Security and Privacy

Miguel Correia and Luís Rodrigues

Abstract Computer security or cybersecurity is concerned with the proper functioning of computer systems despite the actions of adversaries. Privacy is about a person or group ability to control how, when, and to what extent their personally identifiable information is shared. The chapter starts by defining security and privacy and explaining why they are problems. Then, it presents some of the scientific and technological achievements in the two areas, highlighting some research trends. Afterwards, the chapter relates security and privacy to the main topics of the book: machine learning as part of artificial intelligence. Finally, the chapter illustrates the relevance of ML in the area using censorship resistance as an example.

1 Introduction

Computer security, also designated *cybersecurity*, is concerned with the proper functioning of computer systems despite the actions of adversaries (hackers, cybercriminals, etc.). This proper functioning is expressed in terms of properties such as confidentiality, integrity, and availability. The discipline emerged in the late 1960s in the US defense context with concerns about the confidentiality of classified information stored in computers. A 1970 report of the *Defense Science Board* (Ware 1970) stated that:

> With the advent of resource-sharing computer systems that distribute the capabilities and components of the machine configuration among several users or several tasks, a new dimension has been added to the problem of safeguarding computer resident classified information.

Privacy is an old term with related but different meanings. Privacy can be defined as a person's (or a group of persons') ability to control how, when, and to what

M. Correia (✉) · L. Rodrigues
INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal
e-mail: miguel.p.correia@tecnico.ulisboa.pt; ler@tecnico.ulisboa.pt

H. Sousa Antunes et al. (eds.), *Multidisciplinary Perspectives on Artificial Intelligence and the Law*, Law, Governance and Technology Series 58,
https://doi.org/10.1007/978-3-031-41264-6_5

extent his (their) personally identifiable information is communicated to others (Van Tilborg and Jajodia 2014). The original definition speaks of *personal information*, but today the broader term *personally identifiable information* (PII) is used instead to denote that some data that is not strictly personal can be used directly or indirectly to identify a person (e.g., an IP address or data about colleagues) (McCallister et al. 2010). Relevant properties include anonymity, unobservability, and PII confidentiality. Privacy in the context of computer systems also emerged in the 1960s or 1970s (Miller 1971). The area gained much relevance recently with the European General Data Protection[1] Regulation (GDPR) (European Parliament and European Council 2016) and similar legislation in other countries.

Security and privacy are tightly related disciplines. Privacy to some extend is about the security, mostly confidentiality, of personal identifiers and personal information. However, privacy includes aspects that are barely related to classical security. Two examples are statistical disclosure control (Dalenius 1977) and differential privacy (Dwork 2006). The oldest, and one of the top, scientific conferences in the area, shows the connection between the two topics starting with its name: IEEE Symposium on Security and Privacy.

A note has to be made on how security and privacy should be presented today. Traditionally these topics have been presented in a *negative* way: bad things can happen (and they indeed happen as seen in the news) so we must struggle to prevent them from happening. However, we argue that they should be presented in a *positive* way: the digitalization (digital transformation) of our society requires people and organizations to be able to use computer-based systems with peace of mind, without excessive concerns about security and privacy. These are the goals of the security and privacy scientific and technical areas.

In practice, security and privacy are not 100% achievable in a certain environment or system. This is no surprise, as theft or murder were never erradicated in our society. Therefore, the goal is never to achieve 100% security or privacy, but an adequate level of *risk*. Risk takes into account two factors: the probability of some property being violated and the impact of such violation. The probability depends on the level of vulnerability and the level of threat.

The efforts to increase security and privacy are substantial, both from academia and industry. Today, there are many academic journals and conferences devoted to the matter, including top conferences such as the IEEE Symposium on Security and Privacy, ACM Conference on Computer and Communications Security, Network and Distributed System Security Symposium (NDSS), and Usenix Security. The industry in the area is also large. For instance, recently Gartner forecasted a spending of $150.4 billion in security in 2021, with an increase of 12.4% in relation to 2020 (Whitney 2021). Another indicator is the existence of many industrial fairs worldwide. The largest is probably the RSA Conference, organized in several countries

[1] The title suggests the regulation is about data protection, but in fact it is about *personal* data protection, i.e., about privacy.

yearly, and that attracts more than 40,000 participants only in the USA (Government Technology 2019).

After more than 50 years, security and privacy are vast research areas, with many facets. Therefore, this chapter provides a necessarily limited summary. It provides an overview of important topics and recent developments, with a focus on technology. Other angles such as governance, legal, risk management, security operations, incident management, and digital forensics are not covered.

The chapter is organized as follows. Section 2 defines security and privacy. Section 3 explains why security and privacy are problems. Section 4 presents some of the scientific and technological achievements in the area, highlighting some research trends. Section 5 relate security and privacy to the main topics of the book: machine learning as part of artificial intelligence. Section 6 illustrate the relevance of ML in the area using censorship resistance as an example. Finally, Sect. 7 concludes the chapter.

2 Defining Security and Privacy

Expressions like "system X is secure" or "system Y ensures user privacy" are too vague to be useful. What is useful is to state which set of security and privacy *properties* a system satisfies if correctly implemented and configured, given a set of assumptions about the environment (e.g., the computational power of the adversary).

Security is often expressed in terms derived from *trust* (Veríssimo et al. 2003). Trust is the accepted dependence of a person or (sub)system on a set of properties of another (sub)system. These properties can be of several types, including security and privacy. The *trustworthiness* of a (sub)system is the measure in which it meets the set of properties.

2.1 Security Properties

The three core security properties are confidentiality, integrity, and availability (CIA):

- *Confidentiality:* absence of unauthorized data disclosure;
- *Integrity:* absence of unauthorized data or (sub)system modification;
- *Availability:* readiness of a (sub)system to provide its service.

Notice two aspects. First, security is concerned with guaranteeing these properties in the presence of malicious actions of an adversary. This is expressed by the term *unauthorized*. Second, these properties are related to the impact of malicious actions on data (or information) and (sub)systems, but not necessarily on both. Confidentiality is about data, availability about (sub)systems, and integrity about both.

Two other properties can be considered to be related to integrity:

– *Authenticity:* absence of unauthorized modification of the content or information about its source;
– *Non-repudiation:* absence of denial of authorship of data or actions.

