Towards an Internet of Trust

Issues and Solutions for Identification and Authentication in the Internet of Things

Matteo Signorini

TESI DOCTORAL UPF / 2015

Directors de la tesi

Prof. Dr. Vanesa Daza
Department of Information and Communication Technologies
Pompeu Fabra University - Barcelona, Spain

Prof. Dr. Roberto Di Pietro
Department of Mathematics
University of Padua - Padua, Italy

To my wife.

Acknowledgements

I would like to express my sincere gratitude to my advisors Prof. Vanesa Daza and Prof. Roberto Di Pietro for the continuous support during all my Ph.D and for their motivation and unlimited patience. They helped me to grow personally and professionally. I could not have imagined having better advisors and mentors for my Ph.D study.

Last but not the least, I would like to thank my family: my parents for supporting me throughout this experience and my wife, to whom I dedicate this thesis and without whom none of this would ever be possible.

Abstract

The Internet of Things (IoT) (r)evolution is advancing slowly, due to the lack of trust in smart devices that can autonomously interact without human intervention. Standard and consolidated solutions (PKI) and new technologies (hardware intrinsic properties) have strengthened IoT security. However, ubiquitous and powerful attackers able to capture, disrupt and tamper with the environment are still an open problem that requires novel approaches. To address the above issues, this thesis deeply investigates the concepts of identity and authenticity. As regards identity, a new context-aware and self-enforced approach based on the blockchain technology is proposed. With this solution, the standard paradigm focused on fixed identifiers is replaced with a more natural identification approach based on attributes and services that delineates democratically approved names. Moreover, in order to build the basement of an Internet of Trust, authentication approaches are analyzed from both the online and offline perspective to enable smart things in the validation of exchanged messages. Further, a new software approach for online scenarios is introduced which provides hardware-intrinsic properties without relying on any physical element. Finally, PUF technology is leveraged to design novel offline disposable authentication protocols.

Preface

In medieval times cities were surrounded by strong walls and guards were posted at the gates. People willing to enter or leave the city were inspected and questioned about their purposes while treasures were protected against internal and external threats. With our modern eyes we can see those walls and those guards monitoring the accesses to the city as a censorship tool and a violation of freedom and privacy. However, at that time, citizens were thankful for their security.

By the end of 20^{th} century, city walls and guards have been supplanted by firewalls and access control systems designed to protect our resources and sensitive information from attacks. However, unlike medieval cities were transactions and communications were physically accomplished, in the 21^{st} century those activities started to go online thus being more difficult to be controlled. This new virtual world is network-based and all its digital communications and transactions are fundamentally different from the physical world. In fact, in contrast with guards posted at the gates, digital communications usually lack identity properties and can be eavesdropped by unauthorized people. As such, the process of identifying information is usually transferred digitally, across the network, and becomes of paramount importance in our digital world. To give access to our information to outsiders we have to mimic the "who are you?" control accomplished by ancient guards with a process that is called authentication.

Authentication in the 21^{st} century is again rapidly changing as we move from a user centric to a device centric world. Services and devices are changing and becoming more automated and able to collect and profile information by themselves, without any human interaction. As such, knowledge-based

authentication approaches as the one based on passwords, PIN codes or secret questions will not fit in the upcoming digital world. We will not have soon enough guards to protect our digital gates and we will need our citizens to trust each other and protect each other by themselves. Thus, we will soon need to provide an identity to our citizens (no matter if physical or digital) as in the future we will not have any gates and our treasures will be threaten by malicious things willing to access our digital data, the gold of the 21^{st} century.

Contents

List of Figures

List of Tables

Part I

Background

Chapter 1

Introduction

1.1 The Internet of Untrusted Things

In the 80s, personal computers transformed the global economy by giving processing power to ordinary people. Then, in the 90s we have been witnessed the shift from personal computers to mobile devices that have unquestionably made our lives easier and more efficient. As for the revolution brought by mobile devices, today we are living another digital revolution that is quickly changing again the way in which we interact with both the digital and the real world. We are nowadays more concerned about roles, interactions and how the digital and the physical world can be linked together [1]. Furthermore, in last years, devices started to be autonomous both in their behavior and in the way they establish relationships between them. This direct and automated interconnections have rapidly brought the need of a new generation of embedded systems called Cyber-Physical Systems (for short, CPS) [2; 3] capable of controlling physical processes, making them more efficient, reliable and secure [4; 5; 6]. The core novelty of CPS is that rather than having information and communications technologies built on top of a hardware element and acting as an interface to that hardware, they are designed as embedded systems directly built within the hardware element and able to directly communicate with it. The main technologies that fall within this new CPS definition are (a) automation of knowledge work, (b) Internet of Things (for short, IoT), (c) advanced robotics and (d) autonomous vehicle. Among them, the IoT is considered the CPS with the highest impact on the market and an estimated value of 36 trillions of dollars [7], thus being targeted by many research studies over the last years.

The IoT defines an ecosystem where each thing can be any physical or virtual object, identified and reached by other objects and showing smart capabilities [8]. Moreover, such smart things are characterized by embedded electronic components that allow them to sense, compute, communicate and integrate seamlessly with the rest of the network.

The first definition of the IoT has origin in 2000 within the Auto-ID Center group, a development community of the Radio-Frequency Identification (for short, RFID) at the Massachusetts Institute of Technology [9]. As depicted in Figure 1.1, the IoT can be seen as a multilayered environment where each layer identifies a class of things based on their computational power and operational skills. The first layer at the basement of the IoT ecosystem is the one composed by the majority of smart things and is usually referred to as sensory swarm. This layer is the most heterogeneous in terms of resource properties, communication technologies and life span capabilities. The second layer consists of mobile devices or high powerful devices. These devices, already present in our daily life, can be exploited, by the sensory swarm, as a bridge to the standard Internet and also to outsource data for computational or storage purposes in case that the sensory swarm cannot manage them by itself [10]. The final layer is then composed by ubiquitous resources such as the cloud computing.

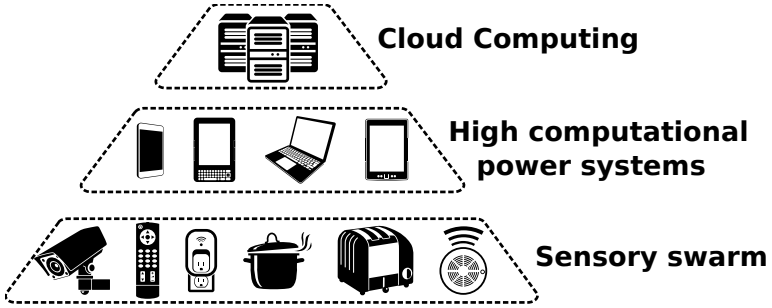

Figure 1.1: IoT pyramid system

Thanks to the Moore's law [11], and to the wide availability of sensors capable of ubiquitous connectivity and long lasting life-cycles, it has been forecast that by 2025 the number of things, specially in sensory swarm layer, will exceed 7 trillions, distributed with an average of 1000 devices per person [12]. By that time, the number of things will become bigger than the number of people [13] and we will witness a shift from a user-centric world to a peer to peer, heterogeneous and device-centric world. All of these connected smart things will then be able to autonomously generate, analyze and share

sensitive data with a global traffic of shared data that has been forecast to reach, by 2019, 4.3 exabytes each month [14].

To realize how concrete and promising is the IoT we will now describe one of the most discussed use case, smart cities [15]. Today, half of the global population live in cities and their concentration is still growing thus leading to an enormous consumption of resources with the consequent decrease in quality, sustainability and security. The realization of a secure and sustainable city requires new and better technologies able to autonomously and seamlessly control and manage resources. As such it is of paramount importance to (a) optimize the usage of available resources, (b) minimize the environmental impact and (c) increase the safety, health and wellness of citizens.

IoT can play a fundamental role for the realization of smart cities as it can improve traditional aspects such as vehicles, buildings, energy, living and governance into their automated and self-managed smart versions. On one hand, mobility services can be created to improve private and public transportation thus enhancing citizen movements within the city. On the other hand, traffic jams can be improved thanks to smart traffic lights and sensors distributed throughout the city that are able to monitor street conditions to plan the waste collection service and to perform environmental monitoring such as water level or air pollution [16]. All this information, gathered by sensors, is collected and shared among different services or devices to improve the efficiency of the city. However, such an impressive amount of data requires an efficient management. In fact, the mixture of heterogeneous information requires a common platform to manage, collect and re-distribute it. Such a platform can operate as a control center thus ensuring interoperability, coordination and service optimization. Things living in the city as well as citizens and authorities would then benefit and interact with such a control center thus indirectly creating additional data, again collected and managed by the control center in an infinite loop.

So far, the first wave of the IoT has focused on high-value applications such as monitoring jet engines [17] or remotely managing health-care systems [18]. Other applications are emerging as well but the demand has been slow to take off in almost any area. This is mainly a result of the lack of trustworthiness in smart devices that scares people about autonomous interactions without human controls. In the rest of this thesis we will analyze the concepts of identity and authenticity within the IoT and how they can be improved both in online and offline scenarios.

1.2 Motivations

The main purpose of this section is to introduce IoT identity and security open issues in order to better understand the motivations behind the solutions that have been proposed in this thesis. Remaining focused on the toy example of smart cities, it is clear that for the IoT to practically emerge, many different challenges have to be solved ranging from hardware elements, architecture designs, communication protocols, service discovery and last but not least data security and privacy. All the above aspects need better approaches as they weaken people trust in the IoT and slow down its adoption in the real world.

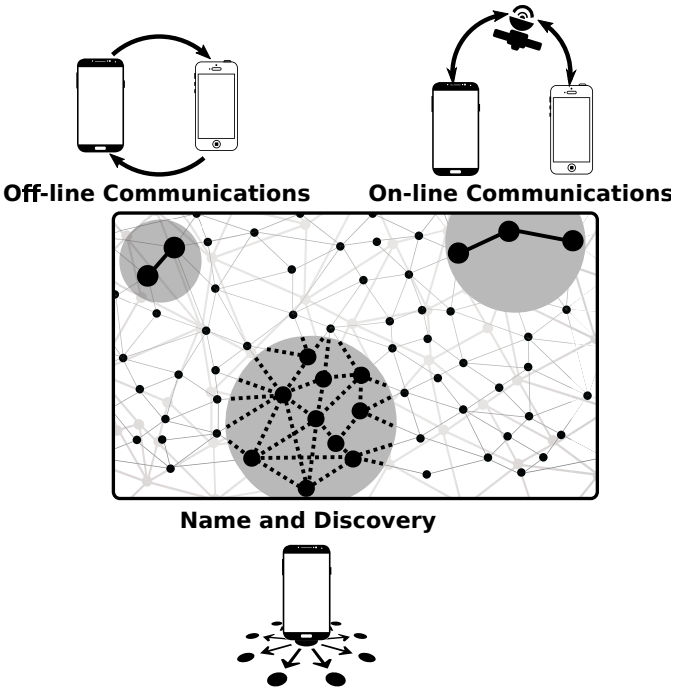

Figure 1.2: Trustworthiness requirements within the IoT

As a major concern, all the above open issues can be grouped under the main concept of digital trust. Trust has many different meanings depending on the context. Even restricting the search to computer science shows different definitions of trust [19]. However, regardless of the definition, the trust that was originally at the basement of the Internet is now, in the *post-Snowden era*, over. Figure 1.2 depicts two aspects of digital trust that are of paramount importance for the IoT to be widely adopted:

- **Trusted Discovery**: depicted at the bottom of the Figure 1.2, it is focused on the concept of identity within the IoT. More in detail it has to be provided a way to easily and securely reach other services or devices within the IoT. This aspect of digital trust deals with the capability of identify and recognize the surrounding environment;
- **Trusted Interaction**: depicted at the top of the Figure 1.2, it is focused on the confidentiality and reliability of interactions between entities. Once the trusted discovery process has been accomplished, the authenticity of each single communication has to be guaranteed as well. This aspect of digital trust can be regulated by machine to machine protocols and has to be provided both for online and for offline scenarios.

In a network of the scale of the IoT, digital trust can be very hard and expensive to achieve but, for widespread adoption of the ever-expanding IoT such a property is of paramount importance. So far, *trusted discovery* and *trusted interactions* have been approached with centralized systems involving trusted third parties, authorities or governments. However all of them have failed as they were often found to collect and misuse private information. Furthermore, communication and authentication protocols proposed so far, have frequently showed not to be enough secure to collect and manage sensitive data by themselves or to interact with other devices/systems managing them. Information security within the IoT covers several layers of abstraction ranging from physical protection to the hardening of digital computation and communication protocols and, while trusted third parties were found to collect and misuse confidential data, systems not providing security protocols were found to open point of access to private networks or to leave backdoors for stolen data exfiltration.

As a toy example, smart TVs are nowadays manufactured with general purpose processors that enable consumers to surf the Internet, make online purchases, share pictures and as many other tasks as provided by laptops or desktops. However, unlike personal computers, TV operating systems are usually easier and less powerful thus adopting weak or outdated security countermeasures. Therefore, sensitive information such as online payments, private contacts or personal data can be targeted by frauds and consumers can be the victim of identity theft attacks [20]. Moreover, if such smart devices are not the final victims, they might be exploited as entry points within private networks as they are usually interconnected to other devices in order to provide digital services. Staying on the example of smart TVs, they

are usually connected to our network attached storage (for short, NAS) and even our personal computer to enable media streaming without the need of physical drives. As such they can be infected to steal NAS content or to create botnets. These potential risks are exacerbated by the fact that making the IoT secure is more challenging than making private or cloud networks secure. The reasons are multifold:

- **Simple Devices**: small devices such as the ones populating the sensory swarm might not have enough computational power to establish secure communications;
- **Untrusted Manufacture**: Most design houses are usually fabless and tend to manufacture their designs in offshore facilities. Moreover, designers tend to embed third party intellectual properties (for short, IPs) in their designs thus raising security concerns about the trustworthiness of both software and hardware elements [21];
- **Upgrade on the field**: if a vulnerability is discovered that affect the security of a device, to patch that device might not be always possible and feasible thus leaving costumers with unsupported or vulnerable devices.

Compared to standard operating systems or networks, the IoT ecosystem pushes cyber attacks in adapting and in changing their nature in order to be effective and efficient. These new attacks can be roughly grouped in:

- **Capture**: attacks based on the capture of devices can have two different forms depending on the nature of the target, whether hardware or software. Hardware target can be captured to gain positional advantage during the current or further attacks. This kind of attack is well suited for IoT devices due to their broad distribution as well as their continuous mobility that creates a perfect target for attackers able to accomplish capture attacks. Attackers are then able to reach devices and to capture them to alter the behavior of the network or to re-introduce them later on in the network. The other form of capture attack is targeted at the information and is usually based on the dump of sensitive or secret data. This attack is aimed at stealing information about the victim (if it is not possible to capture the device) or to steal information collected by the target device but referred to other devices. This attack is common in the IoT where devices are

usually part of a peer to peer network and collect or infer data from the surrounding environment;

- **Disrupt**: destroy, degrade, deny, and disrupt attacks, here are all grouped under the disrupt name. These attacks are different from capture which final goal is to improve the knowledge, thus the power, of the attacker compared to the victim. On the contrary, disrupt attacks are only interested in lowering down the power of the victim. As for capture, also the disrupt attack can be targeted at the hardware or information level. However, in this case, disrupt of information is not that easy as nowadays sensitive data are distributed and decentralized. As such, the correct and complete disrupt of some information might result in capturing and destroying all its replica all around the world;

- **Manipulate**: unlike capture or disrupt attacks, manipulation is aimed at influencing the target's decision cycle. The easiest manipulation is about compromising input data used by the target. Changing the input might cause the victim to take different decisions and this change in the computation can be driven externally by the attacker. Information manipulation can also target embedded data, whether by physically replacing or by software injection. Further into the decision cycle, an attacker may directly manipulate sensors. Unlike feeding the sensor with fake information as we have already described for the input manipulation, an attacker could exploit a compromised sensor that communicates with the target. In this case, a successful attack will again achieve manipulation of the input but rather than change it itself it will force another device within the network in providing the fake information.

It is then clear that mobility and distribution in the IoT increase devices' attack vectors as they make it easier to manipulate entities without fear of detection. As such, it is of paramount importance to design new security mechanisms that do not rely on the data secrecy but on distributed and decentralized protocols as well as on the cooperation between devices in order to build a self-enforced ecosystem. This secure and trustworthiness environment can be designed by addressing the concept of device identity and authenticity. On one hand, solving the problem of device identity means being able to provide unique identifiers to digital entities such that they can be discovered and reached before any information is exchanged. On the other hand, identity is not enough to create a secure digital ecosystem. Once a

device or any other digital entity has been identified, we have to be able to verify it and to send information to it in a secure a reliable way. To do this, we need robust authentication protocols online and offline.

Dynamic Identities

The concept of smart devices is not new and does not belong to the IoT. People use smartphones daily for their personal and business life and their use has exceeded the simple communicative purpose replacing it with a key to the virtual world. These devices are intrinsically linked to the concept of identity. The first and easier identity tied to a smartphone is the identity of the owner. Other identity elements can be retrieved from a smartphone as the International Mobile Station Equipment Identity (for short, IMEI) number, the subscriber identity module (for short, SIM) as well as all the identities attributes that are associated with the many apps that are installed. The smartphone is the most easy example to understand but other devices will soon need as well to be identified and reached such as vehicles, traffic cameras and any other thing that is connected to some sort of network and has enough computational power. These devices will be trillions in number soon and will have enough computational and battery capabilities to manage long lasting services thus being required to be identifiable.

To better understand this identification need, it is important first to define the general concept of identity that, in the Cambridge dictionary is defined as *"who a person is, or the qualities of a person or group that make them different from others"*. In the digital world these characteristic can be expressed by attributes such as names, email addresses and alphanumeric identifiers. Each single attribute can be local or global depending on the context. In fact, taking as a toy example a physical street address, the number is not global. There might be hundreds of thousands of other streets with the same number but, if we pair that street number with the street name, the city name and maybe also the state name what we get is a global and unique identity of a specific house (see Figure 1.3).

The example depicted in Figure 1.3 shows that we have always been used to the concept of object identity as a matter of been able to reach something and this is going to be even more crucial in the IoT. In fact, things will need to find each other and to communicate to each other seamlessly and autonomously without any human interaction. As such, it is important to highlight and to understand that while the identity factor by itself does not make a device smart, a smart device without an identity is useless. In

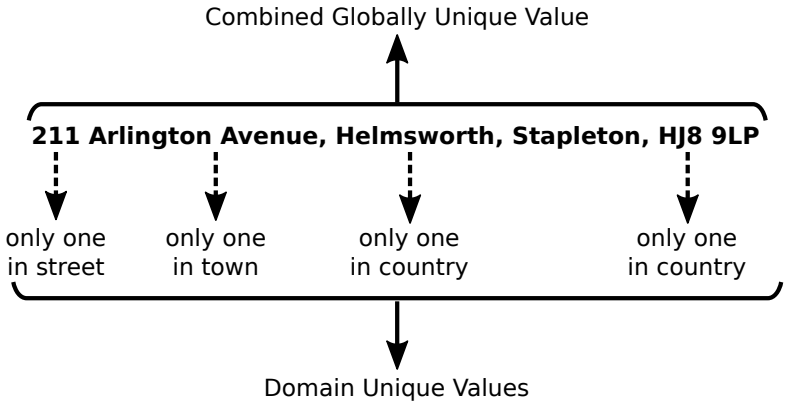

Figure 1.3: Physical example of globally unique addressing

fact, such a device will be unable to introduce itself to other devices and to communicate with them.

The concept of identity is strictly tied to the concepts of uniqueness and scalability. As already shown in Figure 1.3, whilst local attributes might be in some cases re-used across different contexts such as street numbers within the same country, global attributes should not. As such, while local identities might be shared or changed, global identities should never be misused. Identity uniqueness is a contentious concept and examples in the past such as Yahoo announcing to re-use previously disabled email addresses showed that this should be not underestimated [22]. Scalability is another paramount factor as due to the IPv4 running out of addresses we have already witness how static identifiers are not scalable and should not be applied in the IoT [23].

The past few years have witnessed the growth of the IoT and the shift from its theoretical-only analysis to a practical implementation [24; 25; 26]. The IoT, as a giant-sized network infrastructure, is characterized by its large-scale heterogeneous network elements and a high dynamic behavior that have raised attention from both the academic and industrial fields on both security and privacy [27; 28]. Different approaches and solutions have been proposed so far [29; 30; 31] but usually they require either high computational capabilities or the involvement of trusted third parties or even the designation of powerful things to act as gateways within the IoT. To solve this problem and create a truly democratic and distributed environment, both the academia and the industry started to work on IoT environments based upon the blockchain technology. The blockchain is a decentralized

and distributed database able to collect any message sent between things and any operation accomplished by things. As such, it can be seen as an abstract trusted third party even though such third party is the IoT network itself.

Albeit this approach allows resource constrained devices to participate and to claim their identity within the IoT, it assumes all the things to be able to see each other and to communicate with each other. However, specially when things join an unknown and untrusted network, to establish a communication requires a discovery procedure based on a name scheme. Device discovery is a design principle which focus on the capability to find devices within a network by attaching to them some meta-data that can be interpreted and analyzed by other devices within the network. By making devices easily discoverable also their interoperability is increased. In fact, a device that cannot be reached by other devices cannot interact with them and thus, specially in an interconnected world such as the IoT, it is useless.

Current ongoing studies are focused on the definition of a new name standard but the definition of new standards may force manufacturers in changing their own solutions thus hindering their adoption. Furthermore, static name syntaxes might not scale in the future as already proved by IPv4 [32].

Online Authenticity

As already introduces at the beginning of Chapter 1.2, one of the main concerns about online authenticity is about trust. For more than four decades, such problem was solved by involving in the authentication process a trust anchor or trusted third party. This approach is still widely used nowadays and is composed by three players:

- **Claimant**: this is the user or device demanding access to some kind of service. The final goal of the claimant is to provide one or more authentication factors aimed at proving its digital identity to the verifier;
- **Verifier**: this is usually an automated system filtering accesses to some kind of service and capable of verifying the digital identity of each claimant;
- **Trust Anchor**: this is an external entity that provides to both the claimant and the verifier one or more authentication factors.

The above model is at the basement of any public key infrastructure and is widely adopted also in the IoT where tiny, low-cost and low-power devices can now be connected to each other. However, these tiny things can have important limitations both in the storage capability and in the computational and communication power. To solve this challenge, usually IoT dumb devices relies on bigger and more powerful things acting as the trust anchor introduced above and usually called super-nodes. These bigger devices (such as laptops or smartphones) sit in the middle of communications thus allowing tiny devices to speak to each other in a secure and private way. However, unlike computer-based services that use to be remotely located in the Cloud ecosystem, IoT devices are pervasive thus being easily captured, disrupted or manipulated. More in detail, the IoT showed to be vulnerable to sybil attacks [33] where attackers can manipulate fake identities or abuse pseudo-identities to compromise the effectiveness of the IoT and even disseminate spam. To solve the above limitations, new authentication techniques were investigated based on the graph theory and aimed at exploiting the interactions between devices within the IoT. Albeit such approaches proved to be promising for mobile sensor networks (for short, MSNs), they showed some limitations for the IoT due to the lack of historical behaviors and the unpredictable and highly dynamic behavior of involved smart things.

Offline Authenticity

Approaches based on remote trusted third parties or on the cooperation and interaction among devices have proved to succeed in the majority of the use cases as in the IoT any device is connected to the network at any time. However, special use cases such as deserts, flights or any other offline scenario require even tiny devices to care for their own security and privacy. It is then of paramount importance to develop also offline authentication protocols that can be exploited even by power constrained devices in a machine to machine scenario where no trusted third parties or gateways can interact such as the ones listed next:

- **Network Disconnections**: albeit IoT will have pervasive network coverage, there still might be the case for temporary offline scenarios due to maintenance, attacks or simply the lack of coverage;
- **Data Privacy**: in the IoT, specially for those solution based on distributed and decentralized technologies, each message exchanged between devices is shared to the whole network. While this approach

Topic	State Of The Art	Open Challenges
Name & Discovery	-Domain Name Systems -Public Key Infrastructures -Blockchain Ledgers	-Dynamic Names -Context-Aware Identifiers -Private Environments
Online Authentication	-Trusted Third Parties -Fingerprinting -Public Key Infrastructures	-Self-Enforced Protocols -Software-based Uniqueness -Tunable Authentication
Offline Authentication	-Hardware Intrinsic Security -Public Key Infrastructures	-Disposable Tokens -Fully Offline Availability

Table 1.1: Thesis motivation summary

enables more secure interactions, there are still use cases where it can threaten the privacy of messages being exchanged. As a toy example, digital medical prescriptions would need high availability and privacy to be widely adopted. Thus, offline protocols capable to accept medical prescriptions even in the absence of network coverage as well as anonymous protocols capable to protect sensitive patients' data are of paramount importance.

