where convolutional neural networks and deep learning strategies have had a profound impact: computer vision. Traditional problems such as precise image classification or fine grain object detection and labeling in images have recently made enormous progress, with state-of-the-art approaches outperforming by far older methods and even surpassing human capabilities for specific domains.

ADVERSARIAL MACHINE LEARNING

Recent research has shown that such machine learning approaches can be subverted when a small amount of carefully-designed, imperceptible adversarial noise is added to the input data. Such a perturbed image, called an adversarial example, is in turn typically misclassified, although the perturbation is barely visible to human eyes. Adversarial perturbation generalizes surprisingly well across different input images, classifiers, and models, even when trained on diverse learning sets; worse, this phenomenon is not particular to deep learning and can be observed even with simpler classifiers. In addition, given some input data it seems to be relatively easy to compute a small perturbation of that input that will be misclassified. These alarming observations have many practical implications in an area where machine learning technologies are ubiquitous.

Ongoing research at Inria focuses in part on convolutional neural networks and follows several directions. It is necessary to gain a better understanding of the weaknesses of deep learning strategies, to analyze what kinds of attacks are possible and propose mechanisms for protecting convolutional neural networks against adversarial attacks. It is important to gain insight on why adversarial perturbations generalize so surprisingly well. Another approach is to monitor the internal activations that flow between the layers of a deep network, in order to observe where spurious artefacts take place, whether or not this is related to the dimensionality of the intermediate vector representations. It is certainly worth observing the effects of defensive strategies such as distillation on these flows to see why it makes the overall process more robust. Furthermore, it seems essential to connect adversarial perturbations to the subspace that corresponds to natural images, the subspace where classifiers are trained. This may facilitate the identification of unnatural images that in fact belong to different subspaces. Yet, adversarial natural images do exist and succeed in subverting the detection systems. Considering subspaces is therefore not the ultimate solutions for making systems more robust.

Finally, while most research currently focuses on images, other modalities such as adversarial text or adversarial audio are likely to raise a whole new set of difficulties which also have to be addressed. Domains that are outside multimedia are likely to face similar sensitivities to adversarial behaviors as they also heavily rely on machine learning, e.g., applications for spotting intruders from the network analysis or applications dealing with biometric traits.

PRIVACY ISSUES AND MACHINE LEARNING

Using machine learning techniques also raises privacy problems. In addition to the use of machine learning techniques to infer possibly sensitive data, machine learning also raises the question as to whether an attacker who has access to the trained network can gain information about the training data. Different scenarios can be distinguished according to whether an attacker is granted white-box (giving access to the neural network internals) or black-box access to the network, and whether their aim is to extract training data or simply decide whether a given input was part of the training data. These privacy properties have been stated in terms of differential privacy, introduced in the area of database anonymization (discussed in §5.2.2). In the worst case, the attacker may gain access to the stored training data itself. This raises the additional questions on how to transform the data prior to storage, so as to discard any private information that is useless for the task at hand, and how to train in a distributed online fashion, in order to avoid storing all data in a single place which increases the risk of a security breach.

[Inria teams] Cybersecurity and machine learning

While adversarial machine learning is often considered as a cybersecurity issue, as it can be used to attack critical systems that use machine learning techniques, it is actually quite different from traditional cybersecurity attack techniques, and therefore, mostly tackled by teams working in machine learning. Conversely, the privacy issues raised by machine learning are typically cybersecurity ones.

↗ The **COMETE** team is interested in the privacy aspects of machine learning, in particular through the use of differential privacy.

↗ The **LACODAM** team, working in machine learning, is also considering applications to cybersecurity.

↗ The **LINKMEDIA** team investigates adversarial machine learning issues that go beyond adversarial computer vision and considers for instance adversarial audio, adversarial video, adversarial Automatic Speech Recognition (ASR), or adversarial Natural Language Processing (NLP). Adversarial machine learning is also considered from the perspective of each modality, as well as considering truly multimodal inputs. The team also considers a wide range of applicative goals that go beyond classification and include adversarial retrieval.

↗ The **MAGNET** team works on privacy considerations in machine learning, in the context of decentralized learning where several actors collaborate to improve the model without leaking personal data, or in datamining of mobility traces.

↗ The **MULTISPEECH** and **MAGNET** teams work on privacy-friendly speech recognition. The goal is to train a speech recognition system on users' speech data without disclosing information about the identity, the traits (e.g., gender, age, or ethnic origin), or the states (e.g., health or emotional state) of individual users. To do so, the teams

investigate adversarial learning (which is seen here as a solution rather than a problem), decentralized training, and formal privacy frameworks such as differential privacy.
↗ The **ORPAILLEUR** team is interested in the privacy aspects of machine learning.
↗ The **PRIVATICS** and **TYREX** teams are jointly working on privacy preserving decentralized or federated machine learning, in particular in the context of large medical databases.
↗ The **SEQUEL** team, working in machine learning, is also considering applications to cybersecurity.

7

Cybersecurity in France

[Summary]

France is one of the major European actors in cybersecurity. Its academic workforce consists in almost one thousand people, including researchers, faculty, postdoctoral fellows, Ph.D students, and research engineers.