A last property gained visibility recently with the emergence of Bitcoin and other blockchains and distributed ledgers:

– *Decentralization:* absence of dependence on a trusted central authority.

Decentralization does not remove the need for trust, but substitutes trust on individual third parties with trust on sets of parties.

2.2 Privacy Properties

Privacy properties apply to a system that processes personally identifiable information (PII). Consider the auxiliary property:

– *Unlinkability:* given the execution of a certain system, a set of unlinkable PII items can be no more and no less related before and after that execution.

On the contrary of what happens with security, there is some intersection between the classical privacy properties (Pfitzmann and Köhntopp 2001). Consider the identifier of a person (ID) and a set of PII items designated items of interest (IOIs). Privacy can be stated in terms of the following properties:

– *Anonymity:* a person not being identifiable within a set of persons (anonymity set), i.e., unlinkability of the set of IOIs and the ID;
– *Unobservability:* the IOIs are indistinguishable from any IOI at all.

If unobservability is guaranteed, then anonymity is also guaranteed, but the opposite is not true. A related term is *pseudonymity* that means the use of pseudonyms as person IDs. However, pseudonymity is not a property, but a mechanism for obtaining privacy.

The GDPR establishes a set of rights of users before data controllers and data processors (Pfitzmann and Köhntopp 2001). These rights make concrete the "ability to control (...) PII" that appears in our definition of privacy. Therefore, we state them as properties (the names are ours), referencing the article of the GDPR where they are defined:

– *Accessibility:* ability to obtain information about PII being processed (Article 15);
– *Rectifiability:* ability to correct inaccurate PII (Article 16);
– *Erasureability:* ability to delete PII (Article 17);
– *Restrictability:* ability to restrict the way in which PII is processed (Article 18);
– *Portability:* ability to obtain a copy of the PII being processed (Article 20);

– *Withdrawability:* ability to withdraw consent to process data (Article 7) or to object to that processing (Article 21).

3 Security and Privacy Problems

The previous sections stated that there are security and privacy problems that have to be solved. This section presents these problems in more detail.

3.1 Access Control

All security and privacy problems are related to *access*. For example, the initial motivation for security was shared access to computers containing secret (military) information. Another example: the cybersecurity problem today comes mostly from the universal connectivity provided by the Internet. The approach to manage access is twofold. First, it involves separation (or isolation), which can be logical, cryptographic, physical, or temporal. Second, access has to be granted or denied, following some security policy.

This is the context where *access control* comes to play. In abstract terms, there are objects (or resources) that are accessed by subjects (users, processes). Subjects and objects are logically separated, i.e., they are isolated from each other using software and/or hardware mechanisms. Access control is concerned with validating the access permissions (or rights) of subjects to objects. Access control is performed by an abstract component called reference monitor. Whenever a subject wants to access an object (e.g., a user to access a file in an online service), the reference monitor uses an access control database to get information about permissions and evaluates if the user shall be granted access or not. The reference monitor takes the decision and optionally stores data about the access for audit purposes.

The most common access control model is Access Control Lists (ACLs). Each object (e.g., a file) has an ACL that lists the permissions (e.g., read, read-and-write) of each user (e.g., user, admin, any) over that object. There are many other modules, e.g., capabilities, Role-Based Access Control (RBAC) or the current de facto standard, Attribute-Based Access Control (ABAC).

Access control can be discretionary—the access policy is defined by the object owner—or mandatory—the access policy is defined by an administrator for a class of objects.

3.2 Vulnerabilities and Attacks

Access control should be effective if properly implemented and configured—which are challenges themselves—but often another problem allows circumventing access control: vulnerabilities. Computer systems are complex, arguably the most complex creations of humanity. The laptop in which this text is written is the creation of thousands of engineers all over the world, who do not know each other or fully understand the overall system: the laptop in this case. This complexity necessarily leads to errors as engineers are humans.

A *vulnerability* is an error that allows violating a security property. A vulnerability can be introduced during the system design, implementation, or configuration. A vulnerability can be exploited by an *attack*. If the attack is successful, a security property is violated.

Vulnerabilities are so important that they are catalogued. The most important catalogue of vulnerabilities is the Common Vulnerabilities and Exposures (CVE).[2] For instance, in 2014 there was a vulnerability that caused much turmoil called Heartbleed; it received the identifier CVE-2014-0160 in that catalogue (the 160th or 2014). CVE has been registering around 17,000–18,000 new vulnerabilities yearly in recent years.

A particularly dangerous class of vulnerabilities are so-called *zero-day vulnerabilities*. These are vulnerabilities that are known by one or more groups—e.g., an intelligence agency or a hacker community—but have not been publicly disclosed. They allow these groups to attack systems freely, as no protections are deployed against something that is unknown.

A vulnerability can be publicly disclosed in different ways: as part of an update of the software vendor, in the CVE catalogue, in a mailing list like Bugtraq, etc. When that happens, there is the opportunity for organizations and individuals that use the vulnerable software to fix it or protect it, but this disclosure also increases much the probability of the vulnerability being attacked.

Attacks can come through different vectors. A common vector today is called drive-by download. The victim accesses a web page with a browser that contains a vulnerability. The site launches the attack against the browser, exploiting the vulnerability. Note that the site may be legitimate but had itself been a victim of an attack.

3.3 Malware

Many attacks involve *malware*. This term summarizes many others that were previously used in a way that was often inconsistent: virus, worms, Trojan horses,

[2] http://cve.mitre.org/.

backdoors, etc. Malware comes from malicious software and includes all these variants.

A form of malware that is much active today is *ransomware*. When a ransomware specimen enters a computer, it encrypts the content of the disk and requests the payment of a ransom—a monetary fee—often in a cryptocurrency like Bitcoin or Monero, to hide the identity of the attacker. These ransoms vary from hundreds to millions of euros, e.g., as in the high-profile attacks against Maersk (2018) and Colonial Pipeline (2021). Some of these attacks also involve data theft to further pressure the victim into paying.

Another important form of malware are Remote Access Trojans (RATs) or bots, which hide in computer and stay dormant until ordered to do some action. These RATs or bots are controlled remotely by a central server, forming a botnet (a network of bots or robots). These botnets can have thousands of computers and operate as cyberweapons, capable of making systems unavailable using Distributed Denial of Service (DDoS) attacks or stealing large amounts of data or access credentials, among other attacks.