To solve the challenges of authentication in such offline scenarios, mobile devices started to carry hardware elements that can be used for authentication purposes such as SIM cards, mobile trusted platform modules and the latest physical unclonable functions. This new hardware-based approach brought advanced properties, such as tamper evidence and resiliency against software emulation attacks. However, all these approaches showed to be not enough in some special use cases such as event-limited offline authentication processes. Such scenarios range from offline mobile payments, were double spending and money forgery has to be avoided, to e-health offline applications where, as an example, only a limited number of medical prescriptions have to be granted to patients. For all these applications, standard offline solutions, even when based on secure hardware elements, showed to be unable to provide disposable login procedures thus demanding for better approaches.

Table 1.1 provides a summary of the topics we have focused on during this thesis and, for each topic, which open challenges we have investigated.

1.3 Contributions

The main goal of this thesis is the analysis of the concept of identity and authenticity in the digital world both from the online and the offline perspective. As a result, four different solutions have been designed aimed at providing (i) a new standard for naming and discovering of smart things within the IoT, (ii) a self-enforced online authentication system based on the network context and an improved version of the online authentication based on device behavior and (iii) two authentication approaches for offline scenarios. As shown in Figure 1.4, these proposed solution all together serve for the transition from the Internet of untrusted things to the Internet of trust.

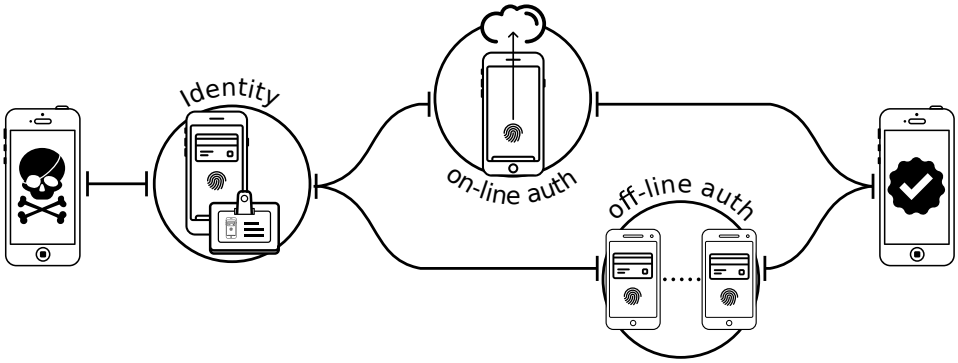

Figure 1.4: Roadmap to the Internet of Trust

Name and Discovery in the IoT

As depicted in Figure 1.4, the first step towards an Internet of Trust is about defining reliable digital identities. This is a key requirement to provide authenticity within the IoT as an authentication process is usually composed by a claimant that provides some identity information and a verifier that validates them. Current protocols such as Domain Name Service (for short, DNS) are not designed to scale into the IoT [32] and thus, to solve this challenging scenario, we have proposed a CONtextual NamE disCovery and resolving with Transactional security (for short, CONNECT) solution, submitted to the *Computer Communications journal, Special Issue on the Internet of Things: Research challenges and Solutions* and actually under review.

CONNECT, is the first solution that solves the name and discovery open

issues by exploiting implicit information such as attributes and relationships within the IoT. As such, CONNECT is able to build a dynamic but yet unique ID that is used by things to discover and to reach any other thing in the network. Details about this solution are provided in Chapter 3.2. We propose a new approach named *proof of knowledge* in order to build such a trust-continuity factor. In this approach, each thing shows its identity within the network by proving its knowledge of the network past activity. This new approach not only removes ;

Online Authentication

As already introduced in Chapter 1.2, one of the major security concern for IoT devices is their vulnerability against physical attacks. As such, authentication solutions can no longer be based neither on data secrecy nor on public key infrastructures but has to be based on decentralized and self-enforced systems. In fact, secrets such as private keys for cryptographic schemes might get stolen and trusted third parties for public key infrastructures might turn malicious thus leaking sensitive information.

In this thesis two novel authentication solutions are then proposed. The first one is part of CONNECT and it is based on the blockchain technology. This approach exploits the past network history as a fix for present interactions with a distributed and democratic protocol. The result is what we called *trust continuity factor* and is based on the interaction among smart things. Our approach, removes the need for the classic blockchain mining procedure while providing a self-enforced and privacy-aware solution. However, this solution showed to be not enough within unattended networks. In fact, for all those scenarios where attackers can easily capture, destroy and manipulate things, *Sybil Attacks* [33] could be exploited to thwart CONNECT's trust continuity factor. In the presence of Sybil attacks, an IoT system may generate wrong reports, and users might receive spam and lose their privacy as shown in a report from 2012 [34] were a substantial amount of online accounts (72% of Facebook accounts) were confirmed as fake or sybil. To solve the problem of sybil things in the IoT we have then designed an additional online authentication solution based on Software-based Unclonable Functions (for short, SUFs) that has been submitted to the *International Conference on Information Systems Security and Privacy* and is actually under review.

SUF takes inspiration from both hardware intrinsic security solutions and device fingerprinting techniques to build a more secure and reliable authen-

tication scheme in the cloud. The core element of SUF is the exploitation of the dynamism derived by the ever-changing status of application meta-data into a security factor thus allowing a better and stronger authentication and identification approach. Unlike CONNECT, this new solution does not build an identity of the whole device but rather can work both on a service and application level. SUF represents a solid and effective authentication mechanism that for the first time, to the best of our knowledge, allows unclonability properties to be fully leveraged without requiring any special hardware.

Offline Disposable Authentication

So far we have introduced two solutions focused on IoT authentication protocols. Both of them exploit the highly interactive and interconnected IoT ecosystem in order to bootstrap unique and unclonable digital identities. This approach works in the majority of the IoT use cases where device are connected to the Internet at any time. However, even within the IoT we can still experience temporary disconnections mostly due to (i) maintenance routines, (ii) network unavailability or (iii) denial of service attacks. In all these scenarios, approaches based on online trusted third parties will not work thus requiring different solutions. Public key infrastructure scenarios where each thing is shipped with a trusted and signed public and private key can be used. Furthermore, hardware-based solutions resilient to physical attacks have already been proposed as a secure and reliable way to protect private keys such as physical unclonable functions or trusted platform module.

To fill the lack of disposable offline authentication approaches we have designed two solutions. Both are based on the advanced technology of hardware intrinsic security and both are able to prevent misuse and re-use of authentication tokens in a complete offline scenario. With the first solution we have designed Fully Off-line secuRe CrEdits for mobile micro payments (for short, FORCE) while with the second solution we improved FORCE thus been able to design a Fraud Resilient Device for Off-line micro-payments (for short, FRoDO). FORCE has been published and presented at the *13th International Joint Conference on e-Business and Telecommunications* while FRoDO has been published in the *Transactions on Dependable and Secure Computing*.

On the one side, the main novelty of FORCE has been to provide a mobile authentication approach where all involved parties can be fully offline and

Topic	Proposed Solution	Main Contribution
Name & Discovery	CONNECT: CONtextual NamE disCovery and resolving with Transactional security	-Attribute-based ID -Self-enforced discovery -Context-aware authentication
Online Authentication	SUF: Software-based Unclonable Functions	-Per-application PUF -No additional HW -Scalable and Upgradeable
Offline Authentication	FORCE: Fully Off-line secuRe CrEdits for mobile micro payments	-Fully-offline -Event-based (disposable)
	FRoDO: Fraud Resilient Device for Off-line micro-payments	-Memoryless -Resilient to data breach

Table 1.2: Thesis contribution summary

relying solely on local data to perform the requested operations. On the other side, FRoDO has been proposed as an improvement over FORCE showing additional resiliency features against data breach attacks. It is of paramount importance to note that both these two solutions have been published focusing on offline payment as, compared to e-health and to other offline disposable authentication use cases, mobile payment is the one that is more mature and that is more suffering from the lack of a secure and reliable fully offline authentication approach. However, as described later in Chapter 5, the architecture and the protocol provided in both of them can be used for general authentication purposes.

Offline disposable authentication protocols, focused on the specific application of mobile micro-payments, has shown potential in solving the problem of anonymous and democratic payment processes. This new kind of decentralized transactions have drawn attention with the birth of cryptographic digital currencies such as *Bitcoins*, and different companies started to produce real devices or applications to bring this new payment model to the people. However, as for Bitcoins, it came out that a lot of solutions were able to provide secure and anonymous online transactions whilst there were none for offline disposable transactions. The solution proposed in this thesis for offline disposable micro payments has also made possible the realization of a Spanish patent that is currently under review.

Table 1.2 provides a summary of the main contributions provided in this

thesis as well as the name of each solution that has been presented here aimed at solving the state of the art open challenges.

1.4 Awards

During my studies I was able to participate in an award winning cybersecurity startup - Excalibur, and to cooperate with the European Institute of Innovation & Technology (EIT ICT) and Deutsche Telekom Labs (T-Labs) where thanks also to the collaboration with the Kantara Identity of Things Discussion Group I was involved in the activities for the realization of an overarching identity framework for the Internet of Things. I have personally worked with the CEO/CTO of Excalibur and together we designed a solution that has been submitted to the *Cisco Grand Security Challenge 2014* where it won the 1st place as the best security solution aimed at hardening the Internet of Things and the next Internet of Everything. The solution proposed for the Cisco Challenge has then been further improved with the collaboration of the Deutsche Telekom Labs and Alcatel-Lucent's Bell Labs and the paper is now under review by the Computer Communications journal. Furthermore, even our offline disposable authentication tokens have aroused a great interest from the academic community. In fact, our thesis proposal has been ranked in the top layer at the *Security CyberCamp 2014*, the first Spanish forum aimed at identifying and promoting talent and innovation in the up and coming IT security industry by bringing all together some of the leading national and international experts in cybersecurity.

1.5 Organization of the thesis

This thesis is organized in four parts. Part I introduces the problem of *trust* in the Internet of Things and gives an overview of motivations and contributions as well as an introduction on all the technologies and standard protocols referenced throughout this thesis. Part II introduces the problem of digital identities and describes our attribute-based approach. Part III introduces the problem of authentication in the Internet of Things and is composed by: Chapter 4 analyzes authentication protocols in which devices have access to the network whilst Chapter 5 analyzes offline authentication protocols and, last but not least, Chapter 6 proposes a real world application that can benefit from our disposable fully offline authentication system. Part IV is the last one and contains a wide spectrum analysis on the work that has been done in this thesis and highlight what remains open as future work.

Chapter 2

Building Blocks

2.1 Public-Key Cryptography

Modern computing has generated a tremendous need for convenient, manageable encryption technologies. Symmetric algorithms, such as the Triple DES [35], provide efficient and powerful cryptographic solutions, especially for encrypting bulk data. However, under certain circumstances such as key exchange and trust, such symmetric algorithms [36; 37] can be inadequate.

The key exchange problem arises from the fact that communicating parties must somehow share a secret key before any secure communication can be initiated, and both parties must then ensure that the key remains secret. Of course, direct key exchange is not always feasible due to risk, inconvenience, and cost factors. In some situations, such direct key exchange is possible; however, much commercial data exchange now takes place between parties that have never previously communicated with one another, and there is no opportunity to directly exchange keys in advance. Regarding the trust problem, ensuring the integrity of received data and verifying the identity of the source of that data is very important. A symmetric key can be used to check the identity of the individual who originated a particular set of data, but this authentication scheme can encounter some problems involving trust and, last but not least, sharing the secret key over an insecure channel might threaten the whole communication.

In the 1970s Martin Hellman, Whitfield Diffie [38], and, independently, Ralph Merkle [39] had a beautiful cryptographic idea. Their idea was to solve the key exchange and trust problems of symmetric cryptography by replacing the single shared secret key with a pair of mathematically related

keys, one of which can be made publicly available and another that must be kept secret by the individual who generated the key pair.

The first advantage of such approach is that no key agreement is required in advance, since the only key that needs to be shared with the other party is a public key that can be safely shared with everyone. Second, whereas the security of a symmetric algorithm depends on two parties successfully keeping a secret key, an asymmetric algorithm requires only the party that generated it, to keep it secret. Third, the issue of trusting the other party disappears in many scenarios, since without knowledge of your secret key, that party cannot do certain malicious activities, such as digitally sign a document with your secret key or divulge your secret key to others.

To depict how the asymmetric cryptographic scheme works, we will refer to two users named Alice and Bob. At the beginning, Bob randomly generates a public/private key pair and allows everyone to access the public one, including Alice.

The contribution of the asymmetric cryptography is twofold depending on the key being used for the encryption and decryption as follow:

- **Public-key encryption**: if the message is encrypted with the sender's public key we obtain a public-key encryption scheme. The message cannot be decrypted by anyone who does not possess the matching private key, who is thus presumed to be the owner of that key. This approach is used to achieve confidentiality of the message being exchanged;
- **Digital signatures**: if the message is signed with the sender's private key then it is possible to exploit the asymmetric encryption to obtain digital signatures. Such signatures can be verified by anyone who has access to the sender's public key and prove that the sender had access to the private key. Furthermore, this also ensures that the message has not been tampered, as any manipulation of the message will result in changes to the digest, which otherwise remains unchanged between the sender and receiver.

Digital Certificates

Digital certificates are data structures able to certify their creators, i.e. the private key owners. In authentication protocols they serve the same purpose of passports or driving licenses for people as they represent a link between a

real identity (i.e. a device, a service or a person) and a digital identity (i.e. an account). Digital certificate basic concept is that the mapping between real and digital identities should not be questioned by those who receive and verify the certificate. Hence, digital certificates are usually issued by certificate authorities (i.e. well known trusted third parties) but can also be create privately or by the same owner of the certificate thus taking the name of *self-signed certificates*.

The combination of standards, protocols, and software that support digital certificates is called a public key infrastructure (for short, PKI). The software that supports this infrastructure generates sets of public-private key pairs that are related to one another through mathematical algorithms. The key pairs can reside on one's computer or on hardware devices such as smart cards or floppy disks and while private key has to be kept secure by the owner, public keys are usually distributed in the Internet. To this purpose, issuers of digital certificates often maintain online repositories of public keys thus making it possible to authenticate certificates in real time without bothering their owners asking for the public key.

A digital signature protocol is usually composed by three steps as follow:

- **Key Generation Step**: that selects a random private key from a set of possible keys. This first step outputs both the private and the public key;
- **Signing Step**: in this phase, the private key and the message are given in input to the algorithm that produces in output the signature, i.e. a digest that identity the message-key pair;
- **Verification Step**: given the message and its signature (i.e. the digest computed in the signing step), the verification step is able to recompute the signature with public key and to accept the message if the received signature is the same that has been just computed or to reject it if the signatures differ.

For a digital signature scheme to be effective and reliable, two main properties are required such as (i) the feasibility in validating a signature if starting from the right public key and and (ii) the unfeasibility of generating a valid signature without the right private key. Both public-key encryption and digital signatures schemes have been adopted during this thesis to validate confidentiality, integrity and authenticity of messages being exchanged.

2.2 Blockchains

In the last few years, a new technology named *blockchain* [40] obtained a great success changing the consideration of centralized authorities. This technology can be roughly seen as a digital ledger that sits at the core of decentralized ecosystems and keeps track of any changes by holding a new record for each transaction. Transactions contain network updates and are generated and verified by network peers with a decentralized and distributed consensus algorithm thus providing a self-enforced environment [41].

Blockchains first emerged with Bitcoin [42], a distributed cryptographic currency payment protocol unleashed in 2009 by an anonymous author who signed itself as Satoshi Nakamoto. As explained by Satoshi, Bitcoins were designed as *a purely peer-to-peer version of electronic cash* that would finally allow people to send payment directly from the payer to the payee without any intermediate trusted third party as we have always did in the past with banks or payment gateways.

This direct payment process is provided by a public key infrastructure (PKI) cryptographic protocol that protects each transaction while ensuring confidentiality and integrity. In fact, each time a message is sent over a blockchain, the sender is first required to sign it in order to make other peers able to verify its authenticity and integrity. Each of these signed messages are then collected within a distributed database called blockchain that can be accessed and browsed by anyone. As the blockchain is usually exploited for cryptographic currencies carrying money from one peer to another one, messages are usually called transactions.

The innovation originally provided by the blockchain is the idea of combining a decentralized consensus protocol with a block-based organization of transactions. In fact, the usage of asymmetric cryptography that allows to verify the consistency of each transaction is not enough. As transactions are verified on top of the transaction history, it is also important to provide consistency of the order in which transactions have been validated in the past thus avoiding transaction ambiguity.

The blockchain is an ordered and back-linked list of blocks carrying transactions which encode exchanged information between two or more participants. Each block consists of a collection of unverified/unspent transactions and is linked to the previous block in the blockchain thus creating a chronological order of blocks that all together build a chronological order of transactions. Usually cryptographic transactions only describe money movements. How-

ever, more abstract and general purpose transactions between participants can be used to represent the concept of a *state* or *snapshot* of the whole network. In fact, each blockchain-based system can be seen as a *state transition system* where there is an initial state and a transition function that produces in output a new state. Such a state transition function is defined as the current set of *unspent transaction outputs* (for short, UTXO) that, as depicted in Figure 2.1, represents all the possible outputs that the network is still able to process. Each different set of spent/unspent transactions thus create a state that represents a snapshot of the system. However, in a system where a high rate of transactions is generated, peers within the same network have to cooperate in order to agree on the same snapshot. This process, called state validation, would be easy to accomplish with access to a centralized and trusted authority which takes care of transactions' order over time (as banks are doing nowadays). However, in a decentralized environment there is not any trusted third party and all transactions have to be verified with a distributed consensus algorithm in order to ensure that everyone agrees on their order. Systems based on the blockchain technology usually collect transactions into *blocks* and requires nodes to create and verify such blocks on a timely base.

As depicted in Figure 2.2, each block is identified by a hash of its header section and contains information which help in verify if the block is correct. The main important elements within a block are the following:

- **Previous Block**: a link to the previous block in the blockchain;
- **Merkle Root**: the root of the tree used for the transactions collected in the block [43];
- **Transaction List**: a list of all transactions collected in the block.

The key role of the blockchain is to keep track of blocks that, in the paradigm of a state transition system, have to be validated as follows:

1. Check if the previous block referenced by the block exists and is valid;
2. Check that the time-stamp of the block is greater than the previous block time-stamp;
3. Check the correctness of the block;

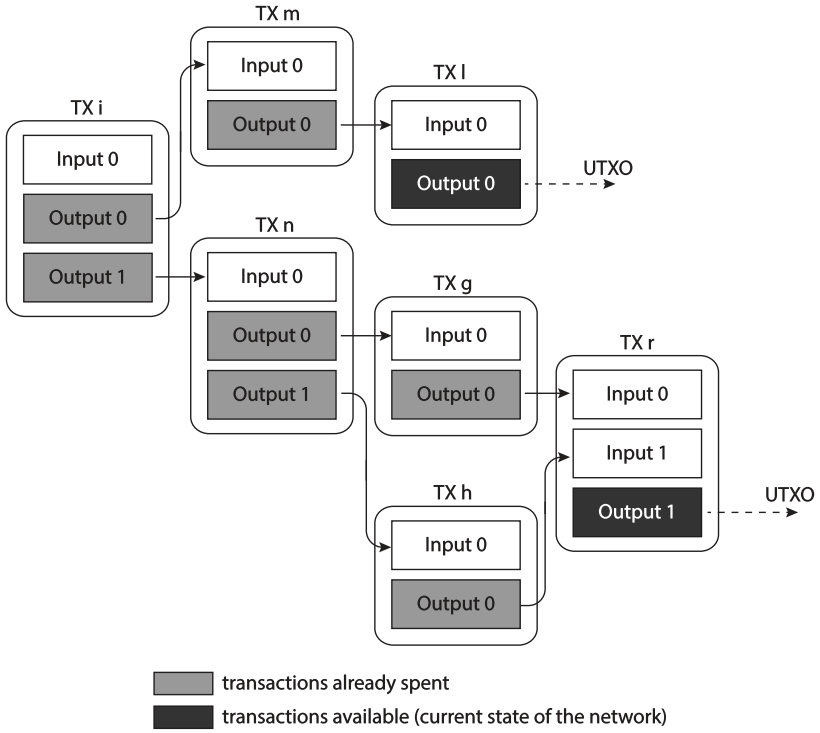

Figure 2.1: Blockchain state

4. Suppose that S_0 is the state at the end of the previous block validation and that TX is the transaction list within the block being validated now, then for any i^{th} transaction in the list set:

$$S_{i+1} = APPLY(S_i, TX_i) \tag{2.1}$$

5. Return TRUE if and only if none of the transition function iterations returns an error and register the new state S_n as the state at the end of this block;

6. Check the correctness of the new state;

Essentially, each transaction in the block must provide a valid state transition. However, the concept of *state* is something abstract and not directly embedded within the block. Hence, the only way to build the state of a system is to follow the blockchain from the beginning of the time up to the current block. This explains why, as depicted in Figure 2.3, each block must

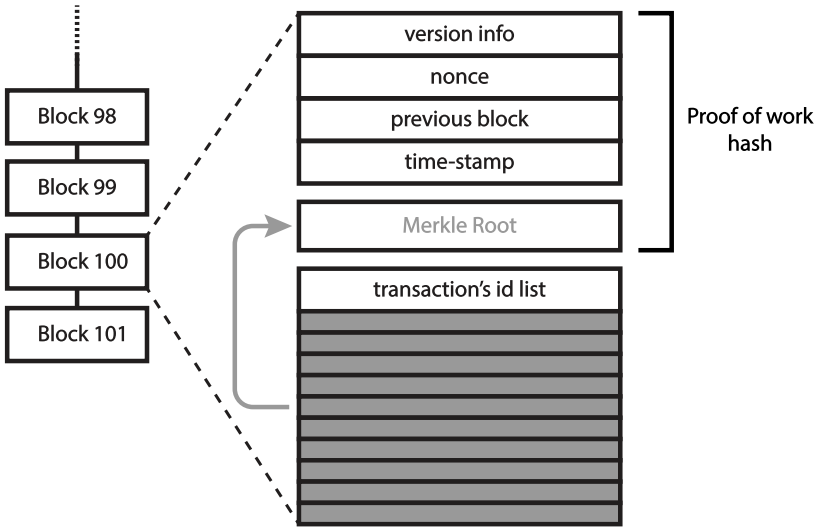

Figure 2.2: Simplified block structure

provide both a valid link to the previous block and a valid link to the next block in the blockchain (this latter only if the block is not the last one).

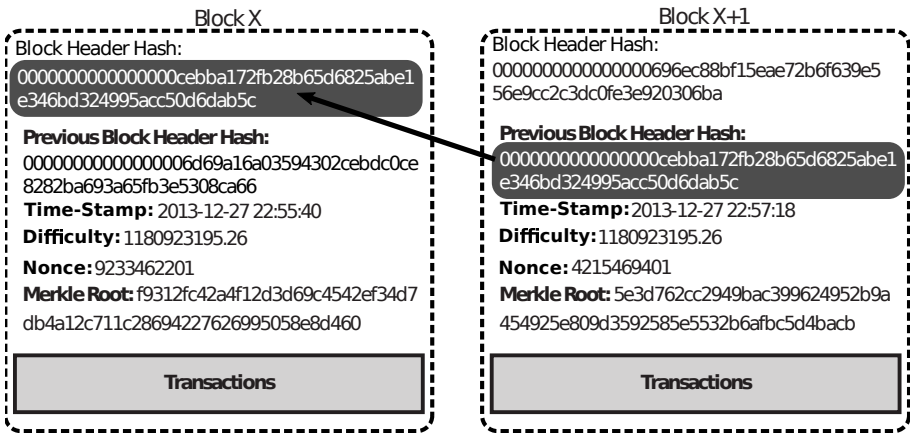

Figure 2.3: Block structure within a blockchain

Blockchain Mining

Virtually any peer can collect transactions and create blocks. However, this creates a competition among peers which might propose different blocks linked in different orders thus resulting in different blockchains. Blocks'

order, as for transactions order, has to be unique to guarantee transaction consistency. Basically, this goal is achieved by slowing down the speed at which peers can build new blocks. Hence, each peer willing to propose a new block has to solve a computation and to provide the solution of this computation as the proof of its work. This approach is called Proof Of Work (for short, PoW) [44] and its difficulty is dynamically adjusted to minimize the chances of two or more blocks being able to propose a new block at the same time.

The peers that usually work to build a PoW are called miners and, as their computations are usually involved in verifying transactions for other peers, they are rewarded for their work. This creates a competition among miners willing to achieves the maximum reward that reflects into a faster transaction validation. Furthermore, miners are responsible to maintain a chronological order of blocks. Each block is then built and added to the chain (as shown in Figure 2.3) only after the PoW has been solved and validated by other peers. In cryptographic currencies, the PoW is based on finding a value named *nonce* that, when combined with the header of the previous block, results in a hash with a specified number of leading zero bits. The number of leading zero bits forces the resulting hash output to be smaller than a value z thus defining the PoW difficulty level.

Since cryptographic hashes are one way functions, the only way to solve the PoW, and to find those results with the given leading zero bits, is to try all the possible nonce combinations. This approach is called brute-force and avoid powerful peers from solving it in a shorter time than standard peers. However, once a nonce is found by a peer, such a value is written within the new block and sent to the network. In this way, peers receiving the new block can easily and immediately verify that the new block is a valid one and use it as the new head of the chain. This approach makes solving the PoW hard whilst verifying the PoW extremely fast thus guaranteeing that once a new valid block is found and sent over the network, other peers can quickly validate it and agreed upon it before another valid block is generated. Furthermore, miners are usually required to mine on the longest blockchain, i.e. the blockchain with the longest path from the beginning of the time (the first block) to the actual time (the blockchain head) in order to avoid malicious or random blockchain forks.