Almost 25% of the French academic activity in cybersecurity is conducted at Inria within joint teams with other academic institutions, primarily CNRS and universities, where on average only half of the staff is from Inria. This gives Inria a leverage effect and makes it one of the main French academic actors in cybersecurity. Inria's forces in cybersecurity represent about 7% of its total activity.

A large and visible effort, both at Inria and in France, is dedicated to cryptography and formal methods applied to cryptographic protocols and privacy. Hardware security is also well represented in France, but with very few forces are Inria. Work on network security, system security, and, more generally, on reactive security are, however, under-represented both at Inria and in France given the increasing challenges of cybersecurity for critical infrastructures and the arrival of the Internet of Things.

Education in cybersecurity is becoming a key challenge, at all levels. Inria contributes to education via the advising of Ph.D students, many of whom will later be working in state bodies or private companies. Transfer via the creation of startups does exists, but is not as of yet an important vector.

French academic forces in cybersecurity are rather well organized and coordinated, in particular via the dedicated pre-GDR and Allistene working group. However, there is still insufficient interaction and coordination between the industry and academic forces.

Although cybersecurity is usually well recognized as a priority in France, in Europe, and worldwide, it is important that France also reinforces its financial support to maintain its leading position and its capacity to seize economical opportunities.

7.1 Academic forces at Inria and in France

Research organization and forces

Research at Inria is organized into small teams sharing a common research project, which are often joint teams with other academic institutions such as CNRS, Universities, specialized engineering schools, or other research institutes (INRA, INSERM, etc.). On average, about half of the research staff in Inria teams are from partner institutions, giving Inria a leverage effect. Hereafter, research at Inria always means research in Inria's teams including partners. The average size of a team is 18 people—counting researchers and faculties, postdoctoral fellows, Ph.D students, and research engineers—but with a large variation, ranging from 3-5 persons for the smallest teams up to 45-50 for the few largest ones.

Cybersecurity has been one of Inria's research priorities for the last fifteen years[1]. Cybersecurity now covers 7% of Inria's activity, with about 30 teams working in this field[2], two thirds of which have cybersecurity as their unique or primary research topic. Altogether, this amounts to about 200 full-time positions. This represents a fourth of the French academic forces in cybersecurity. The other main academic forces are scientists from CNRS and faculty members from universities and engineer schools hosted in CNRS UMRs[3] but outside of the Inria teams, faculty members from the Institut Mines Telecom (IMT), and CEA scientists. French academic forces in cybersecurity have roughly doubled in the last decade. This growth is continuing but at a smaller pace. While there are some new hires, the growth is mostly due to researchers and faculty members moving to cybersecurity from other fields of computer science (see §1.1).

Research domains

Figure 8.1 describes the proportion of research activities between the main domains of cybersecurity at Inria, in red, and in all French academic entities taken together, in blue. The comparison is instructive. Cryptology is a strength of Inria, with a third of its workforce. This results from Inria's long involvement in number theory, computer algebra, cryptanalysis, and coding theory. Indeed, Inria plays a key role worldwide in the design of new cryptographic primitives or protocols and the cryptanalysis of cryptographic primitives.

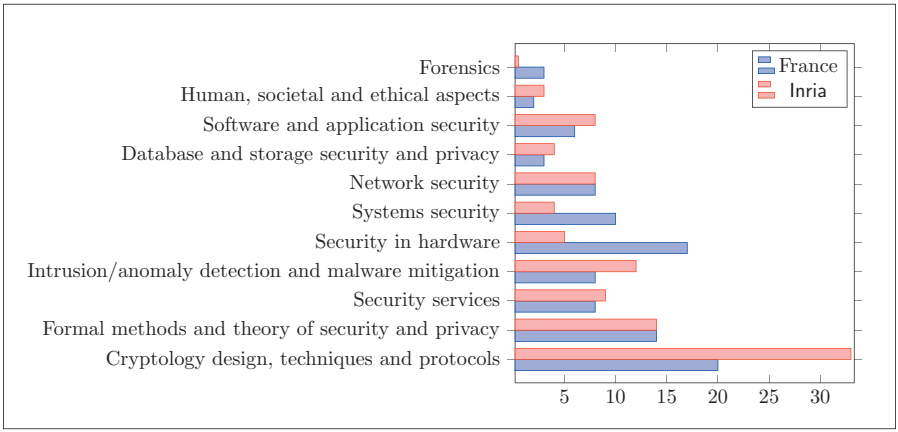


Figure 8.1: Main topics in cybersecurity by % of French academic forces[4]

1. Security was extensively discussed in Inria's Strategic Plan for the period 2003-2007.
2. All Inria teams working in cybersecurity are listed in Appendix A.
3. Mixed (organization) Research Units.
4. Source: cartography of French academic forces in cybersecurity done by the the Allistene Alliance's working group on cybersecurity, see https://www.allistene.fr/files/2018/03/VF_cartographie_2017-06-13.pdf.

Figure 8.2: Geographic breakdown of academic cybersecurity forces

The next most important research domain at Inria is formal methods applied to security and privacy. As explained in §1.1, this is largely due to the transfer of competences coming from formal methods combined with the fact that Inria, and France in general, are very well-positioned in the area of formal methods. Conversely, hardware and system security are underrepresented at Inria, but this area is still well-covered elsewhere in France at CEA, CNRS, and IMT. Notice that there is very little research at Inria on forensics. Nevertheless, in a collaboration with tracIP[5] the **RESIST** team works on the development of a forensics platform dedicated to industrial control systems. Outside of Inria, research on forensics does exist in France[6].