3.4 The Human Factor

The importance of automated attacks that exploit vulnerabilities and/or use malware is undeniable as they are constantly happening. However, many attacks exploit a different class of weaknesses that is much harder to manage: the humans that use computer systems. These attacks are often called *social engineering*.

Phishing is a common attack that aims to steal personal data. This data is often user credentials for a system such as corporate email or homebanking. The attack consists simply in sending emails requesting data. There is no technical vulnerability involved: the data is stolen because users trust the message they receive and act accordingly. Given a large enough set of potential victims, there will always be many that fall for the scam.

A form of attack that combines both technical and social engineering aspects are emails with malicious attachments. Human victims fall for the attack by opening the attachment, but this attachment contains an attack that tries to explore a vulnerability in the victim's computer. This was the first attack vector using the conspicuous Wannacry attack (2017); there was a second that involved exploiting a vulnerability in other computers of the same organization.

4 Scientific and Technological Achievements

This section presents a summary of important and/or interesting scientific and technological achievements in the security and privacy areas. They are organized by subareas, generically inspired by those of the Cyber Security Body Of Knowledge

(CYBOK), "a comprehensive Body of Knowledge to inform and underpin educational and professional training for the cyber security sector".[3] For each topic, we present research trends.

4.1 Cryptography

Cryptography (or cryptology) is an old discipline, around 4000 years-old according to Kahn (Kahn 1996). However, until the twentieth century, its evolution was limited and it remained mostly an art: a struggle between cryptographers that designed coding schemes to protect the confidentiality of messages, and cryptanalysts that would try to break them. These schemes were clever but simple, thus often broken; Mary, Queen of Scots, was beheaded when the encrypted messages she exchanged with her supporters were decoded. The twentieth century first introduced automation with machines like German's Enigma, used in the second World War, but, most importantly, revolutionized the area with the surge of computation and of public key cryptography. Today this is an exciting research area with several top conferences, e.g., the Annual International Cryptology Conference.

A cautionary note: there is some confusion regarding the relation between cybersecurity and cryptography. Some seem to reduce the former to the latter. In fact, although cryptography plays an important role in cybersecurity, cybersecurity includes many topics that are unrelated to cryptography, including most of the areas covered in the following sections.

Classical cryptography involved an encryption and a decryption algorithm. These algorithms were secret to guarantee that the adversary could not read the encrypted messages. In the past century, this idea was slightly modified to become what we now call *symmetric encryption*: the same two algorithms became configurable with a number—designated a *key*—that must be kept secret, whereas the algorithms should be public to be scrutinizable. This led to widely adopted algorithms such as the 1976's Data Encryption Standard (DES), no longer considered secure, and the current Advanced Encryption Standard (AES), published in 1998. Today such schemes are not only used to protect the confidentiality of communications but also of other forms of data, such as the content of disks or individual files.

Symmetric encryption poses a difficulty: the distribution of the secret key, i.e., delivering it to the parties that need it (e.g., sender and receiver). A solution to this problem—*public key cryptography* (or asymmetric cryptography)—was eventually published in 1976 (Diffie and Hellman 1976) and the first public key encryption algorithm, RSA, in 1978 (Rivest et al. 1978). In this form of cryptography there is no longer a single key but a *key pair*. This pair has a *private key* that is supposed to stay secret and a *public key* that can, and often should, be publicly disclosed. If the data is encrypted with the public (resp. private) key, it can only be decrypted with

[3] https://www.cybok.org/knowledgebase/.

the private (resp. public) key. Therefore, if Alice wants to share a secret key K to protect her communications with Bob, she can encrypt K with Bob's public key, so only Bob will be able to decrypt it and get K, even if Trudy obtains a copy of the encrypted K.

Public key cryptography has a second important application: ensuring data integrity using *digital signatures* (Rivest et al. 1978). Alice can sign a message or a document using her private key, so anyone with her public key can verify if the signature is her's. Public keys are typically distributed using *certificates* that contain the identification of the key owner, the public key, and a signature created by a Certificate Authority (CA), among other information.

Signatures are also based in another important type of cryptographic algorithm: *cryptographic hash functions*. These functions are one-way and produce a fixed length output (of, e.g., 256 bits) called hash. Moreover, they satisfy collision resistance properties such as "it is computationally infeasible to find two different inputs that produce the same output".

A current research trend in the area is *post-quantum cryptography* (or quantum resistance) (Alagic et al. 2020). In 1995, Shor published an algorithm that in essence allows obtaining a private key from the public key, effectively breaking public key cryptography algorithms like RSA and ECDSA. This algorithm can only be executed in quantum computers that do not yet exist, but may come into existence within some years. Quantum algorithms against other cryptographic schemes were later presented by Grover (1997) and Simon (1994). In consequence, there is a large research effort on algorithms that remain secure if or when that eventually happens. RSA is based on the difficulty of factoring large integer numbers and Shor's algorithm allows doing that factorization efficiently in a quantum computer; the main approach for post-quantum cryptography is to use different difficult problems that are (arguably) not attackable in quantum computers, e.g., the Module Learning With Errors (MLWE) or Module Learning With Rounding (MLWR) problems.

4.2 Hardware-Based Security

As explained above, security and privacy problems are related to access. The prevention of arbitrary access inside a computer is an important problem as malicious users and malware are always potentially threats. The solution to this problem involves hardware support. Until the 1980s, this support was the one already necessary for operating systems (OSs), e.g., *memory protection* (based, e.g., on paging) that allows isolating processes (programs in execution) from each other (Gasser 1988). Memory protection is implemented by both hardware components (CPU, MMU), and software (OS).

A second example of this kind of support are CPU *execution modes*. A typical configuration is to have the OS running in kernel mode and user processes in user mode. In user mode, the CPU does not allow the execution of some instructions: it generates an exception or silently does nothing when a program tries to execute

one. For instance, processes in user mode are not allowed to execute I/O instructions directly; they have to delegate to the OS their execution using system calls. This separation into a trusted part—the OS—and an untrusted part—user processes—makes sense, but OSs are too large and often vulnerabilities are there found.