Blockchain Forks

As already introduced in the previous section, the big difference in speed between creation and validation makes the blockchain able to quickly discard parallel forks and to agree on one single chain. However it is still possible to have rare cases of multiple parallel chains. In fact, even taking into account the latency of the network, peers may still receive different blocks, each one proposed as the new blockchain head, and each one from a distinct neighbor. An easy solution to solve this problem is for peers to accept the first block received based on a chronological order. However, this approach creates a new issue that has to be addressed as well, blockchain forks.

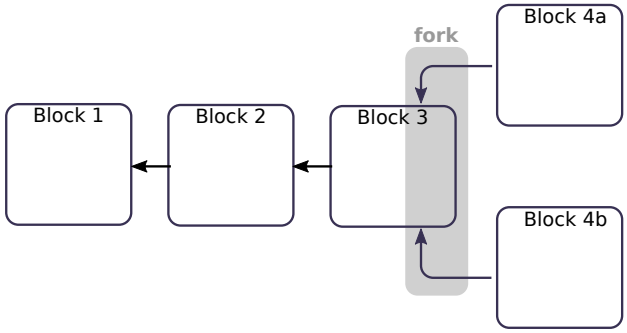

Figure 2.4: Blockchain fork example

As shown in the Figure 2.4 different miners who receive blocks in different orders might create a fork in the blockchain. Forks are extremely dangerous as they create different transaction histories hindering the protocol consistency. In fact, peers working on distinct forks might have different transaction flows over time thus being unable to agree with each other. As a toy example, if the blockchain is used for a cryptographic currency, peers might have different amount of money within the same wallet.

To solve the race condition on the blockchain fork, the longest chain rule is adopted. When a peer receives a new block proposed as the head of the blockchain, no matter if this block belongs to the current mined chain or not, if the height of the new head is higher than the head actually known, the new chain is downloaded and the old one is discarded. This scenario is depicted in Figure 2.5 where starting from $block3$ a fork is generated by mining at the same time the blocks $block4a$ and $block4b$ as new blockchain heads. These two blocks are used in the mining process by different peers and by the time tx two blockchains show to be concurrent in the same

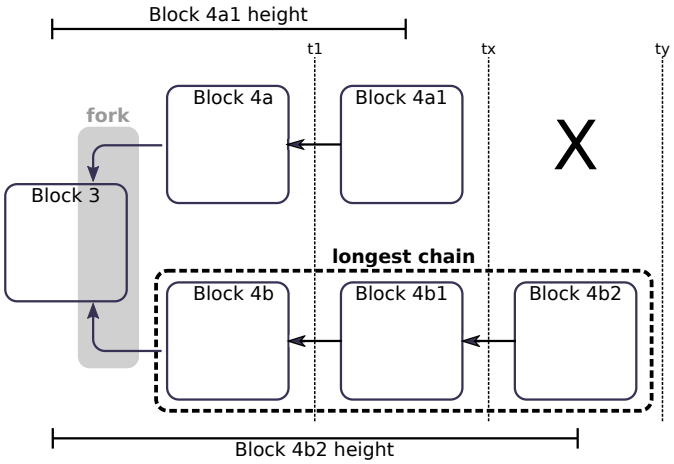

Figure 2.5: Blockchain adoption of the longest chain

network, the one headed by the block *block*4*a*1 and the other one headed by the block *blockab*1. Taking into account both the latency of the network and the increasing difficulty in solving the PoW, this parallel process cannot last forever. At some point, *ty* within the Figure 2.5, a new block *block*4*b*2 will be mined and sent to the network for the validation. At this point, two possible use cases arise:

- If the peer that receives the block *block*4*b*2 is already mining on the chain started from *block*4*b* than such a peer just needs to validate it and to add it at the head of the chain;
- If the peer is mining on another chain but the block received is valid and is longer, then the current chain is discarded and the chain leaded by the new block is used.

The only way to thwart such an approach is for the attacker to build a chain that is longer than the actual one. In such a way, any other peer that receives the malicious chain, due to the longer chain rule, will discard the real chain and adopt the malicious one. This attack is powerful as enables the attacker to change the transaction orders of the past. Staying on the toy example of cryptographic currencies, this means that an attacker might be able to make an online purchase and then remove the spent transaction from the chain. However, it is important to remember that the PoW is made by guessing all the possible hashes and that not only the attacker but

all the other peers in the world compete in finding new blocks. As such, in order for the attacker to succeed in building a longest chain that can replace the current one, the computational power of the attacker has to be greater than half of the computational power of the rest of the network. This attack is indeed called *51%* attack [45] as the attacker has to control at least the 51% of the computational power of the whole network participating in the protocol.

Despite its great capability in enabling distributed payment protocols without relying on any third parties, the blockchain technology still suffers from several limitations, the most important ones are:

- **Lack of Turing completeness**: the scripting language being adopted does not support all the possible operations. As an example, it does not support loops;
- **Lack of internal states**: each transaction within a blockchain can either be *spent* or *unspent*. There is no opportunity for internal states collecting general purpose actions between two peers. This means that a blockchain can only be used for simple on-off applications.

The above limitations force the blockchain technology in being adopted only for cryptographic currencies and requires a better and more mature technology for general systems.

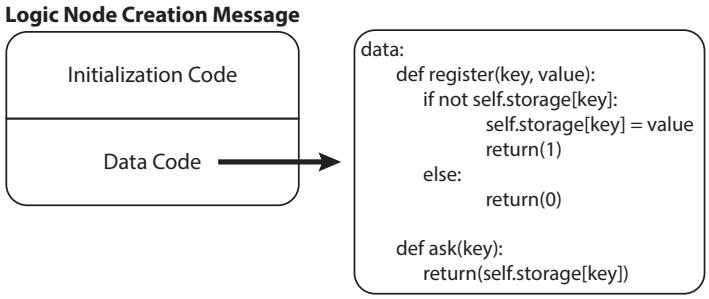

Figure 2.6: Code example of a registry name smart contract creation

Smart Contracts

In order to overcome to the blockchain limitations, an improved technology called *smart contracts* has been designed that incorporates more sophisticated forms of relationships between participants [46]. Smart contracts are

computer programs that can execute the terms of a contract (i.e. a set of instructions) thus being able to implement the *if-then-else* statement on top of the blockchain technology. Two different kinds of peers (also known as nodes) are involved in the smart contract approach and the protocol being adopted is able to manage both of them. The main responsibility of the smart contract protocol is to manage internal states of the network while providing interactions between nodes and the execution of arbitrary code. Each smart contract can be seen as a distributed and decentralized virtual machine [47] composed by two different node types:

- **Logic Nodes**: represent virtual entities, such as cloud services, that have their own execution code and are triggered by other nodes. Logic nodes are usually *passive* as they can only be activated by physical nodes;
- **Physical Nodes:** represent real things, such as devices, able to communicate with each other or to interact with logic nodes.

Physical things within the IoT can actively interact with logic nodes by either create or trigger them. If a message is sent to an existing logic node, this message activates it and make it automatically run its own code. This code has the ability to read/write to the logic node internal storage, to read the content of the received message and also to send messages to other nodes in the network. As such, the activation of logic nodes can be chained in order to accomplish simple interactions as well as complex tasks.

As depicted in Figure 2.6, the message used by physical nodes in order to create logic nodes (i.e. smart contracts) is composed by two sections. On one hand, the initialization code contains all the information related to the node setup (i.e. the sender, a digital signature, a timestamp, etc). On the other hand, the data code section contains all the instructions that have to be executed by the logic when triggered. The example depicted in Figure 2.6, shows how a smart contract can be used to register IP addresses or generic names in a key-value mapping database.

It is important to highlight that logic nodes are not executed within designated and prearrange physical nodes. Their code is embedded into the blockchain and gets executed at run-time, during the mining process, whenever a logic node gets triggered. In fact, as shown in Figure 2.6, the message used for the creation of a logic node is sent to the network thus making all

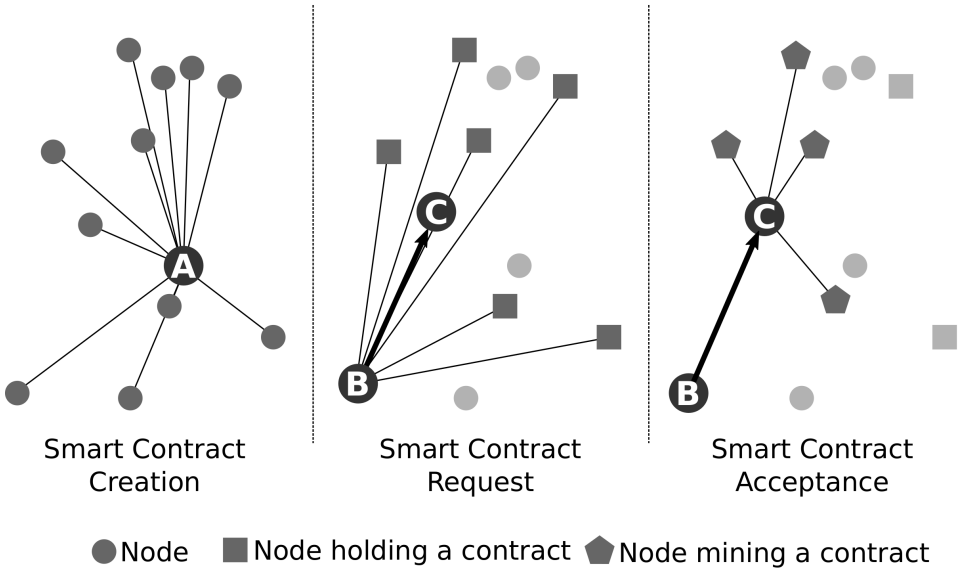

Figure 2.7: Smart contract creation and verification

nodes aware of the new smart contract. This message and the code embedded in it are stored within the blockchain and are publicly accessible. Each time a transaction is sent to a logic node, the message is heard by other nodes that start working on it as miners (i.e. re-executing the smart contract) thus being able to check if the output produced is correct or not. Hence, the process of executing smart contracts' code is part of the definition of the state transition function which is part of the block validation algorithm. This means that, if a transaction is added to the block B, the code spawned by that transaction gets executed by all nodes, now and in the future, that mine the block B of the blockchain.

A toy example of the usage of smart contracts is depicted in Figure 2.7. In the *creation step* the node A creates a new smart contract and sends it to the network. The message is then received by all the other nodes that decide whether to keep the contract (i.e. hold the code within their storage) or not. When, during the *request step*, a node B sends a message to a node C that involves the usage of the smart contract created by A, the message is heard by all the nodes, including those which hold the contract (depicted as red squares in Figure 2.7). Part, if not all, of the nodes holding the contract can decide to participate in the transaction thus becoming miners (depicted as green squares in the *acceptance step* in Figure 2.7). The transaction between

B and C is then finally verified if and only if the majority of miners accept it.

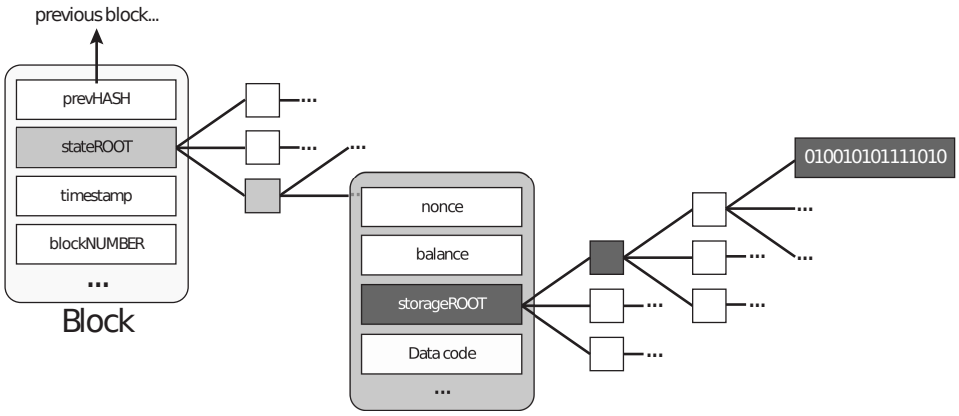

Figure 2.8: Block logic code embedded within the blockchain

In Figure 2.8 it is depicted an example of smart contracts embedded within the blockchain structure. As showed, each block contains both the code executed by a logic node as well as the storage used for the smart contract execution. This allows any node in the network to browse the blockchain back in time and to validate the execution of smart contracts thus building a trusted and self-enforced environment.

An example of the adoption of smart contracts for the IoT is sketched in Figure 2.9 where a truck (which serial number is A) willing to unload its cargo sends an unloading request to the network (i.e. all the other machines). Once the request has been collected within the blockchain, it gets executed during the *mining* process by real machines that locally execute the smart contract in their operating system. If the majority of the miners agree on the output then the transaction is validated and written within the blockchain otherwise it gets dropped and the unloading request denied. On the other side, if the mining process validate the request, the unloading information is embedded within the blockchain and from this time on, the truck A is considered *unloaded* by the whole network.

2.3 Hardware Intrinsic Security

The idea of exploiting physical randomness for authentication purposes is not new [48]. Fingerprints and other biometric approaches are indeed based

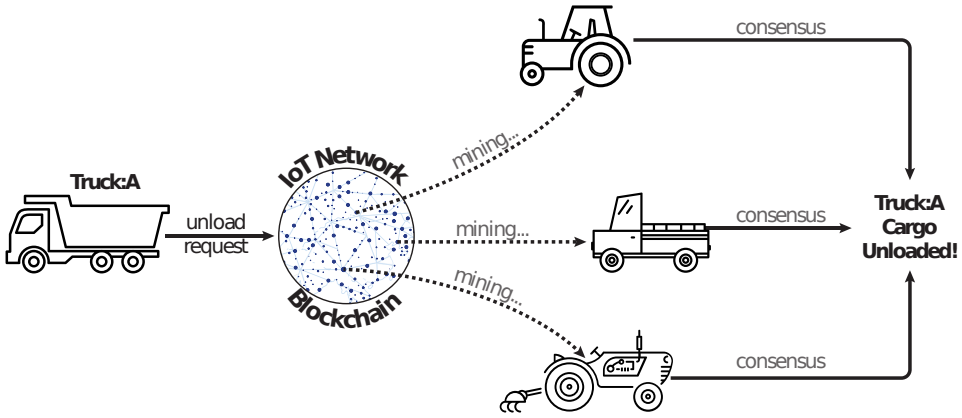

Figure 2.9: Smart contract logic overview

on the exploitation of random, but yet unique, biological characteristics in order to derive identity factors.

Back in the 1980s and 1990s such random features were already exploited and patterns embedded within papers as well as optical tokens were already used for the identification of currency notes and strategic arms [49; 50]. However, for the first standard definition and generalization we had to wait till the beginning of the 21st century when all the functions based on the unpredictability, randomness and uniqueness of physical features were first described as *physical one-way functions* [51], then as *physical random functions* [52] and finally as *physical(ly) unclonable functions* (for short, PUFs). From this latest and final definition it is clear that all those approaches perform a functional operation as they follow an input-output paradigm where each input is mapped to a unique output as with a mathematical function, even though PUF are based on the hardware.

Physical Unclonable Functions

PUFs were introduced by Ravikanth [53] in 2001. He showed that, due to manufacturing process variations, every transistor in an integrated circuit has slightly different physical characteristics that lead to measurable differences in terms of electronic properties. Since these process variations are not controllable during the manufacturing process, the physical properties of a device cannot be copied or cloned and are unpredictable even for the manufacturer. Hence, they are unique to that device and can be used for authentication and identification purposes. Implementing a PUF

requires an electronic circuit that is able to produce hardware outputs to given inputs where the input-output matching depends on PUF properties. As such, PUFs are easy to challenge but hard, if not impossible, to predict or reproduce.

PUF inputs are commonly called *challenges* whilst outputs are called *responses* and each application based on PUFs is made by two distinct steps: (a)*enrollment* and (b)*verification*. In the enrollment step (depicted in Figure 2.10) the PUF is stimulated by sending as many challenges as possible (or as needed, depending on the application) and all the challenge-response pairs are collected within a database called challenge-response database (for short, CRDB). The CRDB represents the identity of a device in the same way that fingerprints represent people identity. The enrollment step can be seen as a registration step and, due to the high unpredictability of the CRDB, it is usually executed by the same server that will later need to identify the device.

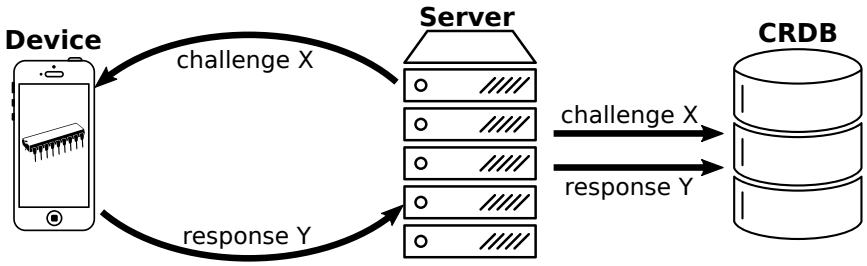

Figure 2.10: PUF enrollment step

Once the enrollment step has been accomplished and the CRDB has been computed on the server side, the verification step can be engaged between the device and the server (see Figure 2.11). During this second step, the server challenges the device with some of the inputs already used in the enrollment step and collect all the outputs produced by the device. Than, challenge-response pairs are compared to the ones stored within the CRDB and if they match then the device is authenticated.

In Figure 2.12 a comparison between fingerprints and PUFs is given. Both are used for authentication and identification purposes. PUFs, as for fingerprints, are unique per device and exploited through a challenge-response scheme. However, whilst in fingerprints the challenge consists in the scanning of the finger (i.e. it is a passive challenge process), PUFs are actively stimulated and executed to exploit the randomness in their behavior.

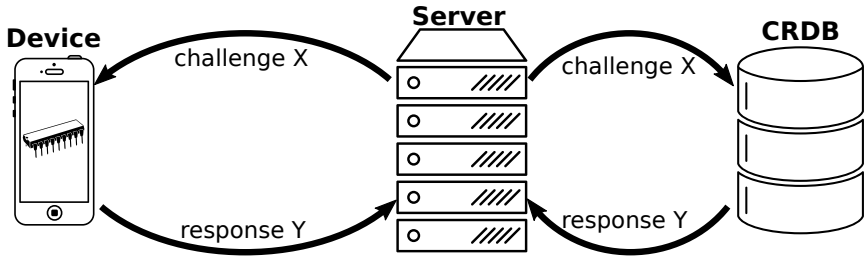

Figure 2.11: PUF verification step

PUFs main goal, as shown in Figure 2.12, is to recognize a device based on its input-output behavior. As such, the mapping between challenges and responses has to provides the following properties:

- **Inter-Distance**: represents the distance between two different responses obtained by challenging two distinct PUF with the same challenge;
- **Intra-Distance**: is the distance between two responses resulting from applying the same challenge twice to the same PUF.

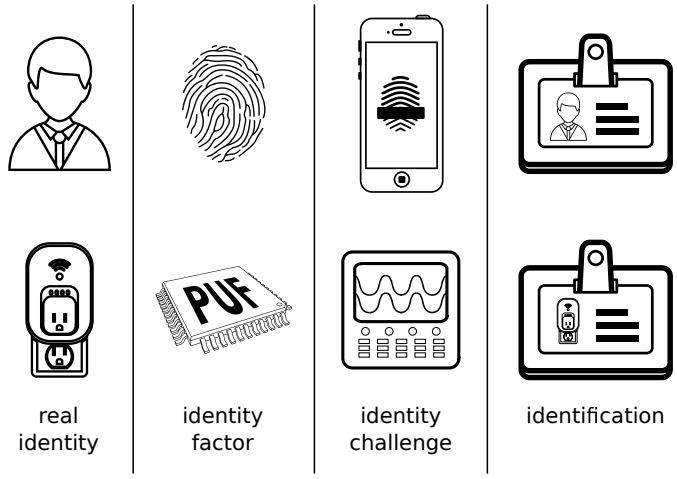

Figure 2.12: Real vs Digital biometric identities

Inter-distance and intra-distance values can change depending on the challenge structure and on the PUF implementation being used. Together are

able to outline how much a PUF is capable of delineate an identity and whether the recognition of such an identity is repeatable over time. In fact, on one hand, the intra-distance express the notion of *noise* in the response generation. The higher the intra-distance the more unstable are the responses when challenged with the same value. It is then clear that a small intra-distance better fits identification and authentication solutions as it yields very reliable PUF responses. On the other hand, the inter-distance value measures the distinguishability of two systems based on different PUFs but challenged with the same challenge. This value is as much important as the intra-distance one as it describes the uniqueness in the behavior of a PUF. The higher is the inter-distance and the more accurate is the identification process (as shown in Figure 2.13). However, it is important to note that it is difficult to have both such values and that usually a trade-off is necessary.

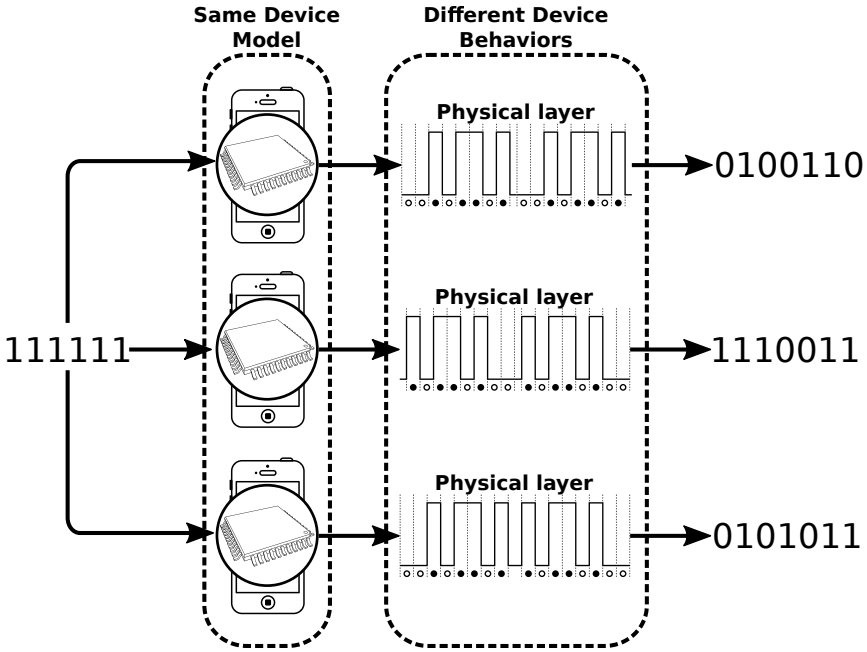

Figure 2.13: HIS-based device behavior

So far we have introduced two measurements that describe PUFs quality in terms of their statistical behavior. We will now describe their quality in terms of properties and features that PUFs can fulfill in order to better understand which scenario could benefit from their application.

PUF quality is usually determined by two main properties: (a) *reliability* and (b) *security*. On one hand, reliability means that PUFs have to be stable in their lifetime behavior. As such, for authentication and identification purposes where responses computed by PUFs represent identities, the reliability property describe whether a PUF is able to provide the same identity even when disturbed by external conditions such as (i) temperature, (ii) core voltage and (iii) electromagnetic radiation. Furthermore, it is also important that each PUF should work properly over time. In fact, it is known that silicon slowly degrades when used for a long time and PUFs stress the hardware layer each time a challenge is received. This effect on the hardware is named *aging effects* and several mechanisms contributes to it as follow [48]:

- **Electro Migration (for short, EM)**: the transport of conductor material due to momentum exchange between electrons and the metal lattice;
- **Hot Carrier Injection (for short, HCI)**: carriers generate sufficient kinetic energy to overcome a potential barrier and get injected into the gate oxide, causing interface states and charge traps;
- **Time-Dependent Dielectric Breakdown (for short, TDDB)**: formation of conducting path through the gate oxide;
- **Negative Bias Temperature Instability (for short, NBTI)**: build up of interface charges due to a negative gate-source bias at an elevated temperature.

The PUF instability brought by the above mechanisms has to be solved for specific scenarios where stable and repeatable CRP are required. Hence, in these particular scenarios, PUFs are challenged multiple times with the same inputs and their outputs are filtered before to be written within the CRDB thus achieving the *reliability* property.

As concern the security property, this is also of paramount importance and can be represented by three main parameters as:

- **Entropy**: in order to derive a high-quality identity factor from a PUF challenge-response behavior, a sufficient amount of randomness is needed in the PUF responses [54];

- **Tamper Evidence**: a tamper-proof hardware is a hardware element that cannot be tampered by attackers without having it's structure/behavior changed. Hence, tamper-proof hardware are used against threat models where attackers are assumed to be able to physically access the victim's device. PUFs, are highly susceptible to physical attacks and can be then used as tamper-proof elements [55];
- **Unclonability**: as already described, PUFs leverage on micro variations produced within the physical layer at manufacturing time. This randomness is involuntarily embedded within the PUF and, as such, it is unpredictable and not-reproducible thus making it impossible to create PUF clones [55].