Figure 8.2 describes the geographic breakdown of academic research forces in France. The unit of measure is ETP, which amounts to a full time research activity. This shows that besides Paris the main forces in cybersecurity are in

5. https://www.tracip.fr
6. The activities in forensics in France are probably underestimated, since the gures do not cover research in humanities.

Brittany (Brest and Rennes) and in the Rhone-Alpes region (Lyon and Grenoble). Next comes Nancy and the Côte d'Azur (Nice and Sophia Antipolis). For Inria, the forces are in the Paris region (Ile de France) and then comes Rennes and Nancy.

Non academic forces

There are also important non-academic research forces. Indeed, research is also conducted in private companies, although it is usually more applied. Moreover, there are institutional bodies, such as DGA[7], which is part of the French ministry of armies, and the ANSSI[8], the French cybersecurity agency, which is part of the SGDSN[9] placed under the Prime Ministry, which have toplevel cybersecurity expertise, including some research activity, and play a key role in defining and conducting the French cybersecurity policies. There are also some laboratories depending on the Ministry of the Interior, such as the CREOGN[10]. Research regarding privacy issues and regulations are also conducted in the LINC[11] innovation lab of the CNIL[12], the French data protection authority. Inria teams regularly have scientific collaborations on some specific projects with most of these organizations.

Community animation

There are also a number of associations that play an important role in the animation of the cybersecurity community. The CNRS has created a pre-GDR[13] on cybersecurity [14] in 2016, whose purpose is to animate the French academic community in cybersecurity, in particular via the organization of workshops or summer schools.

The Allistene alliance [15] has created a working group on cybersecurity [16] where Inria and the main academic actors have representatives and whose aim is to exchange information, to build a shared vision, to conduct general-purpose studies such as the cartography of academic forces described above, and to coordinate actions like the participation of the Allistene members to the FIC[17].

Finally Inria is part of the working group devoted to research and innovation of the CoFIS (Comité de la Filière industrielle de sécurité), whose main role is to foster the French security industry by proposing targeted actions to increase both competitiveness and security at the national and European levels.

7. Direction Générale de l'Armement.
8. Agence Nationale de la Sécurité des Systèmes d'Information.
9. Secrétariat Général de la Défense et de la Sécurité Nationale.
10. Centre de recherche de l'Ecole des officiers de la gendarmerie nationale
11. Laboratoire d'Innovation Numérique de la CNIL
12. Commission Nationale de l'Informatique et des Libertés
13. A GDR (Groupement De Recherche) is a structure lead by CNRS to animate the French scientic academic research community in a particular area.
14. http://gdr-securite.irisa.fr/index.html
15. http://www.allistene.fr/
16. https://www.allistene.fr/organisation-allistene/groupes/groupe-cybersecurite/
17. Forum International de la Cybersecurité

Inria is also part of the ACN[18] professional association whose role is to federate and represent the main industrial actors in cybersecurity. HEXATRUST is another important association composed of 29 SME's in cybersecurity, where Inria (and academic research) is however not represented.

7.2 Education

Education is an important issue, as there is a huge lack of expertise in cybersecurity at all levels, but high-level expertise remains the crux. This situation has been known for a few years, thus some engineering schools and universities have created new cybersecurity programs. Although Inria is not a university, education and knowledge transfer are still one of its important missions, and Inria contributes to education in several ways.

Many Inria researchers are teaching advanced cybersecurity courses at engineering schools and universities, especially in master-level courses, and in dedicated summer schools. Inria also created a couple of MOOCs on cybersecurity.

More importantly, most researchers are advising PhD students. Those that do not continue in academia will be hired by industry or other institutions; this is one of the most efficient ways of transferring expertise. The number of Ph.D students is about the same as the total number of researchers and faculty members, a proportion that is roughly the same for cybersecurity and other research domains. Since the industry is lacking cybersecurity experts, it is important to increase the number of students in cybersecurity at all levels, including Ph.D's. Inria could increase its number of Ph.D's, but only if there is a corresponding increase in Ph.D funding for cybersecurity and in good Ph.D candidates. Hence, the main challenge remains to attract more young students to computer science in general and to cybersecurity in particular (see §8.2.4).

7.3 Inria's impact in cybersecurity

France is one of the worldwide leading countries in cybersecurity. This strength is due to its academic forces including those at Inria. Besides strong research results and maintaining the highest level of expertise in most subdomains of cybersecurity, Inria's research is also a key contribution to the community.

France and Inria have long been participating in cryptanalysis challenges, holding several factorization records. This effort is necessary to check the state of the art in cryptanalysis continuously, both in terms of algorithmics and computing power, and to recommend adjustments of cryptographic keys accordingly or a change of cryptographic primitives. This is a key contribution to the community.

18. Alliance pour la Confiance Numérique

Researchers are also often part of standardization committees, such as IETF. This is often a time-consuming long-term effort but it is important to raise the quality of the standards.