In the early 2000s, a consortium of companies, the Trusted Computing Group (TCG), designed a hardware module (usually a chip) to be included in personal computers. This Trusted Platform Module (TPM) provides a set of security services that clearly departed from older hardware security mechanisms. These services were mostly the storage of cryptographic keys and software integrity verification. The former (key storage) is today widely adopted in different forms in mobile devices. The latter supports remote attestation and is also available today in many forms, although not widely used. Later, these services were slightly expanded in the TPM 2.0 specification.

A limitation of the TPM is that the services it provides are fixed. Trusted Execution Environments (TEEs) are a solution for this limitation as they can be programmed with user software. This software is executed in the TEE, isolated from the OS and other privileged software (e.g., BIOS and hypervisor). Today there is a set of TEE technologies that are available in common computers and mobile devices: they are not supported by hardware modules, but by the CPUs themselves. TrustZone is an extension available in many ARM processors. With this technology there is a normal world where the OS and user processes are executed, but also a secure world—the TEE—where security services are run on top of a small kernel. This allows ensuring the confidentiality and integrity of what is in the TEE.

Intel developed the Software Guard Extensions (SGX) for their CPUs. SGX allows running several TEEs on each CPU, designated enclaves. Additionally, to the assurances provided by TrustZone TEEs, enclaves and their data are encrypted while not being executed, thus both logically and cryptographically isolated from the OS and the rest of the computer.

AMD included in their processors the Secure Encrypted Virtualization (SEV) that turns a hypervisor and each of the Virtual Machines (VMs) it executes into TEEs. SEV encrypts each of these TEEs with its own key, isolating them. A second generation of this technology, AMD SEV-Encrypted State (SEV-ES), additionally encrypts the content of the CPU registers when a VM stops running. The third generation of SEV, still to appear, AMD SEV Secure Nested Paging (SEV-SNP), will provide further integrity protection. In 2020, AMD, IBM and others created the Confidential Computing Consortium (CCC) to promote the adoption of these technologies.

The main research trend in the area are applications for TEEs. This technology is being adopted in different areas, from SGX in blockchain to TrustZone in Internet-of-Things devices.

4.3 Cloud Computing

Cloud computing is a model in which computing is provided as a service (Armbrust et al. 2010). In the typical setting, there is a company—the Cloud Service Provider (CSP)—that provides services in a pay-per-use mode, i.e., the consumer pays for the service it consumes. This contrasts with the classical model in which the consumer first buys hardware, then uses it. Instead, with clouds the consumer uses as much resources as it needs, during the period it needs, without an initial investment. This adaptability of the resources used to consumer needs is often called elasticity. There are three main service models: Infrastructure as a Service (IaaS), in which the CSP provides VMs, networking and storage; Platform as a Service (PaaS), where the CSP provides components to build and run applications; and Software as a Service (SaaS), in which the CSP provides applications.

This approach of delegating software and data to a third party, the CSP, involves risks: data loss, data theft, malicious insiders, misconfiguration, etc. (Cloud Security Alliance 2019). Managing these problems requires a holistic process for building and configuring software, which includes a large set of technological solutions. The topic is too vast for a chapter, so we focus on a couple of examples of advanced mechanisms.

Many consumers use cloud services to store data. In case they consider a single CSP does not provide an adequate level of availability, integrity, or confidentiality assurances, they can resort to a set of CSPs forming a *cloud-of-clouds* (Bessani et al. 2013). The idea is to store the data (files) in several clouds, protected using encryption and digital signatures, with secret keys protected using secret sharing, and applying erasure coding to reduce the total stored data size. This requires Byzantine fault-tolerant replication, so the upload and download protocols are more sophisticated than simply storing copies of files in several places.

Other consumers may deploy applications on PaaS services but be concerned that these applications are attacked and their state (data) modified. This can happen, e.g., if someone steals credentials from a legitimate user and uses them to access the application. Removing the effects of such actions from the application storage (database, file system) is often done manually. A solution is to modify the application to track the effects of the requests it receives on that storage, but modifying the applications can be complex or even impossible. Sanare supports automatic recovery and removes the need of modifying the application by using machine learning to associate application requests with storage commands (Matos et al. 2021). The association algorithm—Matchare—is based on a Deep Convolutional Neural Network (CNN) and requires a training phase.

4.4 Digital Money, Assets and Identity

Digital money exists for many years but gained global attention recently with Bitcoin and thousands of others cryptocurrencies. In reality, most money used today has digital form: most payments and transfers are done using computers, not paper or metal.

In the 1980s, some authors presented advanced cryptographic schemes for digital money and payments that paved the way to current cryptocurrencies. A seminal work by Chaum presented a payment mechanism that prevents entities not involved in the payment to determine the recipient of the payment, the time, and amount transferred, while allowing the payer to provide a proof of payment and to disable payment media reported stolen (Chaum 1983). In a following work, Chaum et al. presented a digital money scheme, which allows payments offline while providing proof in case the spender uses the same money twice (Chaum et al. 1990).

In 2008, Nakamoto introduced the first cryptocurrency, Bitcoin (Nakamoto 2008). In relation to previous work, it introduced two differences. First, it does not depend on a third party, but on an open ad hoc group of entities that run the Bitcoin software in their computers (nodes). This additionally ensures availability. Second, it prevents the double spending of money using a chain of blocks (blockchain) to register transactions, while nodes only accept blocks with valid transactions. This digital ledger, or blockchain, is replicated in all nodes and grows when there is consensus on the next block to add. Each block contains a hash of the previous block, obtained by solving a cryptographic puzzle, which makes it hard to modify the blockchain (ensuring integrity) and allows solving consensus.

Eventually many other cryptocurrencies appeared, but also the possibility of running user programs in the nodes, often called smart contracts (Buterin 2014). This programmability of these systems allowed not only the creation of other cryptocurrencies, but also transactionable *digital assets* (or tokens). A type of digital asset that is gaining popularity are Non-Fungible Tokens (NFTs). They represent some collectionable item, like a digital picture, a music record, etc.