In summary, the Hardware Intrinsic Security (for short, HIS) can be leveraged to prevent cloning of semiconductor products and to derive digital identities from the hardware behavior. PUFs are a practical way to implement the HIS approach and can be used to produce unclonable identity factors, unique per PUF and that are able to recognize distinct devices.

2.4 Device Fingerprints

As described in the above section, hardware-based approaches proved in last years to be useful in deriving unique, per-device characteristics that can be exploited for offline authentication within unattended networks. However, whilst HIS-based approached (such as PUFs) use special hardware components to derive a unique identity factor, other solutions are rather based on the device profiling. All the approaches based on profiles belong to the group of *device fingerprinting* as they try to define a fingerprint of the device. This approach is similar to the one already seen with HIS but, unlike PUFs that use special hardware elements with their intrinsically embedded noise, device fingerprinting tries to derive such noise from common hardware/software behavior or any other property that can help in the differentiation between devices.

A standard fingerprint approach leverages on a two-steps algorithm composed by an *enrollment step* and an *authentication step*. In both of them, the communication between a device to a cloud service is intercepted and filtered by a fingerprint authentication server that acts as a TTP for this protocol. As shown in Figure 2.14, the enrollment step is composed by the following operations:

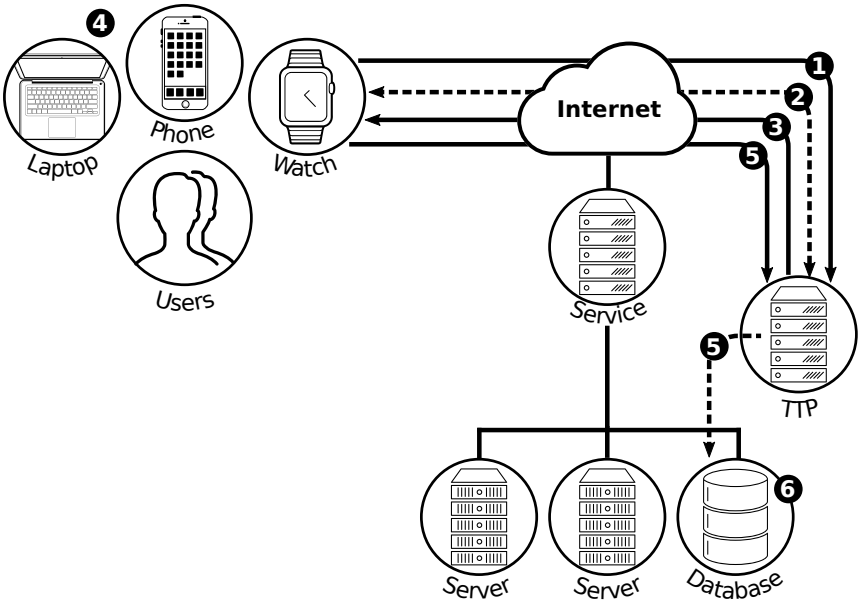

Figure 2.14: Fingerprint Enrollment Step

1. A user or device attempts to access a cloud service and is redirected to a TTP that is actually providing the fingerprint service;

2. The TTP engages a registration procedure in which the device is challenged with one or more authentication factors such as passwords, PIN codes, etc;

3. Once the authentication has been accomplished (i.e. the device has been recognized), the TTP sends commands to the device;

4. The above commands are used to pull from the device some unique characteristics such as headers, fonts, plug-ins, screen size, HTML5 storage facilities, IP address, cookie storage, etc. All of them aimed at building a profile of the device;

5. the TTP creates a virtual identity for the device and links all the above values to it;

6. the virtual identity just created with all its linked characteristics is stored within a DB and kept safe.

Once the enrollment procedure has been accomplished, the TTP knows the device fingerprint and can use it to challenge and to authenticate the device. The operations involved in the authentication steps are the following (see Figure 2.15):

1. A device tries to access some cloud service and its request is intercepted and redirected to the TTP that is providing a fingerprint-based authentication procedure;

2. During the authentication process the device provides its virtual identity and some of the special characteristics previously recorded during the enrollment step such as headers, fonts, plug-ins, screen size, etc.;

3. If the actual device characteristics match the one provided during the enrollment step then the device is automatically logged.

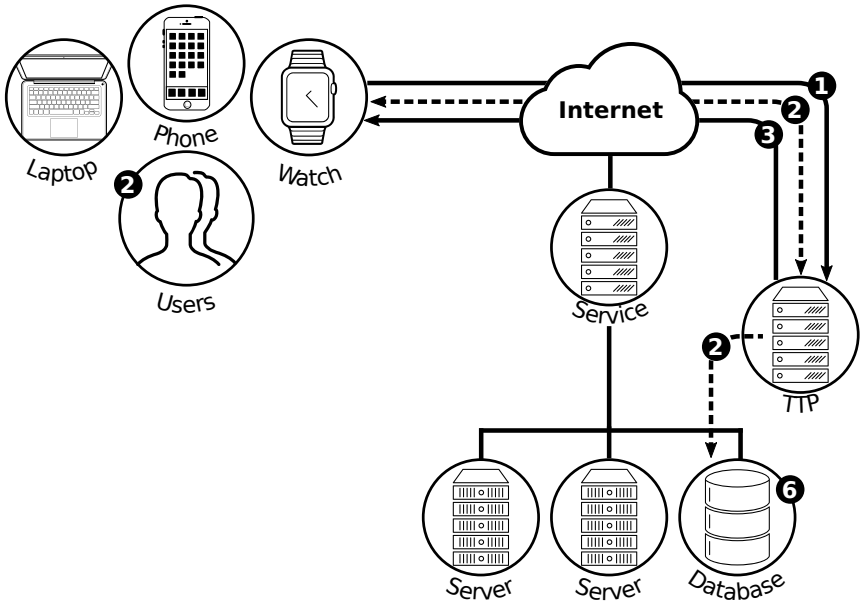

Figure 2.15: Fingerprint Authentication Step

Even though device fingerprint approaches are able to solve some issues related to PUFs, they still suffer from weaknesses. The most important one is the involvement of trusted third parties as described in Chapter 4. With PUFs the TTP is required as a trust anchor for the challenge-response

pair database linked to the device. In this case the TTP is the device manufacturer. In fact, the manufacturer is the only one that fully knows the device behavior and the responses produced as output when challenged with specific inputs. Therefore, in order to exploit PUFs in the cloud, service providers have to rely on TTPs.

The same happens with device fingerprinting solutions. Here a TTP is the one who securely stores private and unique identity factors about the user during the enrollment process. Such a TTP is then able to know, for each registered device, what are the major and minor versions of installed applications, browsers, operating systems etc. Again, the TTP is used during the authentication process in order to verify the correctness of replies provided by the device. Last but not least, current fingerprint approaches are not based on a challenge-response interaction but rely on data secrecy thus hindering their adoption in remote attestation and authentication systems where we cannot assume the data to be safe.

Part II

Identification in the IoT

Chapter 3

Name and Discovery

3.1 Related Work

Applying the blockchain and smart contract concepts (see Section 2.2) to the IoT provides different benefits that in the last few years drove to many solutions both in the academia and in the industry. Some of them are listed next:

- **Data Privacy**: sensitive data should not be given to third-parties where they are susceptible to attacks and misuse even though such third parties claim themselves to be trusted. Personal data should remain so and managed only by the owner. To tackle this problem, Zyskind et al [41] proposed a new platform where the blockchain technology is exploited as an access-control moderator, with an off-blockchain storage solution. On one hand, with this approach the blockchain is able to recognize the data owner and to give controlled access to other entities. On the other hand, thanks to the blockchain distributed mining process, the data owner is also capable of monitoring who else is accessing its data and why;
- **Intellectual Property**: software license control has been one of the first mechanisms to prevent software piracy. Such mechanism evolved first switching from offline to online validation and then from a centralized to a distributed validation. Herbert et al. [56] proposed a new protocol based on the blockchain technology that provides a decentralized and peer-to-peer software license validation. Their solution proved to be able to meet standard requirements for software license

validation but in a cost effective manner thanks to the cryptographic currency theory;

- **Distributed PKI**: current PKI approaches suffer from the lack of identity retention features as they do not prevent a user from registering a public key under the identity of already registered users. To solve this issue, Fromknecht et al. [57] proposed a solution that leverages the consistency check provided by the blockchain technology to build a PKI that ensures identity retention without relying on any central authority;

- **Trusted Timestamps**: formatted strings that represent a certain *date and time* when an important event occurred are crucial whenever we need to keep track of events' chronological order. Gipp et al. [58] proposed a trusted timestamp approach that leverages the blockchain technology to store anonymous and tamper-proof timestamps for digital content. Their solution, implemented as a web-service, allows users to hash files and store the hashes within the blockchain to later verify their creation time thus enabling anyone to prove the possession of some information at a given point in time.

Other solutions such as decentralized computation platforms [59], verifiable computations [60] and affine commitments [61] showed the great potential of the blockchain technology. However, they were all focused on forks starting from the original cryptographic currency-based approach, thus not being able to be deployed in the IoT ecosystem as already described in Section 2.2. Finally, at the beginning of 2015 IBM, in conjunction with Samsung, launched a project called *Autonomous Decentralized Peer-to-Peer Telemetry* (ADEPT) [62] that uses the blockchain to build a distributed network of things. ADEPT uses a mix of proof-of-work [63] and proof-of-stake [44] to validate transactions. The architectural approach adopted by ADEPT recognizes that power constrained things may not have the full computational power and memory to manage the complete blockchain while others may be powerful centers of interaction. As such, three different node entities have been defined as follow [62]:

- **Light Peers**: things with low memory and storage capabilities. They can manage their own data but need to rely to other peers to work with the blockchain;

- **Standard Peers**: things equipped with higher storage and processing capabilities. They are able to manage the blockchain;

- **Exchange Peers**: high end things with vast compute and storage capabilities. They can be owned and operated by organizations or commercial entities. This kind of peer is also capable of manage complete copies of the blockchain.

Albeit this practical solution proposed by IBM proved to be able to exploit smart contracts' properties as well, it still suffers from some open issues such as:

- **Scalability**: blockchains are able to store transactions back in time starting from time 0. As such, albeit special powerful nodes can be designed to manage it, one global blockchain will soon or later be hard to manage;

- **Peer-lists**: blockchains have not being designed to discover things in the IoT. This means that, albeit a blockchain can be used to save the history of a thing, still there is the need to have a peer-list. In this way, it is first required to identify a thing and then the ID of such a thing can be used to browse the blockchain;

- **Single Points of Failure**: by using *special* nodes all running the same code (as the exchange nodes in ADEPT [62] and the gateways in Hypercat [64]), malicious users can exploit unknown or undisclosed vulnerabilities in that code to bring down the whole network;

- **Privacy**: having one single blockchain allows all the nodes within the network to have access to the transactions of all other nodes thus hinder their privacy.

Mostly all the current approaches are trying to design a solution that is able to merge all the different blockchain solutions such that they can communicate to each other [65]. Albeit this is another step towards the design of a blockchain interconnected world, it is still based on banking scenarios and does not solve the open issues listed above.

Name Resolving

Contemporary approaches aimed at the integration of peripherals within the IoT are all facing the same problems [66]. In fact, the integration of devices remains an hard task that involves extensive hardware and software configurations thus limiting the range of applications that the IoT can tackle [67]. The main problem is that, before things can communicate, they need to find and to connect with each other. In the classic Web scenario, a Domain Name Service (for short, DNS) is used to map human readable uniform resource identifiers (for short, URI) to IP addresses. As a consequence, URIs can be used or coded in applications, scripts and software while actual IP addresses might change over time.

In the IoT the situation will be different. A wide number of different things, coming from distinct manufacturers and using diverse protocols, will need to interact thus requiring a mapping between real names (such as IDs) and logical names (such as IP addresses). To fulfill this requirement, custom identifiers from different name-spaces such as Jabber-ID [68], CAN-Bus ID [69] or Z-Wave Home-ID [70] may use common name management infrastructures (such as DNS) but this might create a heterogeneous ecosystem. This situation could then suffer from:

- **Custom Syntaxes**: different IDs might be incompatible to each other. As a consequence, things could be unable to communicate. As a toy example, things powered by protocols designed to work with 32 bit IDs might be unable to interact with things adopting 16 bit name protocols;
- **Redundancy**: without neither infrastructures nor regulations, different things might share the same ID thus making hard for an external player to reach them.

As an example, in smart farms, thing setup might change every day with new trucks, tractors and other machines added and removed as needed. In such a scenario, machines need to find each other although they use different IDs. In fact, harvesters might use serial numbers, tractors might use CAN-BUS IDs and trucks might be identified by their license plate. However, they still need to communicate, and share information thus requiring a mapping function to translate different names into their own readable name-space. Such a mapping is usually achieved as follows:

- **Manual Setup**: users manually configure and distribute public names such as IP addresses. This is rather a theoretical approach but often used for demos, where discovery and identity management is out of scope. However, this approach does not fit in autonomous ecosystems where things are added, removed and modified at run-time;

- **Semi-automatic Setup**: a platform provider can be responsible for the name mapping procedure. Users would have to register their things to the provider and then connect them through the platform. In this approach, each new member has to become a part of the solution and has to register with the platform thus hindering its privacy. Furthermore, vulnerabilities affecting the platform might be exploited to attack things as well;

Other projects focused on IoT discovery protocols [71; 72; 73] with the design of new name standards. However, those proposals were either focused on solving the security or on improving network performances [74] but not both of them. Furthermore, the definition of new name standards would either force vendors in changing their own solutions [75; 76] or leverage on TTPs [77] or data sets [78]. Hence, the challenge with heterogeneous environments is to find a democratic and self-organized way to uniquely identify things without forcing manufacturers in the adoption of new name spaces.

3.2 Attribute-based Identities

To solve the naming and discovery issues described so far we have proposed a CONtextual NamE disCovery and resolving with Transactional security solution (for short, CONNECT). To the best of our knowledge, our solution is the first that is able to solve name mapping open issues in the IoT while allowing things to be automatically discovered and accessed at run-time. Compared to other solutions [79; 80], the core novelty of our proposal is the capability of exploiting relationships and context aware information [81] in order to build trusted identifiers on top of the blockchain technology. Unlike standard approaches, CONNECT does not require any static syntax for names but is based on dynamic information.

The main difference between CONNECT and other static IDs (as shown in Figure 3.1) is the capability of identifying and connecting to things even though they share the same ID. This is possible as things usually have

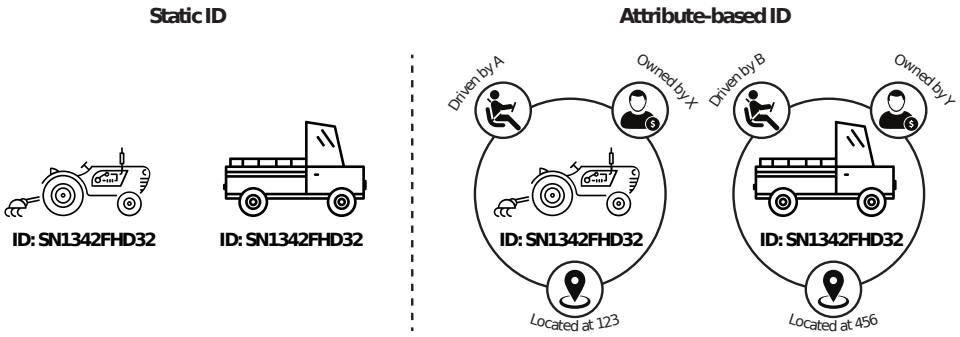

Figure 3.1: Static vs Attribute-based ID

different attributes, relationships or surrounding contexts, that are defined as follows:

- **Attributes**: in general, an attribute is a property or a characteristic. *Color*, for example, is a hair attribute as well as *owned by*, *driven by* and *located at* can be vehicle attributes (see Figure 3.1);
- **Relationships**: connections between two or more things, either physical or digital (e.g. one person in a relationship with one or more things or machine to machine interactions) that have to be provable, actionable and also revocable [82];
- **Contexts**: any information of the surrounding environment composed by attributes and relationships. A context is able to describe the actual state of one or more things. Context information can be measured by sensors or exchanged messages and combined with other context information.

As a toy example, a harvester can be identified by attributes such as manufactured by company X, sold by merchant Y, owned by Z, able to W and located at 123. All such attributes are not unique by themselves (i.e. things belonging to the same manufacturer will have the same *manufactured by* attribute) and, by themselves, cannot be used to define an ID but all together can be used to provide detailed and unique descriptions of things (as also depicted in Figure 1.3 for for street names). CONNECT core idea is to re-use attributes, names and labels to create global and unique names.

In Figure 3.2 an example is given where, an available, unloaded and refueled truck located in Berlin field ABC and able to crop corn ($Cr-Co$) is identified

by attributes instead of by IDs or SNs. As shown at the top of Figure 3.2, attributes describing a thing can be hierarchically designed. In CONNECT, the root is called *attribute ID* and identifies the attribute. Following the root, hierarchical sub-levels can be defined as *attribute classes*. They can be used to describe, more in detail, thing attributes and thus to better identify things. Hence, the more attribute classes are used, the more specific is the query and, thus, the more accurate is the result of the discovery process.

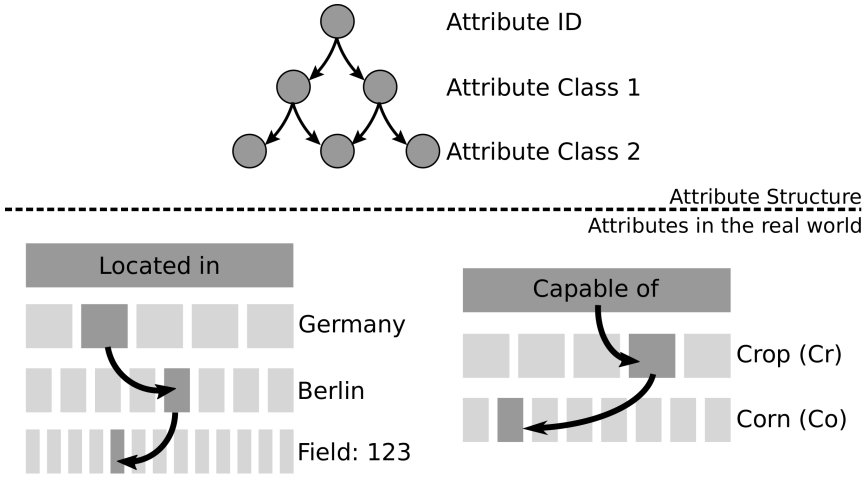

Figure 3.2: Hierarchical structure of attributes.

Unlike web-crawling approaches, the blockchain-based discovery process provides the following advantages:

- **Automatic**: it is automatically fed by the things themselves;
- **Scalable**: quality of the name-resolve query process is strictly dependent by both the number and precision of attributes currently used by the thing, and the number of attributes used in the query;
- **Tweakable**: the query process can be additionally refined if necessary;

As things in CONNECT do not need to have static IDs, queries such as the one shown in Equation 3.1 are replaced by dynamic and more natural queries based on attributes such as the one shown in Equation 3.2.

$$Truck\ A \implies \begin{cases} [\textbf{IP: } 192.168.0.42] \\ [\textbf{SN: } AT45FGR] \end{cases} \tag{3.1}$$

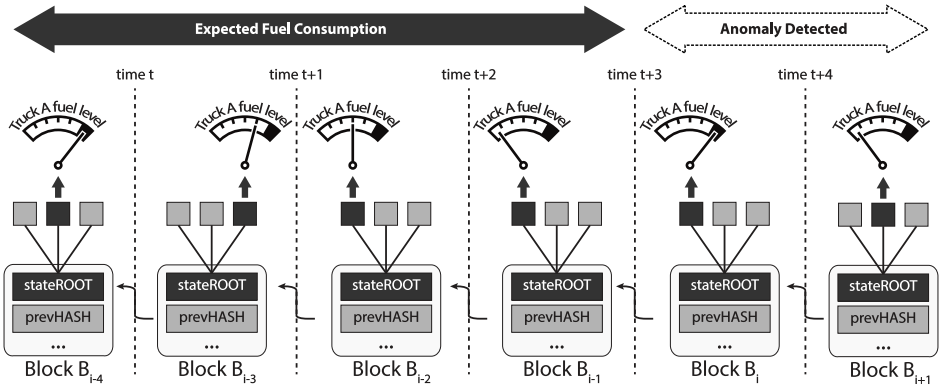

Figure 3.3: Blockchain-based event control

$$Truck\ X \Longrightarrow \begin{cases} [\textbf{LOCATION:}\ Berlin - FieldX] \\ [\textbf{FEATURE:}\ Crop\ Corns] \\ [\textbf{CAPACITY:}\ Five\ Quintals] \\ [\textbf{FUEL LEVEL:}\ Full] \end{cases} \tag{3.2}$$

As such, not only the syntax used but also the whole meaning of a query is different. With our attribute-based queries the focus is moved to the service rather than to the device that is able to provide it. This also improves the network availability and creates a better user experience. In fact, whilst in the Equation 3.1, a specific truck is required regardless of whether that machine is available or not, in the Equation 3.2 the user is not interested in which truck he is going to use but rather on finding a truck that can satisfy his needs.

This approach, based on a semantic-driven IoT [83], allows non-expert users or casual things to focus on the intent rather than on things. An example is depicted in Figure 3.3 where the attribute *fuel-level* is stored within the blockchain. By having access to a history of past fuel operations, at time $t + 3$ a query such as [A truck with fuel-level = $FULL$] can be broadcast to the network. With such a query it is possible to express the need of a truck with a long autonomy, regardless of which is the exact truck that is going to be used. Furthermore, by having access to the whole history of attribute changes, additional information can be derived from the blockchain as shown in Figure 3.3. In that example, frequent refueling requests within the blockchain can be identified and affected things can be further controlled for engine anomalies.

As in CONNECT the discovery process is built on top of attribute-based queries, the current state/value of each attribute need to be trusted by the network. On a blockchain-based approach as the one used in CONNECT, this means that each attribute change has to be validated by all the things in the network. As such, every time a thing wants to change one or more of its attributes such as *unload cargo* or *change driver*, it must send requests for such operations that have to be analyzed and validated by other things before they can be applied.

Communication Scheme

In CONNECT, all the information is distributed and decentralized in the form of a connected and multi-layered graph. Within this graph, two different nodes have to be defined:

- **Attribute Nodes**: (for short, ANodes) are logic nodes in the form of smart contracts (see Section 2.2). These nodes are responsible for the administration of all the operations that involve attribute changes and also for the attribute-based discovery process;
- **Thing Nodes**: (for short, TNodes) represent physical nodes (i.e. physical things) in the IoT.

The multi-layered graph exploited in CONNECT and composed by ANodes and TNodes is depicted in Figure 3.4. As shown, each ANode belongs to a layer that describes a particular attribute and each layer has its own blockchain. This means that changes related to an attribute X will be stored in the blockchain X, designed to keep the history of such an attributed, and distributed among ANodes within the *Attribute X* layer. ANodes and TNodes can communicate to each other in a self-organized approach [84]. However, whilst in standard blockchain-based solutions each request is broadcast to the network, in the IoT the same approach might cause a network overload. To avoid such an issue, in CONNECT requests are not always sent to the whole network but each thing holds a routing table (as big as possible depending on resource capabilities) that is used to directly target nodes holding the desired resource or information. As such, three types of interactions have been designed as follow:

- **TNode to TNode**: this is a connection between two physical things. As message passing protocols are accomplished in the physical layer,

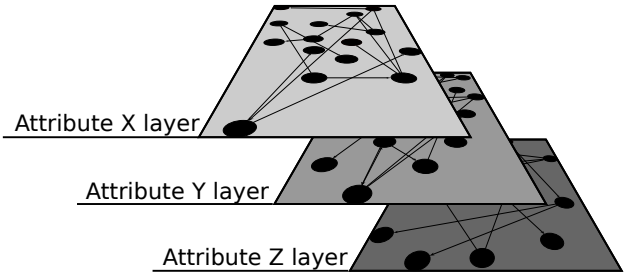

Figure 3.4: Physical nodes and logic nodes layered structure

each time two things need to communicate, a message is sent directly or by using a routing protocol (see Chapter 3.2);

- **TNode to ANode**: this is a connection between a physical thing and a smart contract. As ANodes are loaded and executed within TNodes in the form of virtual machines, once a message has reached the thing holding a smart contract, the rest of the communication is handled locally;

- **ANode to ANode**: this is a connection between two smart contracts. Smart contracts can communicate to each other in order to accomplish complex tasks. As an example, if a truck is required to accomplish a job, the smart contract responsible to manage trucks availability will communicate to the smart contracts responsible to check the fuel level and the cargo capacity of the truck. This kind of communication is handled by the TNode holding them if both the ANodes are executed within the same TNode otherwise a TNode to TNode communication is required.