Inria's startups

In comparison with its research impact in cybersecurity, only a few startups in cybersecurity are directly emerging from Inria teams.

CRYPTOSENSE

Founded in 2013 and located in central Paris, Cryptosense[19] is an academic spin-off of Inria and Ca'Foscari University of Venice. They develop software based on the academic research carried out by Graham Steel and his colleagues from the Prosecco team and the former Secsi teams.

Their main product, Cryptosense Analyzer, discovers security flaws in cryptographic systems using a variety of techniques. Cryptographic hardware is treated as a black box and tested using fuzzing techniques. The fuzzing results are used to infer a logical model of the device that is then analyzed by model checking. Cryptographic software is tested by instrumenting calls to cryptographic libraries. A model is inferred from these calls which is analyzed against a database of cryptographic usage rules. The libraries themselves are tested for a suite of cryptographic attacks.

Cryptosense software is used by a number of international banks, payment providers, technology companies, government agencies, and hardware manufacturers, in Europe and North America.

CYBER-DETECT

The Cyber-Detect[20] start-up was created from research on malware carried out in the former Carte team and continued in the Carbone LORIA team. Since several years, they developed a solution, dubbed morphological analysis, to analyze binary codes and detect malware. Morphological analysis is a method that consists in abstracting the control flow graph of a binary code and fully automatically build signatures from this abstraction. The recombination of signatures allows one to identify malicious functionalities. The resulting prototype is now commercialized by Cyber-Detect as a solution to help reverse analysis and forensics.

LYBERO.NET

The Lybero.net[21] start-up, issued from the Inria Nancy – Grand Est research center, proposes two commercial services. The first one is a quorum-based digital

19. https://cryptosense.com/
20.https://www.cyber-detect.com/
21. https://lybero.net/

distributed escrow that enables the recovery of a forgotten key (or any digital content) if a pre-defined quorum of administrators has been gathered.

The second service is CryptnDrive, a secure cryptographic driver that enables a secure and easy exchange of files within and outside an organization. Through a regular web browser, users can securely exchange important documents without ever exchanging any password with the remote destination, confidentiality being guaranteed by end-to-end encryption. Additionally, this service leverages on the quorum-based escrow in order to recover from lost credentials.

MALIZEN

Although many tools are used to secure our information systems proactively, we regularly see that attackers find ways to circumvent them or exploit flaws. Monitoring solutions allows one to detect, characterize, and respond to these intrusions in the majority of cases. However, it sometimes becomes necessary to handle exceptions or verify that these systems are functioning well. Faced with the masses of data collected from monitoring, security experts are often poorly equipped and have difficulties to respond to security incidents.

Fuelled by the promising results of collaborative research projects between Inria and DGA-MI, Malizen[22] is a startup whose goal is to equip experts in cybersecurity with hardware and software bricks to help them better respond to security incidents.

By using graphical user interfaces and data visualization, especially designed for cybersecurity, the experts are reintegrated into the analysis process. Those responsible for analyzing intrusions can explore their security data more intuitively and better understand critical situations.

Malizen is supported by Inria using research from the **CIDRE** team.

7.4 High Security Laboratories (LHS)

Inria has also two *High Security Laboratories* (LHS) located in the research centers of Nancy and Rennes that are shared with Inria's local partners, CNRS and universities, and also supported by the Grand-Est and Bretagne regions, and DGA in Rennes.

Both LHS are in secured rooms. The Nancy-LHS has a network telescope that captures malware code and attack logs and permits experimenting with Internet probes. It has a closed network for sensitive experiments such as malware code analyses and an isolated room, not connected to the Internet, where highly sensible information can be processed and confidential hardware and software experiments can be conducted. The Rennes-LHS hosts three platforms: one for experimenting

22. https://malizen.com/

with malware and ransomware infection and repair, which contains a database of computer viruses and ransomware; one for electromagnetic observation and analysis to experiment with side-channel attacks; and the last for electromagnetic hardware fault injection.

8

Conclusions and recommendations

While cybersecurity was not an issue for the general public two decades ago, everybody is concerned nowadays: states, industries, and citizens. Cybersecurity is a hot topic with many economic, societal, political, or geopolotical security issues at stake and will very likely remain so in the next decades. We conclude this white book with the list of research challenges that we have identified and a few general recommendations related to the organization of research in cybersecurity and the interaction between researchers in cybersecurity and the society.

8.1 Research challenges

While the challenges identified throughout this white book are not the only important topics, we believe them to be of particular importance and we recommend they be treated as a priority. Therefore, and for convenience, we recall them below. Each challenge can be put in context in the section where it was first introduced.

8.1.1 Hardware-targeted software attacks *(see 2.1)*

Attacks against information systems do not usually involve the hardware layer but exploit a software vulnerability. However, recent attacks, such as Rowhammer, Spectre, or Meltdown, have shown that attacks implemented in software can exploit performance optimizations of the hardware. This new type of attack is especially dangerous as it makes hardware attacks possible at a distance, as opposed to classical side channel attacks. It is not yet completely clear how the current proof of concept attacks can be "industrialized," but they pave the way to a new class of serious attacks. Therefore we need to get a better understanding of how such attacks could possibly be deployed, propose a clear typology of this new kind of attack, and propose countermeasures, both at the hardware and the software levels. This task requires expertise at the hardware, firmware, and operating system levels. The countermeasures can also be difficult to design as they may require to revisit crucial optimizations used for years, such as speculative execution.