Another recent trend in the area is to use blockchains to store identity data. The W3C defined the concept of Decentralized Identifier (DID). The main idea is to allow the DID controller (e.g., the person with that identity) to control the DID, instead of relying on a centralized entity (e.g., a national registry or a company such as Google or Facebook). DIDs can be stored in a blockchain or distributed ledger. Technically, a DID is a small digital document that contains identification data, e.g., a name and a public key. The W3C also defined the notion of Verifiable Credential (VC). A VC provides assurance about information of a certain DID, e.g., that it corresponds to a natural personal that was born in a certain date. Often VCs are not transmitted themselves, but in the form of Verifiable Presentations (VPs) that prove some fact without disclosing undesirable information. For instance, for privacy purposes a VP can prove that a person is more than 18 years old without revealing his/her age.

A recent trend is on the interoperability of this kind of systems, either of the same or different types (Belchior et al. 2021).

5 Security, Privacy, and Machine Learning

Security and privacy have become an arms race. Companies and other organizations are deploying increasingly more sophisticated defenses, but cybercriminals are also becoming increasingly sophisticated. Not surprisingly, machine learning is a powerful weapon in the cybercriminal war, that can be used either to protect or harm computer infrastructures and their users.

Machine learning (ML) can be very effective at capturing and classifying patterns, which is extremely useful to detect suspicious or anomalous behaviour. There are many security-related areas where ML has been applied successfully, including spam filtering (Guzella and Caminhas 2009), fraud detection (Gao et al. 2021), intrusion detection (Buczak and Guven 2016), malware detection (Burguera et al. 2011), vulnerabilities in source code (Shar et al. 2013), among others.

Unfortunately, ML can also be a source of vulnerabilities and attacks (Barreno et al. 2008), as described below. First ML can be used to detect patterns in user behaviour, for instance when they communicate with others, even if the content is encrypted. This can be used, for instance, to detect human rights activists that attempt to escape censorship. In the next section, we discuss this attack in detail.

Moreover, the characteristics of the training data can lead to unexpected or undesirable results, due to ambiguities in the data set or in the classification. An anecdotal example of this effect was a recent incident, where Amazon's Alexa suggested to a 10-year-old girl to touch a coin to the prongs of a half-inserted plug (BBC News 2021). This sort of errors can happen even without the intervention of malicious agents, and are hard to avoid because many ML models cannot be understood by humans, making the task of predicting the outcome almost impossible. Because of this limitation, there is an effort to use techniques that can improve the explainability and interpretability of the ML models (Gohel et al. 2021).

The problems above can be exacerbated by an active attacker that deliberately aims at defining the ML system, to cause the system to malfunction or to steal information from the ML model. For instance, an adversary can carefully edit the inputs to the ML system to evade detection. A common example of this is spam, were the use of misspelled words or unexpected characters may prevent spam to be classified as such. An adversary can also cause a system to operate in a harmful manner. For instance, it was shown that small changes to the visual aspect of street signs could cause automated driving systems to violate speed limits (Wierd 2002). In some cases, an adversary has the opportunity to provide training data to the ML system, and can exploit this to bias the model by providing malicious samples, an attack known as ML *poisoning* (Biggio et al. 2014). Finally, attackers can use an existing model to extract information regarding the training data, obtaining access to data that should be kept confidential (Wang et al. 2020).

An interesting example of the threats associated with the use of ML is the recent Apple proposal to scan the photo library of devices in search for child pornography. Child pornography is certainly a horrendous crime, and secure ways to fight it would certainly be welcome. Apple was proposing to scan the photo library in the user devices, in search for illegal content, to avoid sending the user photos to an external site. This would ensure the privacy of photos for all users except criminals. While the idea has some appeal, many risks have been identified with the approach, which led Apple to postpone the deployment of the system. First, it would be difficult for users to assess if file scanning would just be looking for child pornography, or would also look for other sensitive data (such as health problems, political views, etc.). Moreover, it would be possible that an attacker would send an apparently innocuous photo to a target in order to trigger misclassification, flagging an innocent person. For a more in-depth discussion of the problems of this approach, the reader can refer to (Abelson et al. 2021).

6 Censorship Resistance

In this section, we further illustrate the relevance of ML in the security and privacy arms race, using censorship resistance as an example.

Today, computer networks support the access to most sources of information, including online newspapers, television broadcast, social media, etc. In most countries the operation of these computer networks is controlled by a small number of entities that are under direct control of the government or that can easily be coerced to enforce government directives. This makes it extremely easy to censor access to information.

Examples of wide-scale censorship activities are easy to find: in June 2019 Sudan imposed an internet shutdown (Net Blocks 2019b), and the same happened in Iran in November 2019 (Net Blocks 2019a); there is also evidence that the dissemination of Corona virus related information is tightly controlled by the Chinese government (Ruan et al. 2020; Staff 2020). More subtle forms of censorship can also be found: recently, Reuters reported that Amazon has agreed to censor negative reviews to Xi Jingping's book on their platform (Stecklow and Dastin 2021). In a few cases, censorship can be justified, for instance, to fight criminal activity, such as the dissemination of child pornography content as discussed before, but in most cases it simply deprives citizens of their rights to access free information.

Unsurprisingly, people have developed tools that aim at circumventing censorship. These tools have a dual purpose. First, they aim at allowing users to access information that would be otherwise blocked. Second, that aim at preventing an external observed to detect that the users is accessing such information. This is particularly relevant because, under oppressive regimes, citizens that attempt to evade censorship can be prosecuted, arrested, or even killed. In the next paragraphs we discuss two of these tools, namely *anonymity networks* and *multimedia protocol tunneling* tools.

6.1 Anonymity Networks

An anonymity network is a kind of *overlay network* designed to preserve the privacy of the users. It uses a network of servers that act as *relays* to propagate information, typically between a user and a source of information. Instead of accessing the information directly, the information is routed in the relay network, using multiple encryption layers (a technique known as *onion routing* (Dingledine et al. 2004)). The encryption is set up in a way that prevents any intermediate relay to know the original source or the final destination of the packet (each relay is only aware of the previous and the next hop in the overlay network). Ideally, the mapping between two endpoints would not be possible without the collusion of multiple relays. The most relevant anonymity network today is the Tor network (Tor 2019).