Figure 3.5 depicts an example of ANodes stored and executed by a TNode. The figure shows that a single TNode (i.e. a thing) can manage different ANodes (i.e. smart contracts) and that ANodes can be connected to each other both in the same blockchain or in different blockchains (i.e. on the same layer or between different layers).

TNode to TNode communications are accomplished on the physical layer and have to follow practical constraints such as computational limits, battery levels, etc. As such, three different routing schemes have been designed as follow (see Figure 3.6):

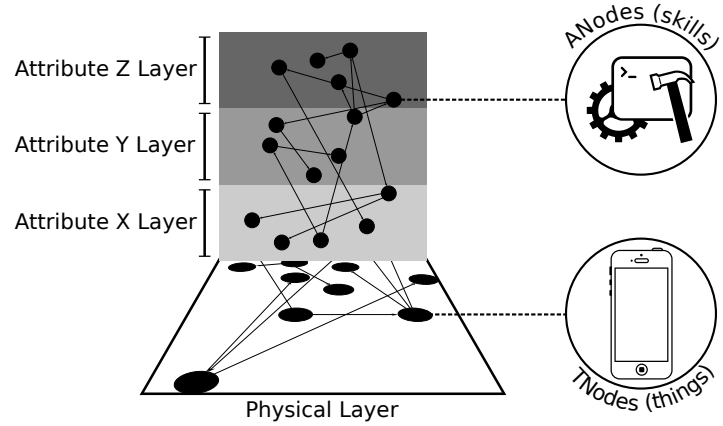

Figure 3.5: Multi-layered per thing attribute storage

- **Direct**: if a TNode already knows which other TNode is the one storing the required ANode and it also has the capability to directly contact it, than the message is directly sent;
- **Routed**: if a TNode already knows which is the other TNode storing and executing the required ANode but it cannot directly send a message to it, it will route its request through the network;
- **Broadcast**: if a TNode does not know which is the TNode that is storing and executing a particular ANode, a broadcast message is sent to the whole network.

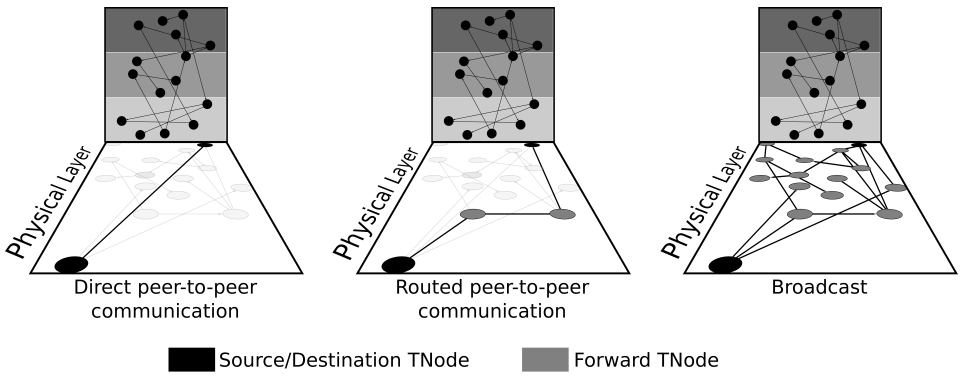

Figure 3.6: TNodes communication available schemes

As our CONNECT solution does not rely neither on PKIs nor on TTPs, once a path between two TNodes has been identified, each message exchanged in the path has to be authenticated hop by hop. Hence, one single authentication process will be executed in the case of a direct communication whilst multiple authentication processes have to be accomplished if a routing protocol is needed (see Section 4.3).

Cooperation Scheme

Unlike standard blockchain-based solutions which are based on a mining approach (see Section 2.2), CONNECT does not require any mining process but yet it is able to incentive TNodes in joining the blockchain validation. By analyzing our *proof of knowledge* approach described in Section 4.3, it can be seen that the reward factor is embedded in the message exchange process. In fact, each TNode exchanges messages composed by some data and a knowledge token. The former contains the information for the recipient whilst the latter contains evidences of the actual surrounding context as seen by the sender. For the sake of simplicity, as CONNECT is based on the blockchain technology, we will refer to the process of *sending messages* to the network as to transactions. We can then have two different transactions:

- **Validated Transaction**: if the message is accepted by the network, this new interaction updates TNodes' attributes. Such new attribute values represent an updated version of the network thus forcing all other TNodes in the update of their old values;
- **Refused Transaction**: if the message is rejected by the network, this information is sent as a feedback for other TNodes. This interaction between TNodes, even though rejected, again changes the network state. Hence, new attribute values represent an updated version of the network and force all other TNodes in updating their old values.

In both cases, the information exchanged updates the attributes of involved TNodes thus updating the state of the network. As such, the more transactions are verified by a TNode, the faster is for it to update its knowledge about the actual surrounding context. This reduces the authentication overhead and makes the network faster and more secure thus representing a valid incentive for all TNodes (details in Section 4.3).

Part III

Authentication in the IoT

Chapter 4

IoT Online Authentication

4.1 Related Work

Generally speaking, authentication can be defined as the process in which one entity is ensured of the identity of another one [85; 86; 87]. Authentication is usually mixed-up with identification. Nevertheless, identification is the process of giving trusted names to entities whilst authentication is about proving to be the owner of one of those names. We have already described the problem of identification within the IoT in Chapter 3. Now we are then going to focus on authentication issues from both an online and offline perspective.

A general authentication model is based on two entities called claimant (for short, C) and verifier (for short, V). The claimant usually provides some *token* proving its identity whilst the verifier has to check for the validity of such a token thus either accepting or rejecting the claimant's request. Each authentication scheme can then be described as a process aimed at providing the following two properties:

- **Feasibility**: in the case that C is truthful, V has to be able to verify the given identity;
- **Non-Transferability**: the identity token given to V_1 by C cannot be used by V_1 to authenticate to another verifier V_2. If this property is not ensured by the authentication protocol, then each verifier could impersonate one of its previous claimants.

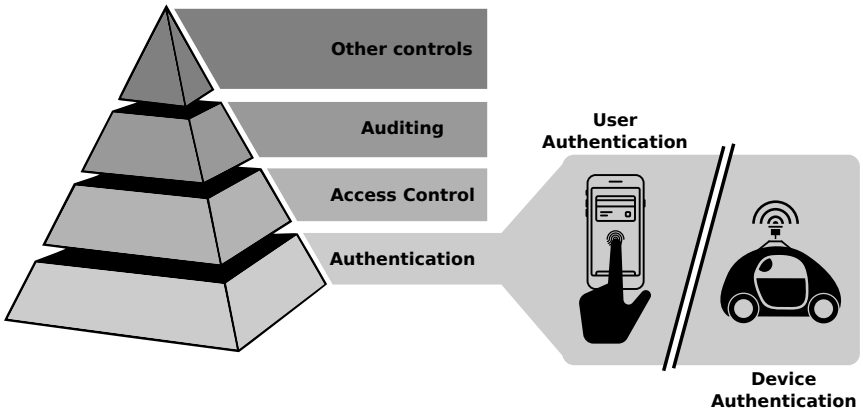

Figure 4.1: Security layers

So far, we have always been used to *user authentication* protocols, i.e. to protocols where the claimant is a person and the verifier is any device the user needs to authenticate to. However, as shown in Figure 4.1, the authentication layer which resides at the basement of any system security policy, is composed by user authentication and device authentication. Albeit both are responsible to verify the identity of the claimant, there is an important distinction between them. User authentication protocols can rely on knowledge factors, i.e. on the people capability to remember secrets. Well known examples are PIN codes or passwords. These knowledge factors can be used for device authentication protocols as well but they need to be stored within the device thus threatening the whole system as they can get stolen or changed.

In the remaining of this section we introduce both user and device authentication states of the art. Further, we describe what is the challenge in device authentication from both the offline and the online perspective. For each perspective, a solution is given along with a security analysis and a comparative study against what has been already published/proposed.

User Authentication Protocols

With the advances in information and communication technology, performances and features of hand-held devices rapidly increased over the last years and more devices were started to be used as access points to the virtual world. In particular, in last years, mobile phones have become everyday personal devices. Formerly only a gadget to business users and aficionados

or geeks, smartphones nowadays serve as tools for organizing users daily lives through productivity applications such as calendars, notepads or e-mail clients [88]. They are turned into multimedia toys with capabilities to play music, videos, and games or surf the World Wide Web and take pictures using integrated cameras. Smartphones are also capable of serve health, emergency services, defense, education, banking, retailing, and other sectors benefiting from information services. Last but not least users also store a vast array of different data on their devices, ranging from personal pictures to messages, emails, contact lists, addresses, birth-dates, music, movies and various other files. Smartphones can therefore be considered a light version of computers with ubiquitous telephony functionality. What we do not have time to do on computers or laptops, we do on these devices and this makes our lives much easier.

One of the main features of such devices is their small size. This fact makes users being able to bring their devices in the pocket and use them during the whole day. However this is also something that can thwart our security and privacy as they can be easily lost or stolen. As a direct consequence, this creates the need to protect smartphones data access as mobile devices ending up in wrong hands represent a serious threat to information security and user privacy. Even worse, users usually do not protect their devices typically due to either impatience with authentication processes or their ineffectiveness considering security and memorability aspects [89].

Traditional password schemes based on a mix of alphanumeric and symbols proved many times to be cumbersome and unpopular thus bringing to light a wide gap between usability and security, which combination still represents an important challenge for researchers. Password-only authentication has been under attack for years by means of multiple techniques such as phishing scams and key loggers. Once password requirement became more complex, users started to avoid using them due to the poor user experience. In many cases this also led to bypass strategies on behalf of the user, such as choosing the same password or PIN for different applications/services or opting for passwords that are easy to remember, such as birth dates or names. Hence, the user turned out to be the weakest link in the security chain and this brought to light the importance of simplicity and acceptance factors in authentication approaches. As an example, to avoid complex passwords or PIN codes authentication schemes Google has introduced the Android Pattern Lock that, inspired by the rapid and effective Draw-a-secret system [90], requires users to enter strokes connecting a specific pattern of on-screen dots. Two key problems with this method are that it is highly susceptible to

observation [91] and also decreases the input space by enforcing connections only between adjacent points and by disallowing repeated selections. Additionally, attackers can infer password patterns from the oily residues left by fingers stroking on the phone screen (a smudge attack [92]). A partial solution to this problem has been done by exploring variations in item selections [93], however this approach still relying on a knowledge factor thus focusing on the user.

This unattractive situation can be improved by exploiting device intrinsic sensors and by applying unobtrusive authentication methods that do not require neither explicit attentions nor actions from users. The rich set of mobile device input sensors including cameras, microphones, touch screens, and GPS, enables sophisticated interactions. Biometric authentication methods using these sensors could offer a natural alternative to password schemes, since the sensors are familiar and already used for a variety of mobile tasks. In the remaining of this section, first we introduce a classification made on the intrusiveness and dynamism of authentication protocols and then we provide an analysis of different aspects and techniques that were used in the past to improve device security.

Intrusive vs Non-Intrusive Authentication

Mobile device protection mechanisms are usually based either on PIN codes, passwords, or biometric-based methods, such as fingerprints [94] or IRIS [95]. Passwords and fingerprints are intrusive in the sense that they require explicit action from the user. However, according to recent surveys [96], 60% to 80% of users choose to turn these verification features off simply because of their inconvenience thus bringing to light the need of non-intrusive authentication mechanisms [97]. Recently, biometric authentication modalities are proposed as non-intrusive methods for smartphone users. As an example, in [98] a gait-based authentication solutions is proposed which exploits the device accelerometer and is able to seamlessly recognize the user during motion activities such as walking. Another solution proposed in [99] uses both the accelerometer and the orientation sensor to authenticate a smartphone user when answering (or placing) a phone call and last but not least in [100] a non-intrusive multi-modality authentication system is proposed, based on four different smartphone sensors, the microphone, GPS, touch screen, and accelerometer. All the above solutions show the trend towards non-intrusive authentication methods where responsibilities are moved from the user to the device.

Static vs Progressive Authentication

The problem of mobile authentication can also be studied from a completely different point of view. Rather than exploring new and smarter authentication schemes, it is also needed to study the problem of when to trigger the authentication process. In fact, unlike desktops and laptops, which users tend to utilize for long and continuous periods of time, mobile devices are accessed periodically or in response to a particular event (e.g., incoming email notification). This lack of continuous interaction with mobile devices creates the need to authenticate users on demand each time it is required. However, even though the interaction between users and mobile devices is not continuous, as users use to put the phone into the pockets after a phone call, a physical contact/proximity with might still hold. Hence, it would be possible to keep the user logged in such that once he pulls the phone out of the pocket, no authentication is required. On the other hand, if the phone lost contact with the authenticated user (e.g., left on a table), then authentication should be required. As a result, if the phone is able to accurately infer the physical interaction between the authenticated user and the device (e.g., through its embedded and surrounding sensors), it can extend the validity of a user authentication event, reducing the frequency of such events. This approach not only significantly lowers the authentication overhead on the user, but also makes the effort proportional to the value of the content being accessed. In fact, if the system has strong confidence in the users authenticity, the user would be able to access any content without explicitly authenticate. If the system has low confidence in his authenticity, the user would only be able to access non sensitive contents (e.g., a weather app) and would be required to explicitly authenticate for sensitive contents (e.g., email, banking).

Progressive authentication establishes the authenticity of the user by combining multiple authentication signals (multi-modal) and leveraging multi-device authentication. The goal is to keep the user authenticated while in possession of the device and de-authenticate the user once him/her drops it (i.e., a discontinuity is detected). The confidence level in the users authenticity is then compared to one authentication threshold for a single-level approach, or to multiple authentication thresholds for multilevel authentications. Possible signal types used for multi-modal authentication could be for example biometric signals, behavioral signals, possession signals and secrets. In combining these signals, several challenges must be considered. First, most signals are produced using unreliable and discrete sensor measurements. Second, certain signals may require combined readings from

sources with different sampling frequencies and communication delays. As a result, most signals are not individually sufficient to determine user authenticity and when combined they may be inconsistent as well. Finally, signals vary in strength. Some signals can provide a stronger indication of authenticity than others because, for example, some may be easier to fake and some are more discriminating than others. For all these reasons, signals need to be combined and cross-checked. However, drawing the correlations across these signals manually is a cumbersome job, prone to errors and inconsistencies.

Continuity is another cornerstone of progressive authentication. It comes from the observation that users are likely to use their phones shortly after a previous use. For example, after the user reads emails, he locks the phone to save energy but he probably keeps holding the phone and talking to someone else. When he tries to use the phone five minutes later, the phone is locked even if he did not let go of it. If the user has been touching the phone (i.. actively holding or storing the phone in a pocket) since the last successful authentication, the authentication level should be maintained unless negative signals are being received (e.g., mismatching biometric signals). A phones placement with respect to the user can be determined by accelerometers, touch screens, light, temperature and humidity sensors, most of which are embedded in modern phones.

Progressive authentication takes also advantage from device connectivity to gather information from other devices owned by the user. If a user is logged in and active in another nearby device, this information represents a strong signal of the users presence. It can also be able to link user authenticity with different confidence levels. This enables the system to leave the all-or-nothing paradigm and allows the user to associate distinct protection levels to different data and applications.

State of the Art

We are now going to individually introduce all those user authentication solutions that have distinguished themselves in the literature for originality, efficacy and efficiency:

- **Accelerometer-based**: in [99] the authors propose a new gesture-based authentication method that offers transparency by identifying if the user that is answering (or placing) a call is the authorized one. In particular, they investigate if a user can be authenticated just by using the movements he performs, from the moment he presses start (to

initiate a call), until he brings the phone to the ear. This movement is named pattern and it is treated as a biometric factor. The authors have demonstrated that there are sufficient differences between different users, such that the movement can effectively be used for identification purposes. In this way, when a call is answered (or placed), the phone can promptly evaluate if the user is authorized to perform this action, and block the system in case of non authorized users.

Another possible use of this system is to perform forensics analysis, as an example to investigate who used the phone at a particular point in time. It is important to highlight that, differently from the solution proposed in [101], in this work the secret is not the answering movement itself but the biometric measures of that specific movement. This means that even if an adversary spies how the user answers the phone, he will not be able to reproduce the movement in a way such that it can replicate the biometric features of the correct user. For the implementation, the authors used both the accelerometer and orientation sensor to measure movement patterns. The proposed biometric measure resulted not only to be effective but it also proved to have a unique feature, it can be transparently used to authenticate a user that is answering (or placing) a phone call, without being affected by external factors (like light exposure or users wearing hats or veils);

- **Gyroscope-based**: in [102], authors proposed 53 new features based on the readings of the orientation sensor to capture the behavioral biometrics of smartphone users. The goal was to investigate the feasibility of using behavioral biometrics collected from the orientation sensor to authenticate smartphone users. To demonstrate the feasibility of the proposed approach, Chien-Cheng Lin et al. developed an app for Android 2.2 aimed at collecting biometric information from the orientation sensors that belong to 11 users when they operate the smartphones in their hands. For each smartphone user, an authentication model is constructed based on 53 new features representing behavioral biometrics that includes the movements of wrist flexion (or extension), the forearm pronation (or supination), and the wrist radial (or ulnar).

  The applications of the hold-and-operate biometric can include authentication and access control. It has been reported that the physiological approaches (such as fingerprints) typically show better performance than behavioral models [103]. However it should be noted that

in [102] authors do not propose orientation sensor as a replacement or sole mechanism of authentication but rather as a complementary mechanism that can be used to improve security in hand-held devices. Users can still use strong biometrics or passwords when authenticating for the first time. Then, orientation biometric can be applied for re-verification in a continuous authentication scenario;

- **Gait-based**: in [104] Nickel et al. proposed an unobtrusive authentication approach for accelerometer-based smartphones biometric gait recognition. Such an authentication method enables the mobile phone to recognize its owner based on the way he walks.

  There are two main advantages of the approach proposed in [104]. First, gait can be captured via acceleration sensors, which are already integrated into smartphones. Hence, there are no additional hardware costs for deploying this method. Second, gait recognition does not require explicit user interaction during verification as the phone does it literally on-the-go. These two factors make accelerometer-based biometric gait recognition a very user friendly method, which does not require extra interaction time. The contribution of [104] is twofold. The k-nearest Neighbor (for short, k-NN) algorithm is comprehensively evaluated on a database collected using a off-the-shelf smartphone. The biometric performance is compared to the one obtained when evaluating Support Vector Machines (for short, SVMs) and Hidden Markov Models (for short, HMMs) on the same database [105]. In addition, the algorithm is implemented on a smartphone and it is shown that the complexity of feature extraction and comparison is low enough to be applicable in practice. During enrollment the users are requested to walk for five minutes. Afterwards the classifier is constructed using the enrollment data of the user and data of 20 other users. Using pre-computed feature vectors from the impostor data, training takes around 1.5 minutes on a Motorola Defy smartphone running Android 2.2. This is a short enough time, because enrollment is only rarely performed. The effort for the user during enrollment is restricted to five minutes of walking. Authentication is currently based on 30 seconds walk data. The whole process takes around seven seconds, a short enough duration to implement a continuous classification. By iteratively collecting data and performing the classification, a current classification results is available when it is required. Therefore, an authentication consists only of retrieving the

last authentication result from the system, which can be done without temporal delay;

- **Biometric-based**: in [106] the authors proposed a biometric authentication with fingerphoto recognition via the smartphone built-in camera. The latest smartphones have at least one integrated high resolution camera to capture the finger in sufficient quality and enough computational capacities to process the photos and execute algorithms for the fingerphoto recognition. Hence, no extra devices are needed. The capture process is performed touchless and no latent fingerprints are left, which is an advantage over many classical fingerprint sensors. Additionally, biometric authentication methods have clear security advantages compared to knowledge factors-based methods as biometric characteristics cannot be delegated, forgotten or copied like.

  The main problem with fingerphoto is that smartphone cameras are not designed for biometric use. Not all cameras are able to focus on the necessary close distance to capture the pattern ridges of the finger and the depth of field is very limited. If the finger is too far away from the camera, the effective usable resolution of the fingerphoto is reduced and the risk that the finger cannot be detected increases. Additionally, the low amount of configuration possibilities of the smartphone cameras tightens the conditions for the fingerphoto recognition. Another problem is that the sensors of the cameras are usually small due the compact design of the smartphones. Thus, these cameras tend to produce higher noise having a high impact on the photo quality. Hence, various potential poses of the finger must be considered such as the orientation angle, the pitch angle, the position of the finger, the distance of the finger from the camera and the background. The fingerphotos are also affected from different light conditions that have impact on the finger recognition. In addition, the structure and the consistency of the finger, like bulge, peculiarity of the finger ridges, wear and dirt, have also influence of the quality of the fingerphoto recognition. Results obtained in [106] show that fingerphoto recognition on smartphones is possible.

  Ear biometrics is another passive approach. Authors in [107] proposed an authentication solution based on ear biometrics as ears do not change over time, whereas face changes more significantly with age. Furthermore, color distribution is more uniform in ears than in human face, iris, retina, etc. Besides, ears are smaller than faces,

which means that it is possible to work faster and more efficiently with lower resolution images. In addition, it is important to note that ear images cannot be disturbed by glasses, beard or make-up. This solution silently captures ear shape images and authenticate users based on them thus obtaining a non-intrusive process;

- **Feedback-based**: in 2012, Azenkot et al. proposed the first work on mobile device security for people with disabilities, in particular blind people [108] with a new accessible and secure authentication method called PassChords. PassChords are based on input finger detection [109] and consist of several multi-point touches, defined by the set of fingers touching the screen. The PassChords algorithm determines which fingers touch the screen in each tap based on an initial set of reference points which the user inputs anywhere on the screen. Reference points indicate the approximate position of the fingers. PassChords have no audio feedback but vibration feedback thus been resilient to aural eavesdropping. In a study with 16 blind people, authors found that PassChord entry was nearly three times as fast as entry of accessible personal identification numbers (for short, PINs) and had about the same authentication failure rate.

In the next section we will focus on device authentication. Differently from human authentication, devices cannot remember secrets without storing them thus making them more vulnerable to attacks and requiring different approaches to be analyzed.

Device Authentication Protocols

One of the well known and commonly used device to device authentication protocol is based on IEEE 802.11 and more in detail on the MAC address. However, MAC addresses can be easily changed thus allowing malicious devices to impersonate other devices. As a result, an attacker could modify the MAC address of its rouge access point (for short, AP) to match that of an existing authorized device and connect to the network without being detected. The Bluetooth pairing system is also another well known and commonly used communication system with its own authentication protocol. However, as for the MAC address, in Bluetooth devices are automatically paired based on tokens stored within the claimant thus making the attacker able to steal them and misuse them. The same problem can easily arise for other machine to machine communications such as in MANETs, Wireless

Sensor Networks (for short, WSNs) or Radio Frequency Identification (for short, RFID) based systems as all of them share the same issue: the lack of any mutual device to device authentication procedure.

The core element of device authentication is the effective enforcement of network access policies. In fact, by checking the identity and authenticity of devices before that they get connected to the network and that any message is exchanged, it is possible to provide the following benefits:

- **Consolidation**: a secure authentication applied to remote devices or third party hardware trying to access a network enables the consolidation of access policies into a comprehensible plan;
- **Control**: device authentication can play an important role as a key enabling technology for e-government or other agencies interested in the control of both their employees and devices;
- **Trust**: in highly pervasive environments such as the IoT, scenarios with smart devices able to autonomously authenticate to each other can impressively increase the trust factor and boost the overall system adoption.

We are now going to describe the most used and well known machine to machine authentication approaches that have been proposed so far as to give an overview on the state of the art:

- **IEEE 802.1X Framework**: the 802.1X framework [110] provides various Extensible Authentication Protocol (for short, EAP) [111] as well as certificate oriented mechanisms for user and device authentication. Based on this framework, devices are able to build their unique identity as a mixture of hardware and software characteristics. As an example, hardware parameters can be MAC addresses rather than processor types or memory capacity whilst software characteristics can be the hash value of some drivers rather than portion of the memory or storage sections. A careful choice of such characteristics all together might result in a unique identity factor capable of identifying devices even though belonging to the same model. However, all these characteristics are static in nature and over time they might get changed (see Section 2.4).

Once a set of properties has been chosen for a specific device, all these properties are hashed together and the result is used to bind it to a private per-device key within the form of an X.509 certificate. Such a certificate can then be used for EAP (such as EAP-TLS, EAP-TTLS, PEAP, etc.) to authenticate a device before that any user authentication takes place. However, this approach mandates several modifications concerning the communication procedures between the AP and the authentication server. More in detail, all APs must act as supplicants at the boot step (i.e. before any IP address is assigned) in order to authenticate themselves to the corresponding server [112]. Moreover, re-key procedures would need manual, human driven, procedures.