8.1.2 Security and usability *(see 2.3.2)*

Very often, when users request a service, they are willing to sacrifice security, and bypass an annoying security mechanism, if it prevents them from using the service. In order to avoid this problem, security must be as transparent as possible. Even though complete transparency is not always possible, security services must be as simple as possible to use. Work is needed to propose interfaces and security mechanisms that are suitable for nonexpert users, that ensure the user is well aware of the consequences of their actions, and that prevent users from making errors that compromise security. Designing such usable security mechanisms calls for interdisciplinary research typically including experts in cognitive sciences.

8.1.3 Post-quantum cryptography *(see 3.1.3)*

Building a universal (as opposed to special purpose) quantum computer is widely believed to become feasible in the next decades. Therefore, it is important to think now about quantum-resistant cryptography, as some information that is encrypted today may still be sensitive in, say, 50 years. Most asymmetric cryptography used today is based either on the hardness of factoring or computing discrete logarithms, these problems are both known to be efficiently solvable by a quantum computer. Hence, there is a need for alternatives: lattice-based, code-based, and multivariate-based primitives are the most prominent candidates. It is urgent to perform an in-depth security analysis of these new schemes.

8.1.4 Computing on encrypted data *(see 3.2.2)*

The need for computing on encrypted data has emerged, in particular, with the appearance of the cloud and outsourced computation. In cryptography, this problem can be solved using homomorphic or functional encryption. Gentry showed in 2009 in his breakthrough paper that it was indeed possible to construct a fully homomorphic encryption (FHE) scheme. However, this construction remained theoretical and was completely impractical due to its poor performance. Since then, significant progress has been made on FHE schemes, achieving approximately a still very low speed of 50 logical gates per second. Significant progress will have extremely useful applications for privacy preserving cloud computing, where any technical advance may quickly be exploited as an economical advantage.

8.1.5 End-to-end formally verified cryptographic protocols *(see 3.3.4)*

As the security of cryptographic protocols is extremely difficult to ensure (pencil and paper proofs regularly contain errors), the use of rigorous, formal methods appears increasingly as the only way to achieve the expected security level for this class of systems. Therefore, the area of computer-aided security proofs is an increasingly important topic and must include all aspects from the specification down to the implementation. Recent work, in particular around TLS 1.3, have shown that this is now achievable.

However, such proofs still require carefully crafted code and a very high level of expertise. Leveraging the proof techniques to make them applicable to more general code and usable by a wider audience is now the main challenge. Different protocols often ensure different security properties, but existing tools for verifying certain properties, such as anonymity, do not yet have the same maturity as tools for verifying authentication properties. Yet another challenge is to consider stronger adversary models, e.g., an adversary that may control part of the computer through malware.

8.1.6 Intrusion detection for encrypted networks *(see 4.4.1)*

Nowadays, intrusion detection is essentially realized at the network level. If, as expected in the near future, the traffic were more systematically encrypted, which would of course be a good practice for security and privacy, the analysis of the network packets would become *de facto* inoperative, apart from the header analysis. Therefore, it becomes important to study and design new mechanisms for monitoring information systems and producing alerts, at the application, middleware, operating system, and even firmware or hardware levels.

8.1.7 Understanding privacy and deriving practical tools *(see 5.1.6)*

Understanding privacy principles and regulations is the foundation of any activity in privacy. Although this is not a new domain (e.g., the "Loi Informatique et Libertés" was adopted in 1978), this area has recently experienced major evolutions with the new GDPR European regulation and at the same time new opportunities to collect personal data. As a consequence, understanding the concepts and the regulation is a first necessity. Being able to derive practical tools is another one: even though the GDPR promotes several concepts and goals, it provides little guidance about the effective implementation of these new regulatory provisions.

In particular the GDPR introduced the right to data portability whereby a user can retrieve their data in a human readable and machine portable format. This right opens new research areas around individualized management and control over one's personal data. The goal is to empower citizens to leverage their personal data for their own good, which calls for secure, extensible, and sovereign personal cloud platforms, three conflicting goals that open new research challenges (see e.g., §5.2.4).

8.1.8 Open data and anonymization *(see 5.2.3)*

Open data initiatives may sometimes mean releasing databases that contain sensitive, personal information. To ensure privacy of the individuals, data need to be anonymized. Robust anonymization, that effectively resists de-anonymization attacks, is an active and hot research topic. If differential privacy has become a key scientific tool to achieve provable anonymization guarantees, challenges remain on its application, for instance in order to improve the privacy/utility trade-off.

8.1.9 Towards a privacy preserving smart connected world *(see 5.3.6)*

Our connected world experiences an unprecedented growth in terms of personal data collection, with practices that are increasingly intrusive for the citizen's intimacy. Surfing the web, using smartphones and other smart devices, driving a connected —and soon to be autonomous— car are activities that generate personal data leaks. The lack of transparency (many services and devices behave as black boxes) and lack of user control (how to express consent or opposition when there is no information, nor user interface) are major issues.