Anonymity networks are not specifically targeted at censorship circumvention, but can, and have been, used for that effect. In fact, if the sensor cannot identify the relays, and cannot block the communication among these relays (unless it completely shutdowns the internet access), it is possible to establish an overlay route from a client residing in a censored region to a relay residing in an uncensored region (for instance, in a different country), in order to access any given internet site. Also, because all communication is encrypted, it is impossible for an entity that observes the traffic of a client to infer which content is being exchanged.

Unfortunately, anonymity networks have a number of limitations. First, the adversary may be able to identify the nodes that provide access to the Tor network (known as Tor *bridges*) and effectively prevent users from accessing the network in censored regions. Furthermore, even if the adversary cannot access the content of the packets being exchanged, it can access the features of these packets to infer what information is being accessed. In this task, ML has proved to be a strong ally of the censor.

There are two relevant attacks that can be used to detect what content a client is accessing, even when it uses a anonymity network: *traffic fingerprinting* and *traffic correlation*.

In a traffic fingerprinting attack, the adversary observes the traffic pattern and tries to match it with known patterns. This is possible, in particular, when users access some known websites. In order to perform this attack, the attacker collects data regarding the packets that are generated when a given site is accessed. Features such as packet size count, packet size frequency, per-direction bandwidth, total time, burst markers, inter-arrival time, etc. are used as patterns known as *website fingerprints* (Liberatore and Levine 2006). Machine learning tools can then be used to learn these patterns and later classify traffic flows collected from end users. Advances in ML, such as the use of modern convolutional neural networks, a technique known as *deep fingerprinting* (Sirinam et al. 2018), has shown to be highly effective, even when clients apply defenses against website fingerprinting, such as adding dummy packets, padding, and/or packet delays, to make classification harder (Dyer et al. 2012; Cai et al. 2014; Juarez et al. 2016).

In a traffic correlation attack, the adversary monitors the network between the clients and the anonymity network and between the anonymity network and the servers, i.e., the traffic between the anonymity network boundaries and the clients/servers outside the network (Danezis 2003; Le Blond et al. 2011; Nasr et al. 2017). Then, the traffic features of the different flows are correlated to find a match, and establish a link between a client and a server. Needless to say, the use of ML, in particular the use of deep neural networks, has also proved to be helpful in performing traffic correlation (Nasr et al. 2018).

6.2 Multimedia Protocol Tunneling

Multimedia Protocol Tunneling is a technique that consists in embedding a covert channel in a multimedia stream. This technique can be used for censorship circumvention by leveraging services that the censor may not be willing to block (Fifield 2017), such as Skype, Zoom, or WhatsApp. Using this approach, a user in a censored region establishes a multimedia call to another user in an uncensored region and then, embeds a *covert channel*, in the multimedia stream; this covert channel can be used to convey standard IP traffic and be used to access censored content. This typically involves replacing part or all of the original multimedia content by the content of the covert channel, encoded in some form (Houmansadr et al. 2013; Li et al. 2014; Kohls et al. 2016; Barradas et al. 2017, 2020). If the multimedia content is encrypted, an adversary has way to inspect the covert channel. Furthermore, if the adversary has no way to distinguish a multimedia call that embeds a cover channel from a call that does not, we say that the channel remains *unobservable*.

Multimedia protocol tunneling is appealing because, in the general case, the censor cannot generally afford to block all multimedia applications, as these are used widely by citizens for daily interactions with family and friends and by companies to perform business. The large number of multimedia channels that are used at any point in time also make the task of observing these channels harder. Unfortunately, these tools can also be vulnerable to traffic analysis, in an attempt to identify patterns that distinguish a normal call from a call that embeds a covert channel. Again, machine learning tools can help the attacker in this endeavour. A study published in the 27th USENIX Security Symposium showed that decision trees and some of their variants are extremely effective at detecting covert traffic with reduced false positive rates (Barradas et al. 2018). Furthermore, recent research has shown that the information required to perform the classification can be collected, in a cost-effective manner, at line speed by leveraging the capabilities of new programmable switches (Barradas et al. 2021).

6.3 Avoiding ML Attacks

As we have discussed above, machine learning tools can be used to perform sophisticate traffic analysis in order to detect anomalies, identify patterns, and perform correlations among different flows. These tools empower the adversary, in particular state-level adversaries to detect attempts to evade censorship. Therefore, modern censorship resistance tools must be designed with these attacks in mind.

Interestingly, new tools are being proposed that can embed a cover channel in a multimedia stream without affecting the key features of the traffic stream. Protozoa (Barradas et al. 2020) is one of such tools. It leverages the WebRTC (Web Real-Time Communication) API to replace real video content, produced in real-time by a multimedia conference tool, by converting content of exactly the same size, shielding the protocol from detection mechanisms based on packet size and packet frequency. Furthermore, Protozoa, unlike several previous protocols that offered very limited bandwidth, can deliver covert channel bandwidth capacities in the order of 1.4 Mbps. Protozoa proved resistant to state-of-the-art classification tools but it is unclear if it is possible to design more sophisticated ML tools, that attempt to exploit other features, such as time-series of inter-packet arrival times, to perform detection.

7 Conclusion

Security and privacy are important aspects of our connected world. The chapter defines the two concepts and explains why they are a problem that must be managed, even if not entirely solvable. The chapter presents an illustrative set of scientific and technological achievements, with an emphasis on current research trends. The chapter also points out links between security/privacy and machine learning, using censorship resistance as a use case.[4]

[4] *See* generally, on the different applications of Machine Learning and AI, in this book A Oliveira and M A T Figueiredo—Artificial intelligence: historical context and state of the art; I Trancoso, N Mamede, B Martins, H S Pinto and R Ribeiro—The impact of language technologies in the legal domain; J Gonçalves-Sá and F L Pinheiro—Societal Implications of Recommendation Systems: A Technical Perspective; A T Freitas—Data-driven approaches in healthcare: challenges and emerging trends; E Magrani and P G F Silva—The Ethical and Legal Challenges of Recommender Systems Driven by Artificial Intelligence; M Lanz and S Mijic—Risks associated with the use of natural language generation: Swiss civil liability law perspective; M S Fernandes and J R Goldim—Artificial Intelligence and Decision Making in Health: Risks and Opportunities; W Gravett—Judicial Decision-making in the Age of Artificial Intelligence; and D Durães, P M Freitas and P Novais—The Relevance of Deepfakes in the Administration of Criminal Justice.