As a concluding remark, an 802.1X-based authentication would be possible but it would require changes within the actual environment thus making it not practically feasible. As an example, the device authentication based on manufacturer certificates has already been designed and standardized under the name of *Privacy Key Management (for short, PKM)* protocol [113]. This approach exploits X.509 digital certificates and the RSA encryption algorithm that binds public keys to MAC addresses. However, this approach showed to suffer from different issues and to not be straightforwardly applicable to low-cost pervasive devices like sensors [114];

- **Trusted Computing Platforms**: a different approach is the one based on custom hardware elements that, by design, enable the secure storage of sensitive information such as private keys. These specialized elements go under the name of trusted computing bases (for short, TCBs) and are aimed at providing trusted computing platforms (for short, TCPs). In the last years, manufacturers from both the hardware and the software world started to cooperate on solutions based on TCBs aimed at identify and authenticate devices thus forming the non-profit Trusted Computing Group (for short, TCG). The main goal of the TCG is to design and develop trusted platforms by exploiting Trusted Platform Modules (for short, TPM) [115]. The specification defined by the TCG states that TCPs are computing platforms with the embedded property of trust, i.e. they provide a secure and reliable way to verify that data stored within them has not being altered over time.

  In the context of device to device authentication, TCPs can provide a

Solution	Type	Feasibility	Scalability	Heterogeneous
IEEE 802.1X [110]	Infrastructure	Moderate	High	Mostly
IEEE 802.16 [116]	Infrastructure	Moderate	High	Partly
TPMs [115]	Infrastructure	Moderate	High	Mostly
Smart Cards [117]	Infrastructure	Fair	Moderate	Partly
Location-based [118]	Infrastructure	Moderate	Moderate	Partly
Signature-based [119]	Infrastructure	Moderate	Moderate	Partly
AKE [120]	Infrastructure	High	Fair	Partly

Table 4.1: Offline device to device solutions

great contribution to the final goal of *self-authenticated* networks where each device by itself is able to interact with other devices in a secure and autonomous way. As a result, devices would have to authenticate themselves before joining a network and malicious devices could be repelled from the network. Nevertheless, TPM security level usually depends on design and implementation details which are sometimes not shared among all the manufacturers or not clear thus creating an heterogeneous ecosystem. Furthermore, the major concern about TPMs is that they were not designed against physical attacks. They initially served as supports for remote attestation protocols but then showed to be threaten by physical advanced attacks (see Section 4.4). In the IoT, physical attacks will be practically easy to unleash thus lowering the chances for TPMs to play a key role in the design and development of device to device authentication protocols.

Other solutions have been proposed in last years aimed at solving directly or indirectly the problem of device to device authentication and the features of the most known ones are compared in Table 4.1. TPM-based solutions provide a safer way to store private and sensitive information. However, in addition to the problem of physical attacks, they were never meant to provide an identity. TPMs are recognized and identified by checking their private keys. Those keys can be stolen and copied to other TPMs thus making the latter able to mimic the victim's identity.

Other approaches are based on device fingerprinting (see Section 2.4). As an example it has been shown that network devices tend to have constant clock skews that can be exploited to distinguish them through their TCP and ICMP timestamps even though their rate is dependent on the experimental environment [121]. Other solutions have been focused on radio frequency

to either enhance wireless authentication [122] or location detection [123]. Network interface cards (for short, NICs) manufacturing imperfections have also been used to derive device identities from transmitted analog signals, as well as smartphone accelerometer idiosyncrasies [124]. Solutions such as timing analysis of 802.11 probe request frames [125], as well as differences in firmwares and device drivers running on IEEE 802.11 compliant devices [126], 802.11 MAC headers [127] and traffic patterns [128] have been also used to track and to authenticate devices based on the device behavior or data structures such as browser configurations [129], logs [130] and installed fonts [131].

Device fingerprint approaches can be roughly classified in:

- **Hardware-Based**: this approaches exploit the hardware behavior. As an example, the steady temperature inside data centers can be used to verify that two arithmetic relations hold with negligible measurement error between servers. Based on this fact, a skew measuring scheme have been proposed showing that clock skews of a remote device can be considered an effective physical characteristic, suitable for device identification purposes [132];

- **Browser-Based**: by analyzing the code of three popular browsers, Nikiforakis [133] discussed relevant techniques that allow websites to track users online without the need of client-side identifiers. This approach proved to be successful in the identification of major and minor browser versions thus being suitable for identification purposes;

- **Audio-Based**: at manufacturing time, subtle imperfections arise in device microphones and speakers, which induce anomalies in produced and received sounds. The solution proposed in [134] showed that not only it is possible to distinguish devices manufactured by different vendors but also the ones that have the same maker and model thus representing an effective tool for identification purposes;

- **Behavior-Based**: the key idea behind behavioral fingerprinting is to recognize devices by their behavior, i.e. based on the way they interact with each other. Solutions based on this approach proved to be feasible and effective. As an example, a real time analysis was conducted over a VoIP network and proved to be able to identify devices with a high accuracy [135];

- **Sensor-Based**: the multitude of sensors on smartphones can be used to construct a reliable hardware fingerprint of the phone. As an example, an implementation has been shown based on analyzing device-specific accelerometer calibration errors. Such a solution proved to be effective in the identification of legitimate users in a remote server [136];

Even though all the above hardware-based and fingerprint-based solutions proved to be effective in some way, they can be threatened by powerful and ubiquitous attackers as shown in the next section.

4.2 Online Threat Model

As already introduced in Chapter 1.2, three main attacks can thwart the IoT such as capture, disrupt and manipulate. All of them require the attacker to have physical access to the victim's device and then seem to be more feasible for public environments that for the private ones as the latter are usually guarded and physically kept secure by the owner. As a toy example, smart things participating within a smart home application may be securely locked inside the house thus avoiding unauthorized accesses from the outside. However, in an online scenario, those securely locked things have access to the network and have their services exposed to the network.

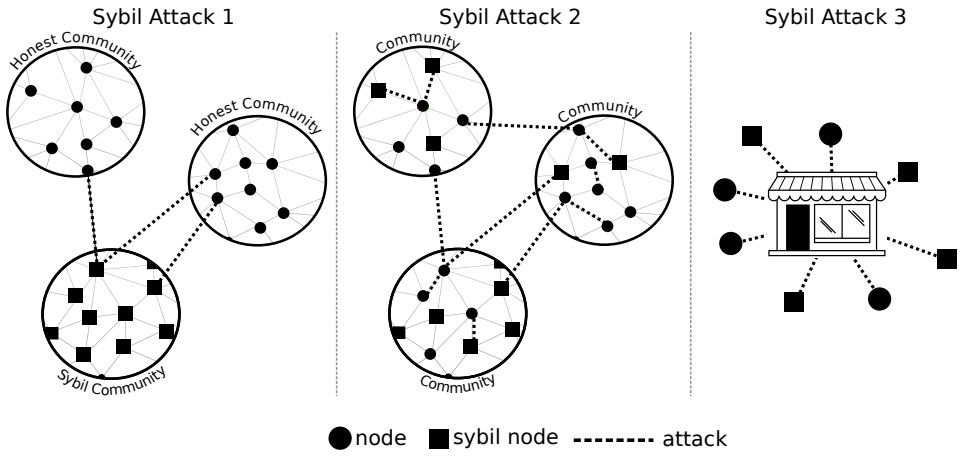

Figure 4.2: Sybil Attacks

Sybil Attacks

In this thesis we assume the attacker to have both physical and remote access to all the smart things needed for the attack to succeed. Such attacker can then infect as many things as needed and turn them into *sybil things*, i.e. malicious things. Based on [33], three different sybil attackers can be defined (all of them depicted in Figure 4.2):

- **Sybil Attacker 1**: (for short, SA-1) this attacker usually builds relationships within the sybil community where sybil things connect with other sybil things. SA-1's capability of building relationships with honest nodes is not strong. In other words, the number of relationships between Sybil nodes and honest ones is limited. As an example, the number of SA-1 attack edges is limited or it is targeted to a specific community and cannot break out to the others.

  SA-1 attackers usually exist in sensing domains and social domains such as OSN voting [137], or mobile sensing systems [138]. The main goal of this attacker is to manipulate the overall portion of the victim's community. In our context-aware attribute-based solution described in Chapter 3 SA-1 can forge fake transactions and indirectly create blockchain forks aimed at changing the aggregated data. In this attack scenario, specially when considering physical capabilities of eavesdropping on IoT things, the behaviors of sybil things are indistinguishable from the normal ones;

- **Sybil Attacker 2**: (for short, SA-2) this attackers usually exist in social domain. Unlike SA-1, SA-2 is able to build relationships not only among sybil things but also with normal unchanged things. In other words, the capability of SA-2 is strong to mimic the normal IoT context structures. Therefore, the number of attack edges is large. The goal of SA-2 is to disseminate spam, advertisements, and malware; steal and violate user's privacy; and maliciously manipulate the reputation system. The behaviors of SA-2 compared to the normal ones can be modeled as a Markov chain [139];

- **Sybil Attacker 3**: (for short, SA-3) this attacker has the same primary goal of SA-2. However, the impact of SA-3 may be in a local area or within a short period thus making SA-3 also capable of thwart mobile networks or highly dynamic networks such as the IoT. Due to the dynamism of mobile networks, mobile things do not usually keep

connections with others for long time. Furthermore, centralized authorities cannot exist in mobile networks at all the time or they might be completely absent as in our blockchain-based approach. Thus, social relationships, global social structures, topologies, and historical behavior patterns in mobile networks are not easy to obtain in order to mitigate and fight against SA-3 attacks.

4.3 Context-aware Tokens

As already described in Section 3.2, attribute nodes could be stored and located within dedicated things but fostering billions of things in the future, this approach will not scale. The other solution is to design ANodes as virtual machines that live within those TNodes that have enough power and storage capabilities to make them run as needed. Hence, CONNECT has been designed as a three-dimensional structure composed by overlapped layers (see Figure 3.4). On the ground we have the physical layer where all the things and their relationships are represented as a graph. The other layers on top of the physical layer, define all the thing attributes and each one is represented by a distinct graph where each ANode is a smart contract and an edge between two ANodes represent an interaction between smart contracts. All the ANodes belonging to the attribute layers are stored by one or more TNodes and all the communications between them are practically accomplished as described in Section 3.2. Furthermore, all the information about the actual and past attribute values is stored, for each attribute, within a different blockchain. Hence, as for ANodes, subsets of each blockchain are stored within those TNodes with enough resources. As a toy example, Figure 4.3 shows how ANodes as well as blockchains are distributed among four different machines. As depicted in the figure, whilst the number of physical devices (i.e. the machines) remain the same, the number of nodes owned by each of them changes over time.

Unlike standard blockchain-based approaches, the approach adopted in CONNECT is not based on proof of work but rather on a *proof of knowledge* algorithm that can immediately and easily validate transactions. Basically, CONNECT authentication protocol is based on a trust continuity factor and is composed of two steps:

- **Initial Authentication**: this step is assumed to be accomplished during TNodes' first boot. As an example, passwords or PIN codes can be used in this step to add a new TNode into an existing network.

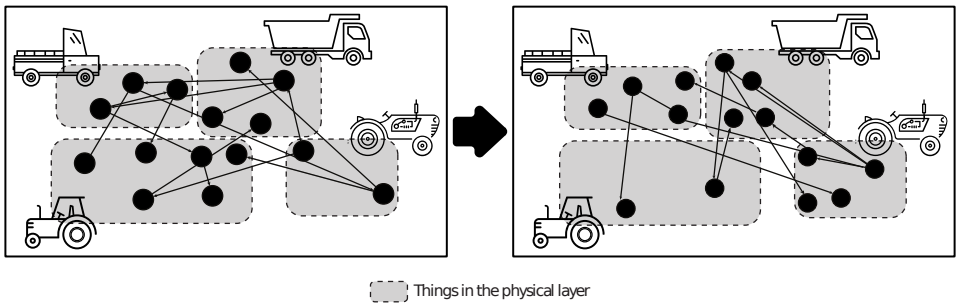

Figure 4.3: Blockchain distribution

This is a manual process but it is accomplished only once during the whole TNode life cycle;

- **Continuous Authentication**: this step is accomplished for each message and it is based on the evolution of each blockchain over time. In this step, TNodes prove their authenticity within the network by providing a witness of their presence over time.

The initial authentication can be seen as a registration step whilst the continuous authentication is the process that ensures the authenticity of communications. The core novelty of such a continuous authentication is to provide a seamless but yet persistent trustworthiness within the network thanks to the high number of interactions between TNodes. In fact, storing TNode attribute changes and interactions within a blockchain makes it continuously and frequently different. The result is what we call *trust continuity factor* and is based on TNode activities, where a TNode is said to be *active* if it is able to send, receive and validate transactions.

The trust continuity factor is built within a *knowledge token* that TNodes have to append to each message in order to prove their trustworthiness. If the knowledge token is valid, it contains the current attribute values for TNodes in the surrounding environment (i.e. TNodes seen as neighbors from the sender and the receiver). A toy example is given in Figure 4.4 where the TNode A needs to authenticate to the TNode B. As depicted, the knowledge token is built on top of information collected about neighbors' attributes as well as attributes that belong to TNodes A and B. More in detail (as shown in Figure 4.4) each knowledge token is composed by:

- **Neighbor Attributes**: they witness TNode A knowledge about the surrounding environment (also known as context);
- **Sender Attributes**: they prove the actual state of the sender;
- **Receiver Attributes**: they prove the state of TNode B as seen by TNode A. If these attributes match with the real ones, this proves that A was able to receive all the messages sent by B to any other TNode.

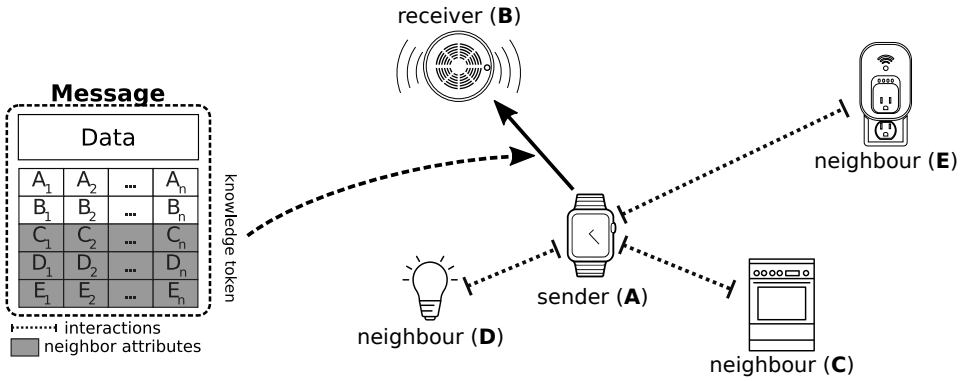

Figure 4.4: Knowledge token used as a trust continuity factor

As each TNode attribute can be changed if and only if it is first validated by the network and added to the blockchain, if A proves to have the latest value for each one of the attributes used in the proof of knowledge, then A is providing a trust continuity factor showing that it was there at each attribute change. Therefore, the TNode A is proving its activity and authenticity at time $i-1$ that finally will allow it to authenticate at time i. Thus, as a general rule, for any $i \in \mathbb{N}$, if a TNode A is able to prove that it was active within the network from t_i to t_{i+1} by providing a knowledge token to TNode B, if A was trusted by B at time t_i, then A can be trusted by B at time t_{i+1}. This process goes back to the beginning of TNodes life cycle (i.e. $i = 0$) when they are physically paired in the initial authentication as already described.

The number of attributes used for the proof of knowledge can vary depending on the storage capability of both the sender and the receiver. The minimum requirement is to provide sender's attributes and receiver's attributes but, the more the number of neighbors' attributes, the less the additional factors required to the TNode A to be authenticated.

Albeit the IoT will provide a pervasive online environment, TNodes may still suddenly go offline for maintenance, crashes or even due to user requests. This might jeopardize the authentication approach used in CONNECT by exploiting the interruption in communication and thus in the trust continuity factor. Hence, to build an Internet of Trust, an additional approach has been designed that provides authentication even for offline scenarios (see Chapter 5).

Security & Privacy Analysis

The major concern usually related to the blockchain technology is about data privacy [140]. In fact, the democratic consensus algorithm embedded in blockchains requires each involved peer to collect, analyze and validate transactions. This can thwart the privacy of those companies that do not want to share their private transactions with others. As such, it is of paramount importance to define privacy and security policies within the blockchain. As an example, it must be defined which thing is able to reach other things and when this can be done.

Unlike standard blockchain-based approaches, CONNECT leverages multiple blockchains thus being able to work with two different privacy aspects:

- **Message Privacy**: a TNode is able to hear transactions exchanged in the network but it cannot understand the content of those transactions. Message privacy can be easily accomplished with encryption. Only those TNodes who have the right key can see the content;
- **Blockchain Privacy**: a TNode can hear only the allowed transactions. As blockchains are based on a peer-to-peer paradigm, asynchronous TNodes need to download new data from other TNodes. Thus, the blockchain privacy is obtained by the authentication protocol.

CONNECT is able to fulfill both aforementioned privacy requirements. In fact, regardless of the routing protocol (see Chapter 3.2) each hop involved in the communication can be encrypted and authenticated. This means that a TNode A willing to access updated attribute values first has to connect to the TNode B hosting those values. This communication requires A to authenticate to B and their communication to be encrypted. Furthermore, it is important to highlight that this encryption and authentication approach can

also fit power constrained things, such as light bulbs, thanks to lightweight cryptographic protocols [141].

As already introduced, TNodes can suddenly go offline. This might seems to jeopardize CONNECT security. However this is mitigated thanks to our trust continuity approach or by fully offline solutions (see Chapter 5). Figure 4.5 depicts all three possible scenarios where a TNode gets temporarily disconnected from the network. In the picture, both valid and malicious transactions are depicted. On one hand, a *valid transaction*, $Tx(A, i)$, is a transaction that has been received and validated by the majority of TNodes in the network. In $Tx(A, i)$, A is a TNode and i is the i^{th} transaction sent by A. Once validated, this transaction is written into the next block and appended to the blockchain. On the other hand, an *invalid transaction*, $T'x(A, i)$, is a fake transaction forged by an attacker. Its final goal is to authenticate a sybil TNode in the network. This transaction can be either accepted or rejected as any other transaction. In $T'x(A, i)$, A represents the victim TNode while i is a counter for the number of transactions sent so far. Valid and invalid transactions within an environment that allows temporarily disconnections of TNodes can lead to three possible scenarios:

- **Use case A**: a TNode disconnects from the network at time $t1$ and reconnects at time $t3$. It now wants to be authenticated back to the network even though it has missed all the transactions being spent at the time $t2$. However, even though the knowledge token provided by A is outdated to time $t1$, other TNodes in the network will browse the blockchain and check that the last interaction of A to the network stopped at time $t1$. This will prove that A was trusted till $t1$ and that it gets disconnected from $t2$ to $t3$. As such, $t3$ will be considered as the t_{i+1} from an online activity perspective thus allowing other TNodes to trust A thanks to its trust continuity factor;

- **Use case B**: in this scenario there is not only a disconnection of a TNode but there is also an adversary trying to gain access to the network. It is assumed that an adversary is able to dump the content of the victim TNode and to use it in order to authenticate. The attack is accomplished between $t1$ and $t2$ and the dump is then used as the trust continuity factor at time $t3$. However, the victim is still connected and continues to generate transactions to the network. In this scenario (as depicted in Figure 4.5) it is assumed that a transaction sent by the victim gets authenticated by the network at time $t2$. As such, the proof

of knowledge stolen by the attacker at time $t1$ and used at time $t3$ will be outdated and rejected by the network. Despite the first scenario, here the attack is mitigated as there is no consistency between the proof of knowledge and the activity of the TNode;

- **Use case C**: in this scenario there is still a malicious TNode but this time the stolen proof of knowledge is used immediately. In this case (as depicted in Figure 4.5) the attacker will success in the authentication and will get access to the network at time $t3$. Furthermore, the victim will be rejected at time $t3$ as its proof of knowledge will not be consistent with the behavior of the attacker. In this extreme case the victim will need to fallback to additional identity factors such as PIN codes etc. By using such factors the victim will force the network in rollback all the transactions being executed by the attacker. The result will be for the attacker to be logged out and for the victim to be logged in.

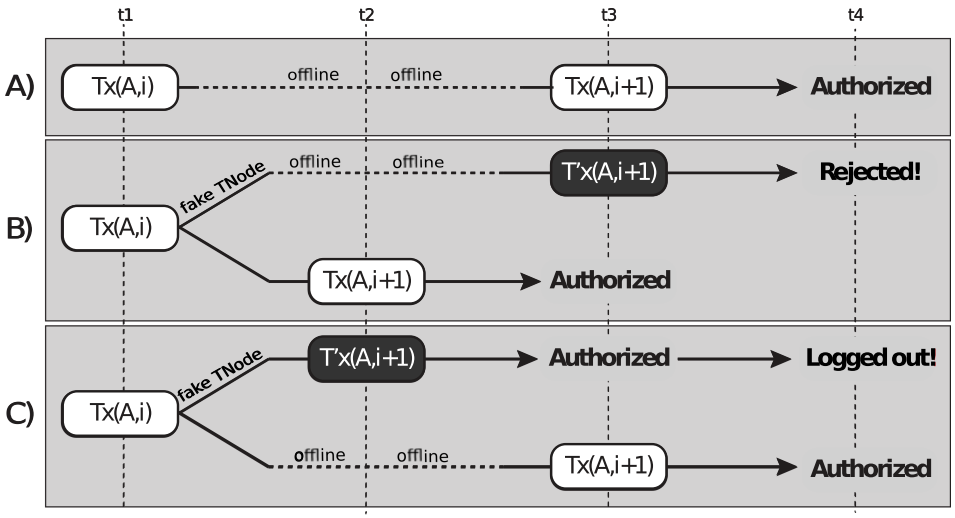

Figure 4.5: Offline attacks scenarios

A final consideration regarding malicious TNodes within the network that are not able to unleash persistent attacks. In this case attackers will only be able to access the network as long as the valid user will remain disconnected. However, in the upcoming Internet of Everything such offline activities will tend to be short and in those limited time windows, fully offline solutions can be exploited to make the above attacks harmless.

4.4 Software Unclonable Functions

In this section we are going to introduce another online authentication solution aimed at solving the challenges remained open in CONNECT. In particular, CONNECT showed to be vulnerable against SA-1 and SA-2 sybil attacks (see Chapter 4 for details). In fact, the main vulnerability in CONNECT is to be only based on the surrounding context rather than including also additional information about the digital identity of the client. This means that, a powerful enough attacker able to reproduce or to compromise the context may also be able to deviate the network for malicious purposes.

The solution proposed here leverages Software-based Unclonable Functions (for short, SUF) to build an additional level of security on top of CONNECT exploiting the synchronization between the client and the server in a cloud environment. The final goal is to provide a challenge-response authentication process that, unlike PUFs and device fingerprint approaches, does not rely on any TTP.

SUF provides an authentication scheme that is not based on the data secrecy but is rather based on the interaction over time between the client and the server. Furthermore, differently from available device fingerprinting solutions, SUF does not derive only a single identity for each device but it rather exploits per-application meta-data exchanged with the servers in order to build a per-application identity. In fact, it is possible to distinguish between two identical devices even though they are produced by the same manufacturer using the same hardware and software by leveraging the:

- **Application Behavior**: devices are usually targeted to different objectives. As a toy example, people usually own two different smartphones, a personal one and a business one. These two devices will be mostly used in different hours of the day to achieve different tasks. As a result, applications will have different log files and data structures, thus providing distinct data fingerprints;

- **Application Data**: devices might need to upload data for remote computing or to download data for local usage. The way in which such communications are performed creates temporary data within the device. Such data is unique for each system, and can be exploited as well to differentiate with respect to other devices.

Unlike previous solutions, SUF does not rely neither on device sensors nor on static data such as browser characteristics or operating system information. On the contrary, it only relies on the synchronization meta-data and application data. For the latter, various tools can be used such as *sdhash* [142] or *ssdeep* [143] to derive unique features from data (even from a single document) that can be used to build the identity of the applications that are using them.

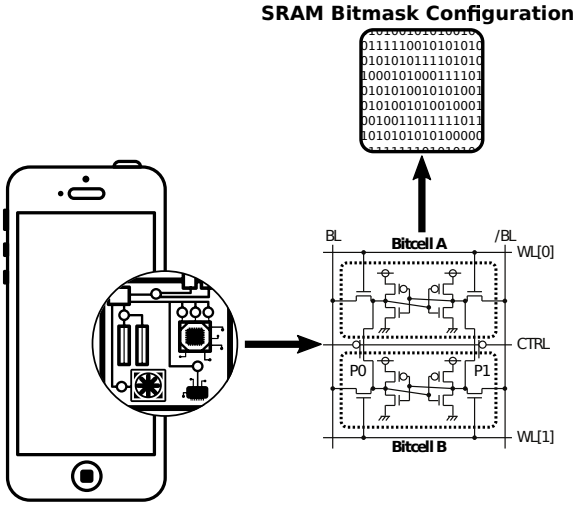

Figure 4.6: Device configuration setup derived from SRAM PUF

In order to better understand how data fingerprint can be used to design software-based unclonable functions, it is important to analyze the architecture of SRAM PUFs. Although SRAM cells are symmetrical, small and random manufacturing variations exist that can cause an intrinsic mismatch that, at boot process, is responsible for cell preferences to either bias towards a logic 0 or towards a logic 1. This cell susceptibility is then exploited to create a device configuration. As shown in Figure 4.6, SRAM cells create a configuration similar to a bitmask where black cells can be seen as logic 1 and white cells can be seen as logic 0. This bitmask can then be XORed to any input given as a challenge in order to compute an output that uniquely identifies the device. Even though this XOR procedure is performed in hardware with the SRAM PUF, the core element for SUF consists in leveraging an ephemeral random bitmask that changes over time and that can be used to uniquely identify the app, the device and ultimately, the user.