Identification of such hidden behavior is hindered by the number and complexity of underlying technologies specific to each domain. For instance, identification of tracking practices in a web page requires advanced JavaScript execution analyzes, while monitoring of smartphone applications needs dedicated frameworks, and monitoring of certain wireless communication technologies remains mostly unsolved. The analysis of these data flows is required to assess potential privacy leaks, e.g., in a smart home.

Such challenging and diverse research activities are essential to bring transparency, highlight good and bad practices, and enable regulators to enforce data protection laws. As such, this research directly helps in the shaping of our future smart connected world.

8.1.10 Securing the Internet of Things (IoT) *(see 6.2.1)*

Security in IoT is a major challenge: attacks are still relatively easy (many devices have not been designed with security in mind), invasive (e.g., pervasive in our lives), and potentially with a major impact due to the multiplication factor made possible by the large number of devices available and to the direct implications some of them have in the physical world (e.g., connected cars). The research directions are manifold, with for instance the ability to securely update embedded devices' software, the design of lightweight cryptographic primitives adapted to limited resources, the analysis of the security of new low-power wide area wireless technologies, the detection and mitigation of intrusions or misbehaving devices, and the need for secure-by-design frameworks, protocols, and operating systems.

8.1.11 Secure Industrial Systems *(see 6.2.2)*

Industrial systems rely more and more on software mechanisms that can be attacked. Their security has thus become a major issue, especially since the consequences of an attack against such systems can be dramatic. Although traditional security approaches seem applicable to the case of industrial systems, their specificities require reviewing the traditional security mechanisms to adapt them to this new context. In particular, the communication protocols used in this context cannot be modified overnight. There must be a transition during which legacy communications should be embedded in secure protocols. In addition, real-time control of the system is usually required. Security must thus also be applicable in real time. Finally, it is often impossible to modify industrial devices. Therefore, preventive security mechanisms cannot be used. It is then mandatory to use reactive security and thus extremely important to study how effective attack detection mechanisms could be deployed in this context.

8.2 General recommendations

We give here a few general recommendations related to the interaction between researchers in cybersecurity and society, the organization of research in cybersecurity, and to conclude, the importance of cyber-resilience.

8.2.1 Society should profit more from academic scientific expertise

For most topics of cybersecurity, there are academics with very specific technical expertise who can provide useful scientific advice and who occasionally already do so. However, scientists are often underrepresented in national or industrial advisory committees, in comparison to industrial members. Moreover, some positions or decisions taken, at different levels, show a lack of scientific advice.

8.2.2 Transfer of expertise between cybersecurity and other domains

The need for cybersecurity expertise is striking in most of its application domains, e.g., industrial systems, medical systems, robotics, and, perhaps even more crucially, IoT (see §6 for the complete list). Unfortunately, cybersecurity is not yet sufficiently identified as a priority in these domains, and thus not taken into account in early design stage of applications, failing to uphold the security-by-design principle. Conversely, research in cybersecurity sometimes needs more expertise in formal methods or artificial intelligence for instance. A transfer of expertise between research teams in cybersecurity and these different domains is needed to turn the tide.

8.2.3 Promoting security also as an experimental science

System security and network security appear to suffer from a lack of prestige in the academic world, at least in France. This may be a cultural issue, partly due to the fact that research in these domains is more experimental and technological. Another concern is the lack of real-world datasets. For instance, intrusion detection and alert correlation suffer from a lack of real data on which it is possible to test and compare new mechanisms proposed by researchers. Helping researchers to access or generate such data appears crucial.

8.2.4 Education

Education is essential to security (see §1.3 and §2.3.3) and major dissemination efforts should be made for all audiences: teachers, faculty members, researchers, industrial actors, and specialists, to everyday citizens including young children. Concerning awareness, there is need for scientific mediation in cybersecurity, targeting the society in general, and with a special focus on scholars and teachers in primary and secondary schools. On the professional side, Ph.D training in cybersecurity areas is of course a natural education tool. But dedicated professional master degrees or final-year programs of engineering schools could already bring

to students a clear vision of the possible future of information security (beyond the industrial state of the art). In this context, MOOCs could be an efficient dissemination and education vector, as well as a major communication tool, as they potentially reach several thousand users during each MOOC session and require an active participation from users.

8.2.5 Cyber-resilience by design

Many recent reports on the analysis of cybersecurity threats claim that states will have to face massive cyber attacks. Are we well-prepared to resist a cyber tornado? The resilience of critical infrastructures and, in particular, of operators of vital importance (OVI) is a real issue. A large part of this challenge is actually organizational and engineering-related, rather than academic—and the ANSSI has already spent many efforts increasing our resilience. Nevertheless, research can help: the security-by-design principle should also apply to cyber-resilience. This means that cyber-resilience should not be considered after the fact, but taken into account from the initial design when constructing digital systems, networks, and digital infrastructure.

Moreover, many subdomains of cybersecurity are also indirectly implied in this challenge: reactive security or malware detection are of utmost importance; preventive security to ensure state-of-the-art protection levels is also relevant; formal methods including proofs of protocols, their implementations, as well as critical parts of operating systems can have a significant long-term impact on communication system resilience.