References

Abelson H, Anderson R, Bellovin SM, Benaloh J, Blaze M, Callas J, Diffie W, Landau S, Neumann PG, Rivest RL, Schiller JI, Schneier B, Teague V, Troncoso C (2021) Bugs in our pockets: the risks of client-side scanning. arXiv preprint arXiv:2110.07450

Alagic G, Alperin-Sheriff J, Apon D, Cooper D, Dang Q, Kelsey J, Liu YK, Miller C, Moody D, Peralta R, Perlner R, Robinson A, Smith-Tone D (2020) Status report on the second round of the NIST post-quantum cryptography standardization process. NIST, Gaithersburg

Armbrust M, Fox A, Griffith R, Joseph AD, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I, Zaharia M (2010) A view of cloud computing. Commun ACM 53:50–58

Barradas D, Santos N, Rodrigues L (2017) Deltashaper: enabling unobservable censorship-resistant TCP tunneling over videoconferencing streams. In: Proceedings on privacy enhancing technologies. De Gruyter Open, Minneapolis, pp 5–22

Barradas D, Santos N, Rodrigues L (2018) Effective detection of multimedia protocol tunneling using machine learning. In: Proceedings of the 27th USENIX security symposium. Usenix, Baltimore, pp 169–185

Barradas D, Santos N, Rodrigues L, Nunes V (2020) Poking a hole in the wall: efficient censorship-resistant internet communications by parasitizing on WebRTC. In: Proceedings of the 2020 ACM SIGSAC conference on computer and communications security. ACM, New York, pp 35–48

Barradas D, Santos N, Rodrigues L, Signorello S, Ramos FMV, Madeira A (2021) Flowlens: enabling efficient flow classification for ML-based network security applications. In: Proceedings of the 27th network and distributed system security symposium. ACM, San Diego, pp 1–18

Barreno M, Bartlett PL, Chi FJ, Joseph AD, Nelson B, Rubinstein BIP, Saini U, Tygar JD (2008) Open problems in the security of learning. In: Proceedings of the 1st ACM workshop on workshop on AISec. ACM, Alexandria, pp 19–26

BBC News (2021) Alexa tells 10-year-old girl to touch live plug with penny. https://www.bbc.com/news/technology-59810383. Accessed 1 Dec 2021

Belchior R, Vasconcelos A, Guerreiro S, Correia M (2021) A survey on blockchain interoperability: past, present, and future trends. ACM Comput Surv 54:Article 168

Bessani A, Correia M, Quaresma B, André F, Sousa P (2013) DepSky: dependable and secure storage in a cloud-of-clouds. ACM Trans Storage 9:Article 12

Biggio B, Fumera G, Roli F (2014) Security evaluation of pattern classifiers under attack. IEEE Trans Knowl Data Eng 26:984–996

Buczak AL, Guven E (2016) A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Commun Surv Tutor 18:1153–1176

Burguera I, Zurutuza U, Nadjm-Tehrani S (2011) Crowdroid: behavior-based malware detection system for Android. In: Proceedings of the 1st ACM workshop on security and privacy in smartphones and mobile devices. ACM, Chicago, pp 15–26

Buterin V (2014) Ethereum: a next-generation smart contract and decentralized application platform. White Paper

Cai X, Nithyanand R, Wang T, Johnson R, Goldberg I (2014) A systematic approach to developing and evaluating website fingerprinting defenses. In: Proceedings of the 2014 ACM SIGSAC conference on computer and communications security. ACM, Scottsdale, pp 227–238

Chaum D (1983) Blind signatures for untraceable payments. In: Chaum D, Rivest RL, Sherman AT (eds) Advances in cryptology. Springer, Boston, pp 199–203

Chaum D, Fiat A, Naor M (1990) Untraceable electronic cash. In: Goldwasser S (ed) Advances in cryptology — CRYPTO' 88. Springer, New York, pp 319–327

Cloud Security Alliance (2019) Top threats to cloud computing: egregious eleven. Cloud Security Alliance, Washington, DC

Dalenius T (1977) Towards a methodology for statistical disclosure control. Stat Tidskr 15:429–444