SUF uses the same approach based on HIS but at the application level. In

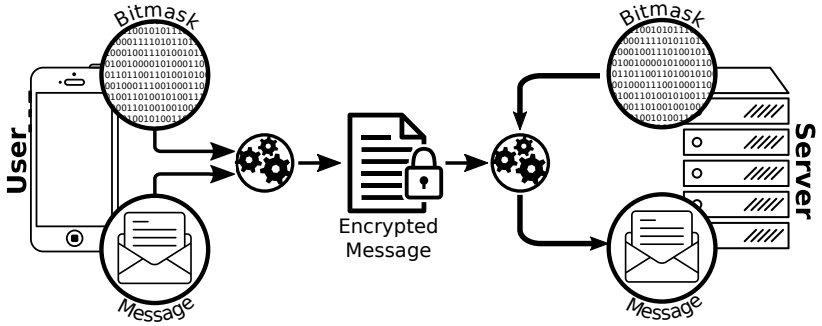

Figure 4.7: SUF communication model

fact, as shown in Figure 4.7, each application leverages its own bitmask to XOR messages that have to be sent to the corresponding service provider. SUF bitmasks are data-sets composed of application data and synchronization meta-data. As a toy example, Dropbox can exploit user data as well as meta-data such as time-stamps, synchronization beacons and all the other information obtained in past communications between the Dropbox server and the user device. Once the XORed message is received by the service provider, this latter just applies the same bitmask again and obtains the plain text of the message.

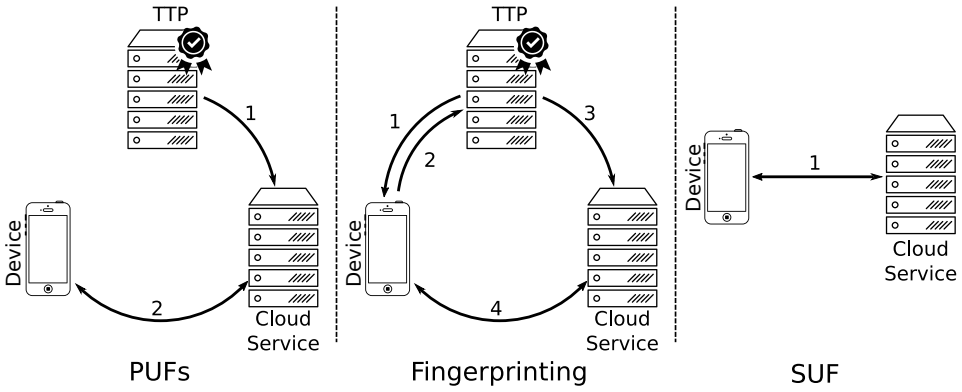

Figure 4.8: Device authentication approaches comparison

The bitmask synchronization is what renders SUFs unique and different from both PUFs and device fingerprint solutions. Further, as shown in Figure 4.8, it also avoids the requirement of a TTP. In fact, in the previous example, the server (e.g. Dropbox) does not need to ask the user or any other TTP for the bitmask. Dropbox already owns the user data and all

the meta-data exchanged in past transactions with that particular client. As such, Dropbox can compute the bitmask by itself and use it to verify messages received by the client, similarly to a pre-shared evolving key or to a symmetric synchronous OTP.

Model

As shown in Figure 4.9, SUF architecture is simply based on a client-server model. On the client side, each client has multiple applications installed and each of them has its own bitmask that is locally stored. On the server side, we have two databases. The first one is used to keep track of all the registered clients whilst the second one is used to keep track of all the client bitmasks. We have depicted such two databases as separated for the sake of clarity to make it clear that the bitmask is something that is computed on top of the client actual data, however the two databases depicted in Figure 4.9 can be stored together.

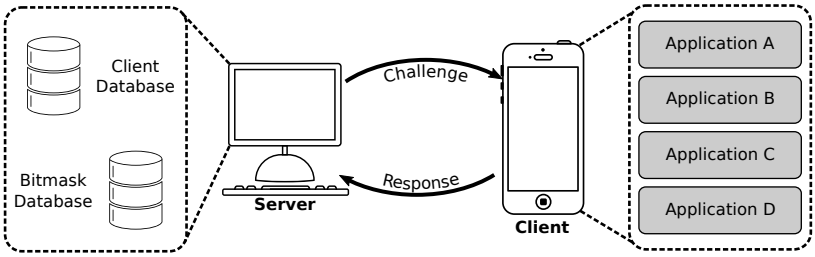

Figure 4.9: SUF main architecture

Security Analysis

It is important to highlight that SUF is an authentication tool based on data exchanged between a client and a server. As such, if the adversary has full-privileged access to the whole client device, he will be able to access all data on the device. He will also be able to intercept all the communications to/from the device, thus being able to create an exact copy of the exchanged data. This would allow him to actually steal the client's identity. Albeit this attack is feasible in the IoT scenario, we assume the attack to be temporary and not persistent otherwise we would end up with an ubiquitous attacker that has complete control of the whole client. Hence we assume ubiquitous and permanent attackers to be unrealistic in our scenario but we allow them to occasionally and temporarily access the clients and steal their content.

We can now better detail the different adversaries we aim to protect from:

- **Passive**: a passive adversary is only able to sniff information exchanged between the client and the server. Against this adversary we assume the PKI or CA scenario as trusted, given that without such an assumption, the adversary might be able to perform a MITM attack thus exploiting data generated from a client to authenticate from another client;
- **Active**: an active adversary is able to gain access to either the client or the server or even to both of them. As such, he could be able to compromise the exchanged data that the client uses to authenticate. However, thanks to the high frequency of exchanged communications between the client and the server, such an adversary will only be able to unleash the attack within the time window comprised between two sequential synchronizations. As a consequence, the more frequent is the interaction between the client and the server, the more secure is the authentication process.

In the case of a passive attacker, because the client is actively being used (such as email synchronization, application update, etc) the data is constantly changed and a clone attack would de-synchronize both the original and malicious clients quickly. To make this attack practically feasible and successful, the attacker has to unleash an advanced persistent threat attack where target device data gets stolen over time. However this attack is considered overkill and not realistic, hence it will not be analyzed in this thesis.

As already mentioned in Chapter 4.2, clients can have offline periods in which no data is exchanged with the server. This might seems to jeopardize SUF security. However, this is mitigated thanks to the unpredictability of the synchronization process (see Section 4.4). Figure 4.10 depicts all three possible scenarios where a client gets temporarily disconnected from the network or simply interrupts the synchronization with the server. The picture is the same already used to describe attacks for the CONNECT authentication protocol and shows both valid and malicious transactions. On one hand, valid transaction $Tx(A, i)$ is a message sent to the server that has been received and validated by the server itself. In $Tx(A, i)$, A is the label of the client and i is the i^{th} transaction sent by A. On the other hand, an invalid transaction $T'x(A, i)$ is a fake transaction forged by an adversary.

Its final goal is to authenticate a fake client, stealing the identity of the real one. This transaction can be either accepted or rejected by the server as any other transaction. In $T'x(A, i)$, A represents the label of the client that the adversary is trying to steal the identity from, and i is a counter for the number of transactions sent so far. We can then have the following three scenarios:

- **Use case A**: the client disconnects from the network at time $t1$ and reconnects at time $t3$. It now requires to be authenticated even though it missed all the transactions sent by the server at the time $t2$. However, even though the data exchanged with the server is outdated to time $t1$, the server is aware of this fact, as no further authentication requests have been done at the time $t2$. Consequently, the server allows the client to log in;
- **Use case B**: in this scenario there is not only a client disconnection but there is also an adversary trying to gain access to the server. It is assumed that an adversary is able to dump the client's exchanged data and to use it in order to authenticate. The attack is accomplished between $t1$ and $t2$ and the dump is then used at time $t3$. However, the client is still connected and continues to synchronize data with the server. As such, by having two clients requesting to be logged in with different data sets provided as their identity, the server could fall back to traditional authentication processes, asking for additional identity factors such as passwords or PIN codes. As the adversary will not have such information, its login request will be dropped as depicted in Figure 4.5;
- **Use case C**: in this scenario there is still a malicious client but this time the dumped data is used immediately. In this case (as depicted in Figure 4.5) the adversary will succeed in the authentication and will get access to the server at time $t3$. However, as soon as the original client will request to be logged in, an inconsistency among the requests will be detected by the server that will again fall back to traditional authentication processes asking the new client to provide additional identity factors such as passwords or PIN codes. This time, unlike the use case (B), the original client will be able to provide such information and, as a result, it will be logged in whilst the fake client will be immediately logged out.

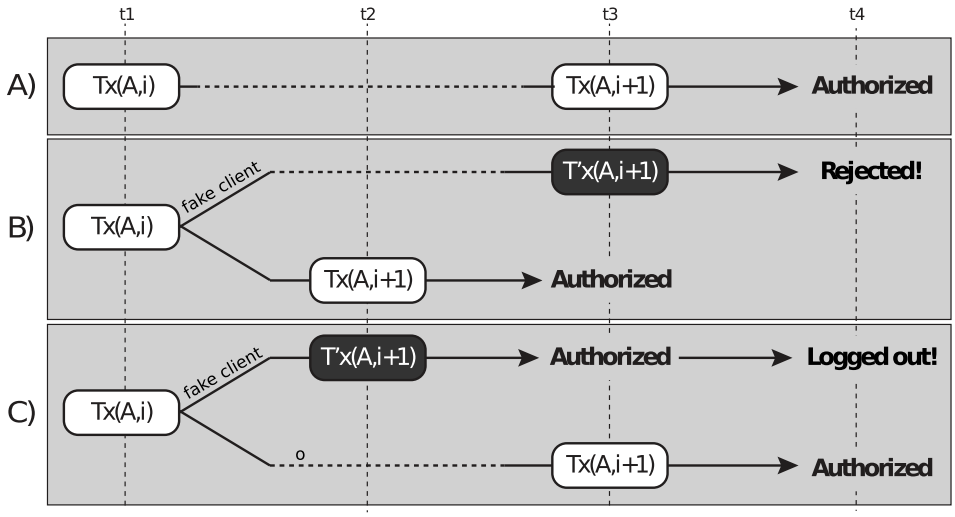

Figure 4.10: Offline attacks mitigation

Bitmask Quality

As introduced above, the security of our Software-based Unclonable Functions approach is not based on data secrecy but rather on the unpredictability and uniqueness of the bitmask. Hence, an analysis of the bitmask unpredictability can help to evaluate its quality (i.e. how much the bitmask is capable of uniquely and securely identify the client). As a toy example, we describe the use case of a cloud storage service (i.e. Dropbox or Google Drive). The randomness of the content of any folder shared and stored in the cloud is somewhat proportional to the randomness of the behavior of the applications that access/modify that folder. However, if we take into account an attacker that is able to physically access the client, the content of the bitmask can be stolen and misused by the attacker.

It is then important to notice that SUF bitmask quality is not based on the content (i.e. on the data secrecy), but rather on the timing of the synchronization process. This fact gives us an indirect measure of the randomness. In a client-server system where the client can autonomously trigger the synchronization process, the leading source of randomness is composed by environmental parameters (e.g. HD sector layout [144] or memory access time [145]) that could be exploited to render the client synchronization activity unpredictable enough to discourage an external attacker. The memory access time is usually orders of magnitude faster than the CPU wake

up time from low-power sleep states [146], that are widely used in modern mobile devices together with tick-less kernels [147]. However, RAM access randomness measured as the number of CPU cycles needed to accomplish I/O operations is still orders of magnitude bellow the randomness that can be extracted from network I/O latency as shown in Figure 4.11. In fact, network I/O is usually the combination of many different parameters such as (a) local CPU wake up time from sleep states, (b) local CPU load, (c) network access time, (d) network load, (e) routing tables load, etc.

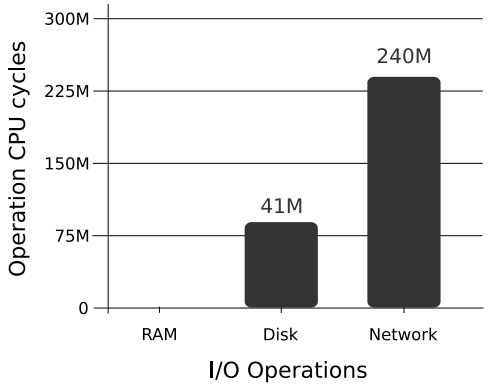

Figure 4.11: I/O tasks CPU cycles

Such network latency can be exploited by both the client and the server that can trigger synchronizations in two different ways:

- **PUSH**: the server sends synchronization commands to the client. This can happen either when the client is offline or online;
- **PULL**: the client sends messages to the server.

The bitmask quality is derived from its randomness, i.e. from the latency of the operations involved in the synchronization process. As such, it is important to measure/evaluate if the message exchange interaction between the client and the server has enough latency to be almost impossible for an external attacker to predict it. However it is also important to note that the same randomness exploited here to create unpredictable synchronizations may result in duplicate messages. Hence, a tunable bitmask size has to be adopted. Power constrained clients, such as light bulb in the IoT, would then be forced to drop duplicated messages whilst more powerful clients, such as smartphone, would leverage on bigger and duplicate-free bitmasks.

The best example of such latency analysis is the one conducted for the Google Cloud Messaging (for short, GCM), the default push messaging solution for the Android platform [148]. GCM is a service which allows to send push messages to Android devices from the server and it is able to handle the queuing of the messages as well as the delivery of those messages to the target applications on the devices. As depicted in Figure 4.12 and in Figure 4.13, the study conducted by Yilmatz et al. showed that both online and offline push notifications had a high arrival unpredictability within a specific time window [149]. Both online and offline messages (even though with different percentages) are shown very unpredictable in their arrival starting from the 10^{th} second up to the 15^{th} minute. This proves they are exploitable to obtain a high level of randomness in the synchronization process. In fact, if both the client and the server use the network time protocol, they could use only those messages received in the above time window to update their bitmask thus rendering bitmask changes unpredictable for the attacker.

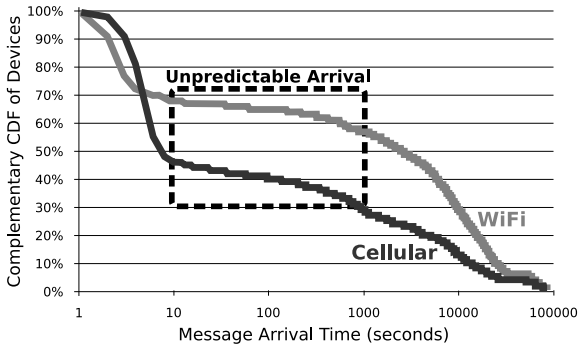

Figure 4.12: Offline message unpredictability

Results obtained both in Figure 4.12 and in Figure 4.13 were obtained in a real world online game scenario and at a reasonable scale involving thousands of real users. However, in a cloud-based scenario where no human interaction is required (as the one proposed with SUF synchronizations) the message rate and the time interval can be increased in order to obtain a higher percentage of affected devices and a wider time window with the final goal of an even better randomness level.

Attack Mitigation

Taking into account the weaknesses of PUF and device fingerprint approaches already described and the comparison summarized in Table 4.2, our Software-

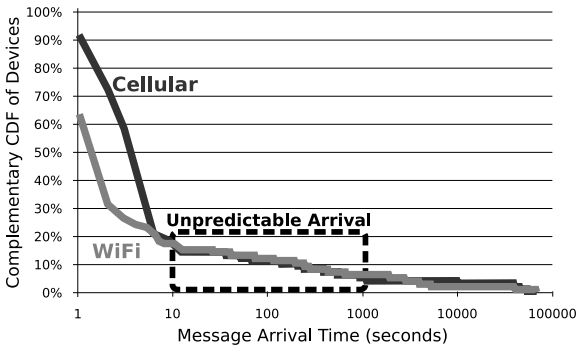

Figure 4.13: Online message unpredictability

based Unclonable Functions approach has multifold advantages:

- **Unclonable**: bitmasks are based on user and application behaviors. More in detail, each time an action is triggered by the user, by the app or by the remote cloud service, user data and meta-data are synchronized between the client and the server. Bitmasks change at each and every of the above synchronizations and their frequencies can be tuned as required. This means that, albeit cloning bitmasks from the user device is still feasible, such a clone will be valid only until the next synchronization takes place. As a consequence, if the communication between the client and the server does not experience long lasting offline periods, cloning the bitmask is practically useless and harmless;

- **Unpredictable**: whilst cloning bitmasks is feasible, even though harmless, thanks to the network latency and to the randomness in the synchronization, predicting the bitmasks is unfeasible. In fact, bitmasks are updated based on the interactions between the user device and the service provider. As such, synchronizations can be triggered manually or randomly by both the user device and the service provider. Thus, it is not practically feasible for an adversary to predict when a new communication or synchronization will be triggered and which data will be exchanged;

- **Unique**: as described in the previous point, due to the unpredictability of both user and service provider behaviors and thanks to the network latency each bitmask is unique per device and per application;

- **Resilient**: PUFs and device fingerprint techniques need to be resilient against physical attacks as they are based on data secrecy. However, PUFs can benefit from special hardware properties, whereas device fingerprint techniques are based on the behavior and the characterization of the device, and their data is physically accessible from within the device. This affects the security of device fingerprint solutions and imposes an additional requirement, i.e. the existence of a TTP. With SUF there is no data secrecy and no TTP is involved. In fact, all data stored within the user device is also synchronized with the service provider. As such, any attempt to tamper with the user device causes changes in the bitmasks. Such changes are detected and recovered when the next synchronization with the service provider takes place;

- **Live**: unlike PUFs, SUF can be upgraded on the field as no hardware element is involved in the architecture and each operation over user data is always synchronized with the service provider;

- **Stable**: PUFs need a fuzzy extractor to make their output stable. When using a SUF, everything is digital and there is no noise in the output thus avoiding fuzzy extractors;

- **Scalable**: unlike PUFs which have a limited number of outputs and configurations depending on the underlying hardware, SUF can define as many bitmasks as needed. Further, the bitmask size can be freely chosen in accordance with the service provider;

- **Cheap**: as no hardware is required for a SUF to work, no additional cost is charged to the user;

- **Seamless**: differently from device fingerprint solutions, the identity factor exploited by SUF is intrinsically contained within user data and meta-data. As such, neither setup processes nor manual operations are required;

- **Distributed**: both PUFs and device fingerprint solutions are based on local data. As such, if those data get stolen, the identity of the client has to be re-built by the TTP or by the client manufacturer. In the SUF approach, given that data are synchronized with the service provider, bitmasks can always be rebuilt on the field by simply authenticating with additional factors;

Feature	PUF	Fingerprint	SUF
Unclonable	✓	✗	✓
Unpredictable	✓	✗	✓
Unique	✓	✗	✓
Resilient	✓	✗	✓
Live	✗	✓	✓
Stable	✗	✓	✓
Scalable	✗	✓	✓
Cheap	✗	✓	✓
Seamless	✗	✗	✓
Distributed	✗	✗	✓
Synchronized	✗	✗	✓

Table 4.2: Device authentication features comparison

- **Synchronized**: the synchronization of bitmasks with the service provider is the core element that mitigates the need of a TTP. In fact, differently from PUFs or device fingerprint techniques, bitmasks are built and stored from the server side and do not need any further authenticity or integrity validation.

Chapter 5

IoT Offline Authentication

5.1 Related Work

Hardware-based security solutions proved in last few years to be a promising approach in solving machine to machine offline security challenges thanks to their very low area and energy requirements and their resiliency properties against side-channel and physical attacks. However, they still suffer from some vulnerabilities as they use to store identity factors within the hardware that is then challenged only during the authentication process. Usually, such private and sensitive identity factors are encrypted in order to avoid any malicious user to read them. Nonetheless, they need to be readable, and locally available, during the authentication as they cannot be downloaded from the cloud in an offline scenario. This means that identity factors have to be stored and protected within the device itself. This, however, makes them easy to be stolen, specially in the IoT scenario where devices can be captured by malicious users. To face with this challenge, new approaches have been requested providing the following features:

- **Data Volatility**: identity factors are not stored within user devices but computed on-the-fly as needed. Once used, identity factors are wiped out from devices;
- **Data Identity**: identity factors are strictly tied to the device they are computed from and can not be reconstructed by other devices.

Current authentication solutions based on the storage of identity factors can be roughly divided in two main categories: on-chip and off-chip. Off-chip

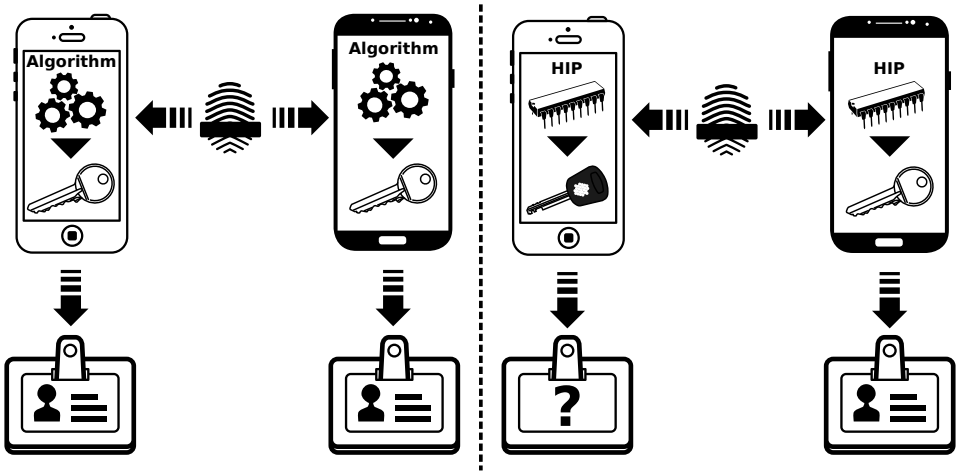

Figure 5.1: Algorithm-based vs HIP-based key reconstruction

identity factor storage mechanisms are the most commonly used and leverage on memory elements, either internal or external, queried by the chip as needed. Such mechanisms suffer from data eavesdropping during the transmission between chip and memories. As such, the solution is to use on-chip storage elements like read-only memories [150], fuse-based mechanisms [151], floating-gate-based mechanisms [152] and battery-backed volatile memory mechanisms [153]. However, all the above approaches can suffer from the following drawbacks:

- **Security**: the permanent storage of identity factors within non-volatile memories can cause them to be stolen by having physical or remote access to the device [33];
- **Cost**: smarter solutions based on more complex elements usually require complex manufacturing processes that raise production costs [154];
- **Production Time**: non standard-technology device components are built on demand. As such, the request for such particular devices may cause significant production delays;
- **Flexibility**: the majority of hardware-based solutions such as ROM [150], EPROM [155] and also fuse-based memories [151] are not upgradeable in the field;

- **Reliability**: battery-based devices [153] have power constraints due to lifetime, temperature variations, shocks and external stresses. Furthermore, as soon as the battery is damaged or over, identity factors might get lost. Flash memories [156], on the other hand, have reliability problems at high temperatures due to charge leakage.

As a general concern, the majority of solutions based on the storage of identity factors all suffer from data extraction. Common examples of tools used to steal identity factors from devices are: scanning electron [157], laser scanning [158], confocal microscopes [159] and focused ion beams [160]. With such tools, information can be glimpsed through the device thus breaking the system security. It is therefore necessary to establish new affordable, but effective, security schemes not only based on key secrecy.

A new technology called hardware intrinsic property (see Section 2.3) proved to solve the above issues. The main difference from HIP and classical software-based approaches is that, as depicted in Figure 5.1, identity factors are computed at run-time by challenging the device instead of by running a software. As such, whilst an attacker in a classical approach could re-execute the same algorithm thus being able to re-create the identity factors within another device, with HIP another device will output different values as shown in Figure 5.1. The result is that the only way to re-create a device identity factor is to physically steal the device.

The HIP approach makes the authentication process more secure but still it suffers from some vulnerabilities. Its core problem derives from its unpredictability. In fact, by being based on the noise within the hardware layer, each HIP-powered device has unique outputs when challenged with some inputs and these input-output pairs need to be stored within a so called challenge-response database. Devices willing to interact with HIP-powered devices must know their CRDBs in order to verify their responses and then authenticate them. As such, stealing the CRDB results in knowing each input-output pair for a specific device that also means to have access to its identity (as shown in Figure 5.2).

It might then seems that standard authentication protocols and HIP-based authentication protocols suffer from the same vulnerability but it is not as in HIP-powered approaches there is nothing to steal from the target device. However, as shown in Figure 5.2, the attacker could steal the CRDB from the server side (i.e. from within the device that is responsible to authenticate the target device) and reuse it to authenticate towards other devices.