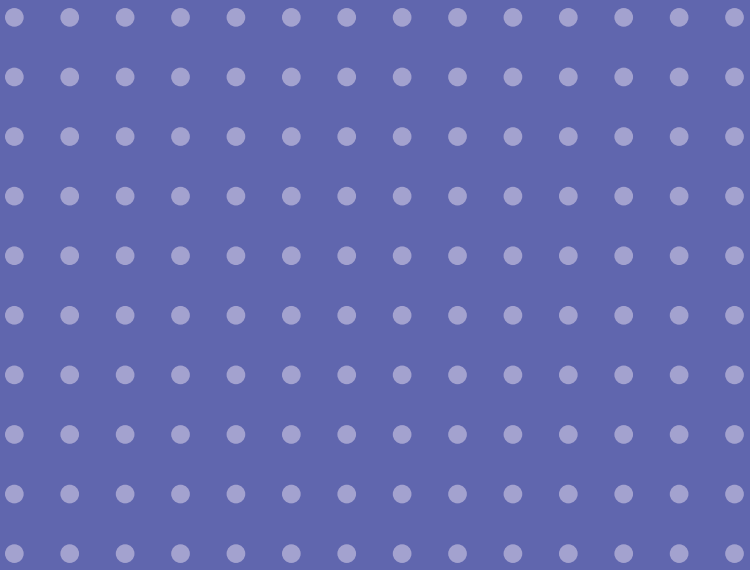
Final remarks

In recent years, a major national effort has been made to strengthen the security of the state and vital operators' information systems. In particular, the ANSSI (Agence Nationale pour la Sécurité des Systèmes d'Information) and the DGA (Délégation Générale de l'Armement)—to give only two examples—have recruited many top talents to adapt their human resources to identified needs.

Nevertheless, even if the awareness of the importance of cybersecurity research and training has increased, no such recruitment effort to enroll more faculty members and researchers has been made. In the area of cyberresilience, human resources are essential: we therefore call for a continuation of the efforts undertaken by some national institutions, and that these efforts be extended to research institutions, universities, and schools.

Similar to the countries making a significant effort to improve their cybersecurity capabilities, France should maintain or even reinforce its academic forces and all its innovative ecosystem in this field.

Appendix A



List of Teams

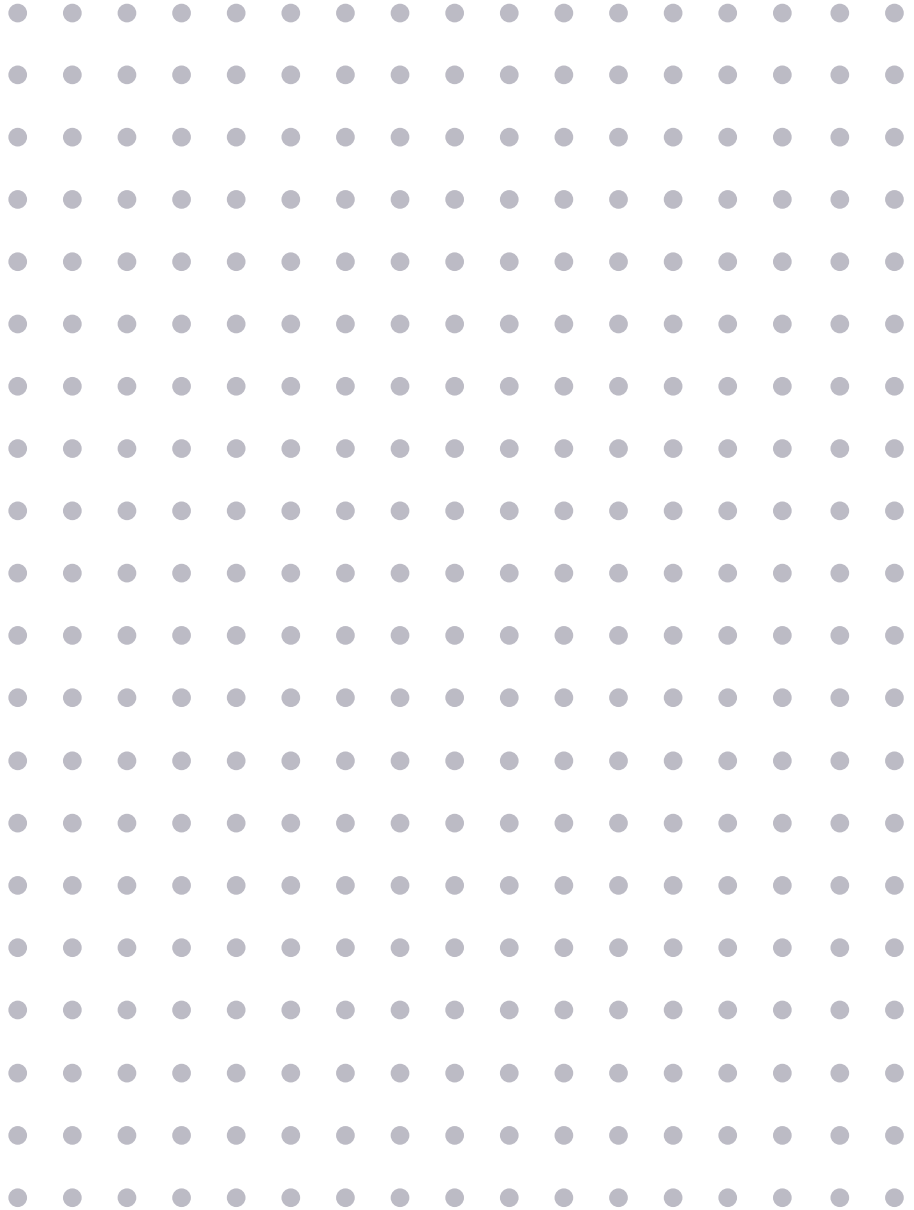
LEGEND: Teams whose name is in capital letters have their main activity in cybersecurity. Other teams whose name is just capitalized in straight font have some significant activity on cybersecurity, while teams whose name is capitalized and slanted have only a marginal contribution to cybersecurity. Teams websites and 2017 annual reports can be found at the URLs https://www.inria.fr/en/research/research-teams and https://raweb.inria.fr/rapportsactivite/RA2017/.