Danezis G (2003) Statistical disclosure attacks. In: di Vimercati SDC, Samarati P, Katsikas S (eds) IFIP international information security conference. Kluwer, Athens, pp 421–426
Diffie W, Hellman M (1976) New directions in cryptography. IEEE Trans Inf Theory 22:644–654
Dingledine R, Mathewson N, Syverson P (2004) Tor: the second-generation onion router. In: Proceedings of the 13th conference on USENIX security symposium, vol 13. USENIX Association, San Diego, p 21
Dwork C (2006) Differential privacy. Automata, languages and programming. Springer, Berlin
Dyer KP, Coull SE, Ristenpart T, Shrimpton T (2012) Peek-a-boo, i still see you: why efficient traffic analysis countermeasures fail. In: 2012 IEEE symposium on security and privacy. IEEE, San Francisco, pp 332–346
European Parliament and European Council (2016) Regulation (EU) 2016/679 of the European parliament and of the council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). Off J Eur Union L 119:1–88
Fifield D (2017) Threat modeling and circumvention of Internet censorship. Ph.D. thesis, University of California, Berkeley
Gao Y, Zhang S, Lu J, Gao Y, Zhang S, Lu J (2021) Machine learning for credit card fraud detection. In: Proceedings of the 2021 international conference on control and intelligent robotics. ACM, Guangzhou, pp 213–219
Gasser M (1988) Building a secure computer system. Van Nostrand Reinhold Co., New York
Gohel P, Singh P, Mohanty M (2021) Explainable AI: current status and future directions. arXiv preprint arXiv:2107.07045
Government Technology (2019) RSA conference 2019: what you need to know. https://www.govtech.com/blogs/lohrmann-on-cybersecurity/rsa-conference-2019-what-you-need-to-know.html. Accessed 1 Aug 2021
Grover LK (1997) Quantum mechanics helps in searching for a needle in a haystack. Phys Rev Lett 79:325
Guzella TS, Caminhas WM (2009) A review of machine learning approaches to Spam filtering. Expert Syst Appl 36:10206–10222
Houmansadr A, Riedl TJ, Borisov N, Singer AC (2013) I want my voice to be heard: IP over Voice-over-IP for unobservable censorship circumvention. In: Proceedings of the 20th annual network & distributed system security symposium, NDSS 2013. The Internet Society, San Diego
Juarez M, Imani M, Perry M, Diaz C, Wright M (2016) Toward an efficient website fingerprinting defense. In: Computer security – ESORICS 2016. Springer International Publishing, Cham
Kahn D (1996) The codebreakers: the comprehensive history of secret communication from ancient times to the internet. Simon and Schuster, New York
Kohls K, Holz T, Kolossa D, Pöpper C (2016) SkypeLine: robust hidden data transmission for VoIP. In: Proceedings of the 11th ACM on Asia conference on computer and communications security. ACM, Xi'an, pp 877–888
Le Blond S, Manils P, Chaabane A, Kaafar MA, Castelluccia C, Legout A, Dabbous W (2011) One bad apple spoils the bunch: exploiting P2P applications to trace and profile Tor users. In: Proceedings of the 4th USENIX conference on large-scale exploits and emergent threats. USENIX Association, Boston, p 2
Li S, Schliep M, Hopper N (2014) Facet: streaming over videoconferencing for censorship circumvention. In: Proceedings of the 13th workshop on privacy in the electronic society. ACM, Scottsdale, pp 163–172
Liberatore M, Levine BN (2006) Inferring the source of encrypted HTTP connections. In: Proceedings of the 13th ACM conference on computer and communications security. ACM, Alexandria, pp 255–263
Matos D, Pardal M, Correia M (2021) Sanare: pluggable intrusion recovery for web applications. IEEE Trans Dependable Secure Comput 1:13. https://doi.org/10.36227/techrxiv.13725991
McCallister E, Grance T, Scarfone KA (2010) SP 800–122. Guide to protecting the confidentiality of personally identifiable information (PII). ISAO, Watthana
Miller AR (1971) The assault on privacy: computers, data banks, and dossiers. Signet, Hamilton

Nakamoto S (2008) Bitcoin: a peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf. Accessed 1 Aug 2021

Nasr M, Houmansadr A, Mazumdar A (2017) Compressive traffic analysis: a new paradigm for scalable traffic analysis. In: Proceedings of the 2017 ACM SIGSAC conference on computer and communications security. ACM, Dallas, pp 2053–2069

Nasr M, Bahramali A, Houmansadr A (2018) DeepCorr: strong flow correlation attacks on tor using deep learning. In: Proceedings of the 2018 ACM SIGSAC conference on computer and communications security. ACM, Torontos, pp 1962–1976

Net Blocks (2019a) Internet being restored in Iran after week-long shutdown. https://netblocks.org/reports/internet-restored-in-iran-after-protest-shutdown-dAmqddA9. Accessed 1 Aug 2021

Net Blocks (2019b) Sudan internet shows signs of recovery after month-long shutdown. https://netblocks.org/reports/sudan-internet-recovery-after-month-long-shutdown-98aZpOAo. Accessed 1 Aug 2021

Pfitzmann A, Köhntopp M (2001) Anonymity, unobservability, and pseudonymity — a proposal for terminology. In: Federrath H (ed) Designing privacy enhancing technologies: international workshop on design issues in anonymity and unobservability Berkeley, CA, USA, July 25–26, 2000 proceedings. Springer, Berlin, pp 1–9

Rivest RL, Shamir A, Adleman L (1978) A method for obtaining digital signatures and public-key cryptosystems. Commun ACM 21:120–126

Ruan L, Knockel J, Crete-Nishihata M (2020) Censored contagion: How information on the coronavirus is managed on Chinese social media. https://citizenlab.ca/2020/03/censored-contagion-how-information-on-the-coronavirus-is-managed-on-chinese-social-media/. Accessed 1 Aug 2021

Shar LK, Tan HBK, Briand LC (2013) Mining SQL injection and cross site scripting vulnerabilities using hybrid program analysis. In: 2013 35th international conference on software engineering (ICSE). IEEE, San Francisco, pp 642–651

Simon DR (1994) On the power of quantum computation. In: Proceedings 35th annual symposium on foundations of computer science. IEEE, Santa Fe, pp 116–123

Sirinam P, Imani M, Juarez M, Wright M (2018) Deep fingerprinting: undermining website fingerprinting defenses with deep learning. In: Proceedings of the 2018 ACM SIGSAC conference on computer and communications security. ACM, Toronto, pp 1928–1943

Staff R (2020) Report says China internet firms censored coronavirus terms, criticism early in outbreak. https://www.reuters.com/article/us-health-coronavirus-china-censorship-idUSKBN20Q1VS. Accessed 1 Aug 2021

Stecklow S, Dastin J (2021) Special report: Amazon partnered with China propaganda arm. https://www.reuters.com/world/china/amazon-partnered-with-china-propaganda-arm-win-beijings-favor-document-shows-2021-12-17/. Accessed 1 Dec 2021

Tor (2019) Tor project: Tor faq. https://2019.www.torproject.org/about/overview.html. Accessed 1 Aug 2021

Van Tilborg HC, Jajodia S (2014) Encyclopedia of cryptography and security. Springer Science & Business Media, Boston

Veríssimo PE, Neves NF, Correia MP (2003) Intrusion-tolerant architectures: concepts and design. In: de Lemos R, Gacek C, Romanovsky A (eds) Architecting dependable systems. Springer, Berlin, pp 3–36

Wang X, Xiang Y, Gao J, Ding J (2020) Information laundering for model privacy. arXiv preprint arXiv:2009.06112

Ware WH (1970) Security controls for computer systems (U): report of defense science board task force on computer security, R-609-1. Rand Corp, Santa Monica

Whitney L (2021) Cybersecurity spending to hit $150 billion this year. https://www.techrepublic.com/article/cybersecurity-spending-to-hit-150-billion-this-year/. Accessed 1 Aug 2021

Wierd (2002) Security news this week: a tiny piece of tape tricked teslas into speeding up 50 MPH. https://www.wired.com/story/tesla-speed-up-adversarial-example-mgm-breach-ransomware/. Accessed 1 Dec 2021