- ***Almanach,*** *Automatic Language Modelling and ANAlysis & Computational Humanities,* (web, report), 46
- **Antique,** *Static Analysis by Abstract Interpretation,* (web, report), 132, 137
- **ARIC,** *Arithmetic and Computing,* (web, report), 56, 63
- ***Avalon,*** *Algorithms and Software Architectures for Distributed and HPC Platforms,* (web, report), 127
- ***Aviz,*** *Analysis and Visualization,* (web, report), 132
- ***Cairn,*** *Energy Efficient Computing ArchItectures with Embedded Reconfigurable Resources,* (web, report), 36, 137
- **CARAMBA,** *Cryptology, arithmetic : algebraic methods for better algorithms,* (web, report), 56, 63, 70
- **CASCADE,** *Construction and Analysis of Systems for Confidentiality and Authenticity of Data and Entities,* (web, report), 56, 63, 71, 127, 132
- **Cedar,** *Rich Data Exploration at Cloud Scale,* (web, report), 46
- **CELTIQUE,** *Software certification with semantic analysis,* (web, report), 83, 137
- **CIDRE,** *Confidentiality, Integrity, Availability, and Repartition, 3* (web, report), 36, 42, 47, 83, 88, 89, 93, 103 121, 127, 132, 141, 146, 157
- ***Coast,*** *Web Scale Trustworthy Collaborative Service Systems,* (web, report), 132
- ***Coati,*** *Combinatorics, Optimization and Algorithms for Telecommunications,* (web, report), 129
- **COMETE,** *Concurrency, Mobility and Transactions,* (web, report), 83, 110, 121, 148
- ***Ctrl-a,*** *Control for safe Autonomic computing systems,* (web, report), 94, 141
- ***Dante,*** *Dynamic Networks : Temporal and Structural Capture Approach,* (web, report), 46
- ***Datasphere,*** *Economie des données et des plateformes,* (web, report), 39
- ***Delys,*** *DistributEd aLgorithms and sYStems,* (web, report), 132
- **Diana,** *Design, Implementation and Analysis of Networking Architectures,* (web, report), 110, 129
- **Diverse,** *Diversity-centric Software Engineering,* (web, report), 47, 122
- **Fun,** *self-organizing Future Ubiquitous Network,* (web, report), 83, 137
- **GRACE,** *Geometry, arithmetic, algorithms, codes and encryption,* (web, report), 53, 56, 64, 132, 138
- ***Graphik,*** *GRAPHs for Inferences and Knowledge representation,* (web, report), 46

EDITORS
Steve Kremer - Ludovic Mé - Didier Rémy - Vincent Roca.

LIST OF CONTRIBUTORS
Laurent Amsaleg - Nicolas Anciaux - Daniel Augot - Frédéric Besson - Nataliia Bielova - Luc Bouganim - Anne Canteaut - Claude Castelluccia - André Chailloux - Konstantinos Chatzikokolakis - Mathieu Cunche - Jérôme François - Teddy Furon - Georges Gonthier - Guillaume Gravier - Gilles Guette - Hélène Kirchner - Jean-Louis Lanet - Cédric Lauradoux - Arnaud Legout - Gérard Le Lann - Daniel Le Métayer - Gaëtan Leurent - Anthony Leverrier - Stéphane Mocanu - Christine Morin - Maria Naya Plasencia - Catuscia Palamidessi - David Pointcheval - Tamara Rezk - Michaël Rusinowitch - Kavé Salamatian - Guillaume Scerri - Nicolas Sendrier - Olivier Sentieys - Éric Totel - Valérie Viet Triem Tong.

LIST OF PEOPLE WHO PROVIDED USEFUL COMMENTS
Gildas Avoine - Emmanuel Baccelli - Hugues Berry - Karthikeyan Bhargavan - Bruno Blanchet - Bertrand Braunschweig - Isabelle Chrisment - Hervé Debar - Claude Kirchner - Philippe Pucheral - Emmanuel Thomé - Frédéric Tronel - Damien Vergnaud - Emmanuel Vincent.



Domaine de Voluceau, Rocquencourt BP 105
78153 Le Chesnay Cedex, France
Tel.: +33 (0)1 39 63 55 11
www.inria.fr