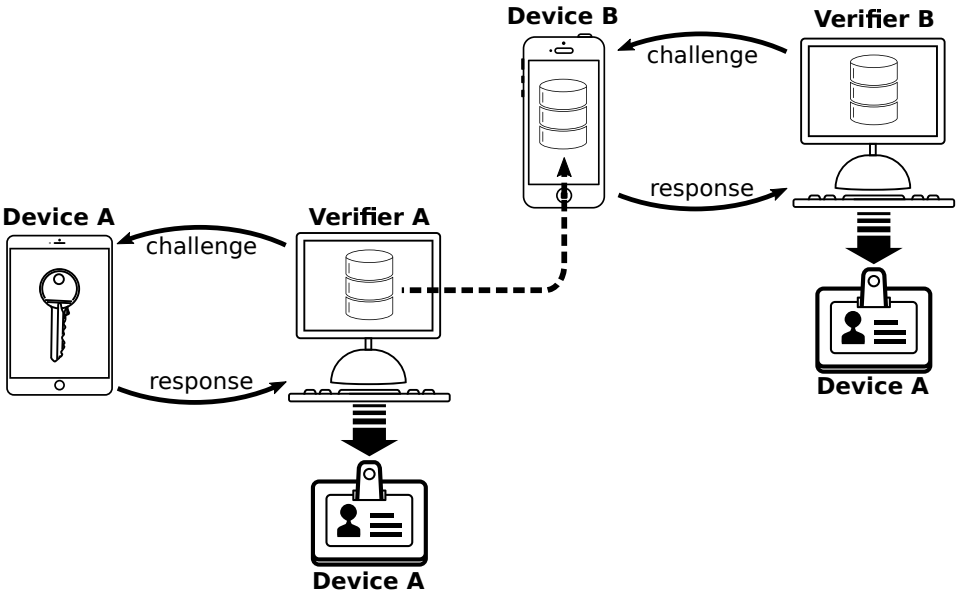

Figure 5.2: Challenge-Response database steal attack

Uniqueness and unclonability properties of HIP approaches make impossible for an adversary to steal private user information and use them to compute responses when challenged by remote services as already shown in Figure 5.1. Hence, by using a HIP approach the adversary is forced to steal both victim credentials and victim's user device in order to pretend to be authenticated as the victim.

The approach obtained with HIP guarantees that the authentication process is not only based on user private credentials, like passwords or biometric information, but is also based on a registered device owned by the user. However, even if such new authentication approach is more secure, it still requires some environment constraints. The main constraint is that, in order to avoid attacks like *stolen CRDB*, CRDBs are required to be stored in remote servers contacted only at the time of user's authentication process (as shown in Figure 5.3).

If a service needs to authenticate a user, such a service requires some CR pairs to the remote CRDB server to check the correctness of user's device responses. As such, at least one device involved in the authentication scheme, has to be assumed to be under some network coverage. This requirement hinder the use of HIPDs in particular scenarios, labeled *critical*, where such

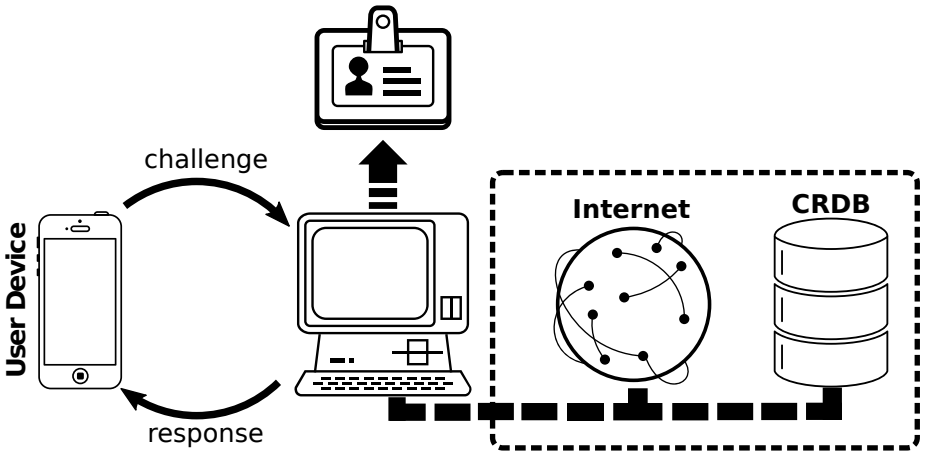

Figure 5.3: CRDB access over internet

network coverage constraint can not be satisfied but, due to the high privacy and sensibility of exchanged data, a secure authentication process is required. During the rest of this thesis we will mainly focus on two critical scenarios extensively studied in recent years, *e-cash systems* and *e-health systems*, which require special attention in the authentication process as they deal with money savings and health.

Offline authentication approaches based on standard digital certificates and PKIs proved in the past to be enough secure if supported by additional technologies such as hardware intrinsic security. However, in the following of this chapter we are going to analyze a special subset of offline authentications where each authentication session needs to be also disposable. Mobile payments and medical prescriptions represent two critical scenarios that can benefit from disposable offline authentications. In fact, it is well known that double spending attacks are the more complex to mitigate and also the double usage of medical prescription might be dangerous if exploited by malicious patients.

5.2 Offline Threat Model

In order to better describe all the possible threats that thwart a fully offline environment, a detailed description of both attacks and attackers is introduced in this section. The first important distinction that has to be made is about the position of the adversary:

Adversary / Device	Channel	Claimant Device (CD)	Verifier Device (VD)
Collector	✔	•	•
Malicious Claimant	✔	✔	•
Malicious Verifier	✔	•	✔
Ubiquitous	✔	✔	✔

Table 5.1: Adversaries classification

- **Internal Attacker**: this adversary is directly involved in the authentication as a claimant or a verifier. As such, he is capable of tweaking both the claimant device (for short, CD) and the verifier device (for short, VD) either by injecting malicious code or by having physical access to it;
- **External Attacker**: this adversary is not directly involved in the authentication. As such, he can only access/alter the data being exchanged between the VD and the CD while in transit.

The second classification is based upon the number of tweaked devices as follows (see Table 5.1):

- **Collector**: this is an external adversary able to eavesdrop and alter messages being exchanged between the CD and the VD but without any access to the devices;
- **Malicious Customer**: this is an internal adversary that can either physically open the CD to eavesdrop sensitive information or inject malicious code within the CD in order to alter its behavior;
- **Malicious Vendor**: is an internal adversary that can either eavesdrop information from the VD or inject malicious code within the VD in order to alter its behavior;
- **Ubiquitous**: this is an internal adversary with complete access to both CD and VD.

In our offline authentication system no restrictions have been made on the capabilities of the adversary, always considered as ubiquitous. In Table 5.2 a grid is depicted considering most relevant types of adversarial models and

Attack / Adversary	Collector	Claimant	Verifier	Ubiquitous
Double Usage	•	✔	•	✔
Forgery	•	✔	•	✔
Data Dump	•	✔	•	✔
Data Poisoning	•	✔	✔	✔
Data Deletion	•	✔	✔	✔
Hardware Emulation	✔	✔	•	✔
Software Emulation	✔	✔	•	✔
Information Stealing	•	•	✔	✔
Reverse Engineering	•	✔	✔	✔
Man In the Middle	✔	✔	✔	✔
Replay	✔	✔	✔	✔
Hardware Modification	•	✔	✔	✔
Hardware Eavesdropping	•	✔	✔	✔

Table 5.2: Offline attacks

attack techniques showing that the greater the ability of an adversary to physically access the devices, the more complex the attacks that can be unleashed. Only a subset of the attacks listed in Table 5.2 represents real dangers in a fully offline scenario. In fact, in such a scenario only the VD and the CD are involved in the protocol and no connection to the external world is provided.

Offline scenarios are harder to protect than the online ones already discussed in Chapter 4. In these cases, claimants' data are kept within the VD for much longer time, thus being more exposed to attackers. In fact, many different ways to exploit VD vulnerabilities and steal claimant's data exist:

- **Skimmers**: in this attack, the claimant input device that belongs to the verifier system is replaced with a fake one in order to capture claimant's inputs. As an example, input devices can be either physically replaced or directly purchased with vulnerable or misconfigured software [161];
- **Scrapers**: in this attack, a malware is installed within the verifier system in order to steal claimant's data. As an example, cyber-criminals can infect the system using phishing attacks. However, in some other cases, the malware is installed with the help of an insider or via a backdoor;

- **Forced offline authorization**: in this scenario, the attacker exploits a DoS attack to force the verifier's system to go offline. By doing so, the attacker will force sensitive claimant's information to be locally processed. This means that any information read from the claimant's device will be locally decrypted and verified, thus creating an opportunity for the attacker to easily collect all the required information [162];

- **Software vulnerabilities**: login protocols and authentication applications themselves are also vulnerable to several attacks. In Application Programming Interface (for short, API) attacks, software bugs are exploited to retrieve sensitive claimant's information. Disassembling techniques are also used either to alter firmwares/software or to replace them with malicious functions.

With respect to verifier system vulnerabilities, there are three specific attacks that have to be analyzed:

- **Data in memory**: malware injected within the verifier's device may be exploited to steal claimant private information while being processed in RAM during the authentication protocol;

- **Data in transit**: data exchanged between the claimant and the verifier might get stolen or eavesdropped during the authentication protocol. Furthermore, sensitive data may also get stolen while in transit from distinct hardware elements within the same device;

- **Data at rest**: data stored within non-volatile memories might get stolen as well. This is the most easy attack to be unleashed as private claimant information are stored for a long time and attackers have all the time they need.

5.3 Offline Disposable Tokens

FORCE is the first solution that neither requires TTPs nor assumptions on trusted devices to mitigate all those attacks that usually affect fully offline disposable authentications. To achieve such a goal, FORCE leverages physically unclonable functions and proposes a novel, fully offline system based on disposable tokens, i.e. precached tokens that can be used only once.

Furthermore, by allowing FORCE claimants to be free from any registration procedure, makes it particularly interesting as regards privacy. In fact, anyone can obtain a FORCE scratch card (e.g. provided by medical doctors in hospitals) without disclosing its real identity. Hence, FORCE disposable tokens can be seen as digital version of paper vouchers and, as such, they are not linked to anybody else than the current holder.

Differently from other solutions, FORCE assumes that only the chips built upon PUFs can exploit their tamper evidence features. As a consequence, the assumptions made in this thesis are much less restrictive and more realistic than the others. FORCE can be applied to any scenario composed by a claimant's device and a verifier's device that, in the remaining of this chapter we will call CD and VD respectively. In its current version, as depicted in Figure 5.4, FORCE has been designed using a smartphone as the CD, a personal computer as the VD and a Near Field Communication channel [163] (for short, NFC) for all the communications between the CD and the VD. The rationale behind the choice of an NFC channel is that it is much easier to use compared to other wireless communication technologies like Bluetooth or WiFi and last but not least, it requires the CD and the VD to be physically close to each other.

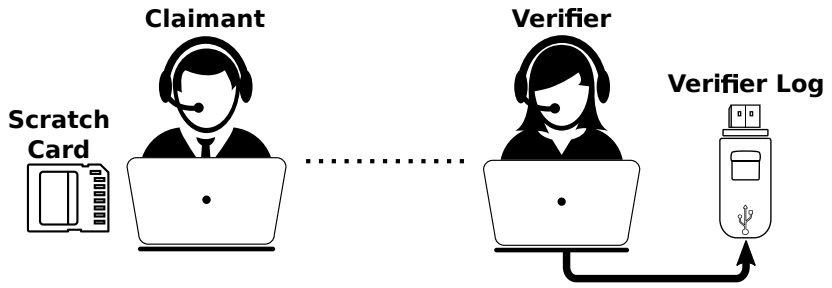

Figure 5.4: FORCE model

In FORCE all the involved devices can be tweaked by attackers and are considered untrusted except for the storage device, that we assume is kept physically secure by the verifier. It is important to highlight that such an assumption does not thwart the security of our solution. In fact, similarly to real vouchers, the storage device is not involved in the authentication protocol but represents a secure and write-only place where logs are stored. Furthermore, rather than being focused on a specific application, FORCE has been designed to be a secure and reliable encapsulation scheme for disposable tokens thus making it applicable to any application that needs non

repeatable authentication procedures (see Chapter 6).

Architecture

The core element of our architecture design is a scratch card that can be built within the CD or used as a separate element, such as Secure Digital cards (for short, SD), USB thumb drives, etc. A scratch card is composed of:

- **Scratch Memory**: it is a special read once memory used to store disposable tokens;
- **Authenticator**: it is used to compute, on-the-fly, all the cryptographic keys required for the authentication protocol;
- **Memory Mapping Unit**: it is used to retrieve disposable tokens' layout and to detect malicious attacks based on the guessing of the memory content.

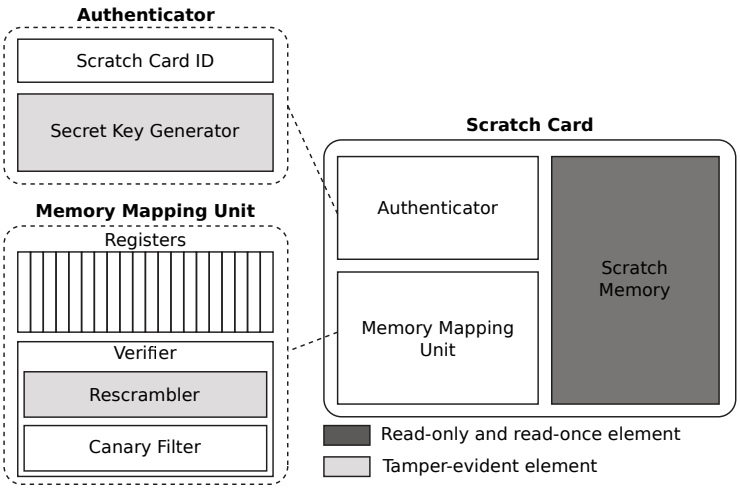

Figure 5.5: FORCE scratch card architecture

Scratch Memory

At the hearth of the scratch card lies a read-once memory[164] named *scratch memory*. Such memory, used to store disposable tokens, has the property that reading one value destroys/erases the original content (see Figure 5.6).

Furthermore, to improve the resiliency of our solution against random guessing attacks of the memory, an additional layer of protection has been added in the form of *canary bits*. These security-wise bits are not linked to disposable tokens but rather serve as security flags to raise alerts if malicious or unattended reads of the memory are demanded. As explained more in details in Section 5.3, canary bits erased the whole memory once accessed, thus making the scratch card useless.

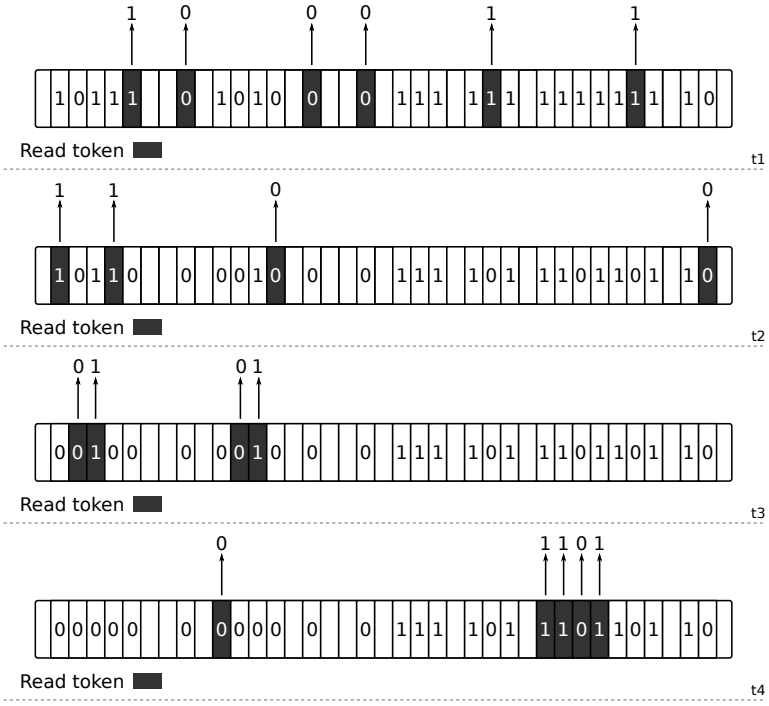

Figure 5.6: FORCE memory read

FORCE scratch memory is not tied/limited to any static format. It just requires each token to be composed of at least two fields, namely the content of the token and an integrity verification value (i.e. a digital signature written by the creator of the token and attesting its authenticity). This last integrity value is used to guarantee that a specific disposable token is created to be used by a designated scratch card only. To mitigate forgery attacks, such value is computed at manufacturing time by first encrypting the token content with the public key of the scratch card and then additionally signing it with the private key of the card issuer. Once a token has been created, it is stored within the scratch memory in a non contiguous way. During

this step, the card issuer creates unique random layouts, one for each token, where unique means that taken two disposable tokens T_a and T_b and given L_a (the layout of T_a) and L_b (the layout of T_b) then $L_a \cap L_b = \emptyset$, $\forall\ (a, b)$ with $a \neq b$. The randomness of disposable token layouts mitigate attacks based on the guessing of the memory. In addition to the independence from a static token's format, FORCE does not rely either on a specific scratch memory size or disposable token numbers. It is the card issuer that has the responsibility of managing the scratch memory layout as regards to both the size of tokens and their number in memory. As such, FORCE can work with scratch memories of any size and containing any number of tokens. It is also important to highlight that the scrambled layout of disposable tokens within the scratch memory is not the core security element of the solution proposed in this thesis. Token layout is only meant to prevent a subset of attacks based on the guessing of the scratch memory whilst other security properties are provided by the system design, the protocol being used and the technology exploited as described more in detail in the remaining of this chapter.

Authenticator

The authenticator element is used in our solution to compute on-the-fly the scratch card private key needed to decrypt verifier requests. Rather than embodying a single cryptographic key within the device, thus potentially allowing an adversary to steal it, PUFs have been exploited in FORCE to implement a strong challenge-response authentication process. The challenge used as input for the PUF is a publicly known scratch card identifier, hard-coded within the card and used in the authentication protocol as the card public key. Each scratch card is indeed shipped with its own public key, signed by the card issuer to avoid forgery attacks and hard-coded into the card itself. This allows the claimant to broadcast the card public key to VDs which are not required to know it in advance.

Verifiers can encrypt authentication requests with the public key of a scratch card thus being sure that such requests can be only read by that card. Further, PUFs tamper-evidence feature ensures that any attempt to open on-the-fly the authenticator element to read the computed private key will alter the behavior of the PUF causing a different key to be produced. Changing the private key leads to the impossibility to read verifier requests and thus renders the whole scratch card useless.

Memory Mapping Unit

The memory mapping unit element (for short, MMU) is composed by a set of disposable token registers and by a filter element as depicted in Figure 5.5. The registers are hard-coded into the MMU and each one is given as input to the rescrambler element in order to compute the actual layout of each disposable token within the scratch memory (see Figure 5.7). Again, token layouts are not stored anywhere within the scratch card but are rather computed on-the-fly each time, making it hard for an adversary to steal them.

The latest component of the MMU is a canary filter used to protect the scratch card from memory guessing attacks by using canary bits. These bits have the main goal of keeping track of scratch memory malicious accesses and, as depicted in Figure 5.7, they are designed as input-output mapping functions. If a bit given as input to the canary filter matches a canary bit, the output is multiplexed to the whole scratch memory. This guarantees that any attempt to read a canary bit will automatically cause the entire scratch memory to be read and, as such, erased. As for the authenticator, the MMU takes advantage of the tamper-evidence feature of its embedded rescrambler which is actually based on a PUF.

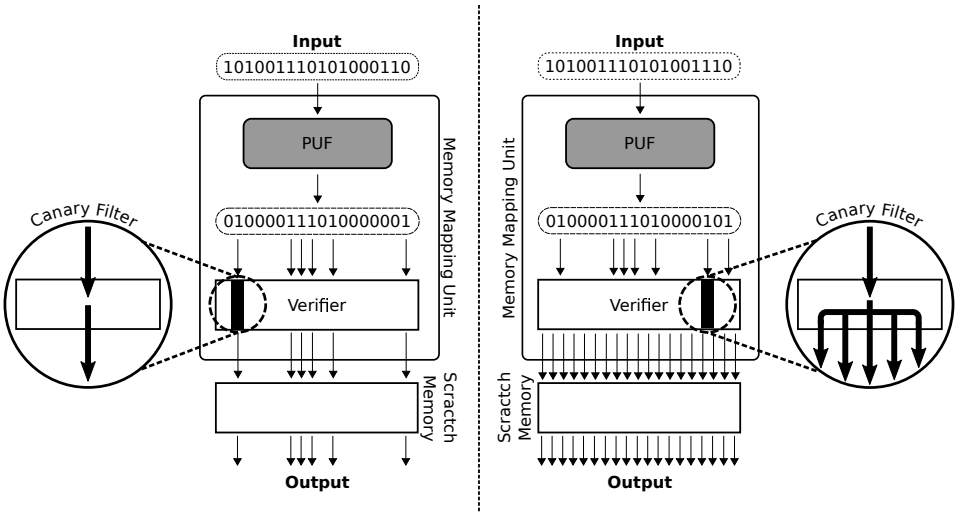

Figure 5.7: Authorized vs malicious scratch card read

Stable PUF extraction

PUFs have been used in FORCE to compute the scratch card private key and the actual layout value of each disposable token. However, given a fixed input, PUFs can produce a responses that is not bitwise identical when regenerated multiple times. As such, in order to use PUFs in algorithms based on stable values, such as for cryptographic keys, an intermediate step is required in order to build a stable output (see Chapter 2). Recently, some solutions have been proposed to correct PUF output on-the-fly thus providing the generation of secret stable values within the device. FORCE leverages on this approach for the design of both the key generator element (embedded in the authenticator) and for the filter element (embedded in the MMU). Such special PUFs are built upon a lightweight error correction algorithm proposed in [165] and depicted in Figure 5.8. The algorithm proposed by Yu et al. is based on top of a 64-sum PUF block that looks at the difference between two delay terms, each produced by the sum of 64 PUF values. Hence, given a challenge, its i^{th} bit called C_i determines, for each of the 64 stages, which PUF is used to compute the top delay term, and which is used to compute the bottom delay term. The sign bit of the difference between the two delay terms determines whether the PUF outputs a '1' or '0' bit for the 64-bit challenge $C_0 \cdots C_{63}$. The remaining bits of the difference determine the confidence level of the '1' or the '0' output bit. The k-sum PUF can be thought of as a k-stage Arbiter PUF [166] with a real-valued output that contains both the output bit as well as its confidence level. This information is used by the downstream lightweight error correction block that is able to produce in output a stable value within the scratch card in one single step.

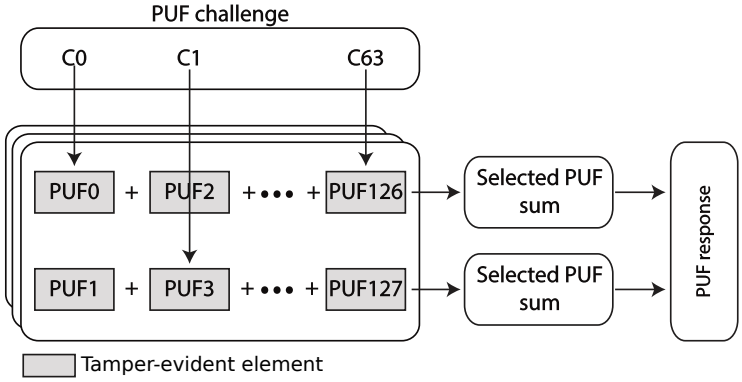

Figure 5.8: Stable PUF-based key reconstruction architecture

By using such on-the-fly stable value generation process, FORCE does not store neither private keys nor disposable tokens within the claimant's device thus protecting them from malicious claimants or outsiders and ensuring that only the right scratch card can retrieve its own information each time it is needed.

Protocol

This section describes the *Pairing*, *Authentication* and *Rollover* steps that belong to FORCE. Furthermore, in Chapter 6 the adoption of our disposable tokens solution for mobile payments is given. In that chapter, we describe how our tokens can be used as digital fully offline credits thus also describing additional protocol steps such as *Redemption* and *Transaction Dispute*.

Pairing Step

FORCE uses the NFC technology for all the communications between CDs and VDs. Even though NFC requires both the involved devices to be very close to each other, an adversary could still be able to unleash man-in-the-middle attacks (for short, MITM) by using NFC boosters. As such, a pairing setup process has being used as the first step in our protocol to physically verify the claimant's device.

For the pairing step, FORCE relies on standard and well known protocols such as the *Passkey Entry* of the Bluetooth Simple Pairing Process (for short, SSP). At the end of this step, both the CD and the VD share their public keys used to guarantee integrity and authenticity of messages being exchanged. Furthermore, in order to avoid brute force attacks in the pairing step, FORCE adopts a *fail-to-ban* approach based upon a failure threshold value. In this case, if a malicious claimant consecutively fails the SSP protocol procedure, the system stops for few seconds (usually 20 or 30 seconds). If the number of consecutive fail-to-ban reaches a security threshold value, the verifier can decide to refuse further authentication requests made by the same claimant.

Authentication Step

FORCE authentication step is composed by the following operations, depicted in Figure 5.9 (see details for the payment application in Chapter 6):

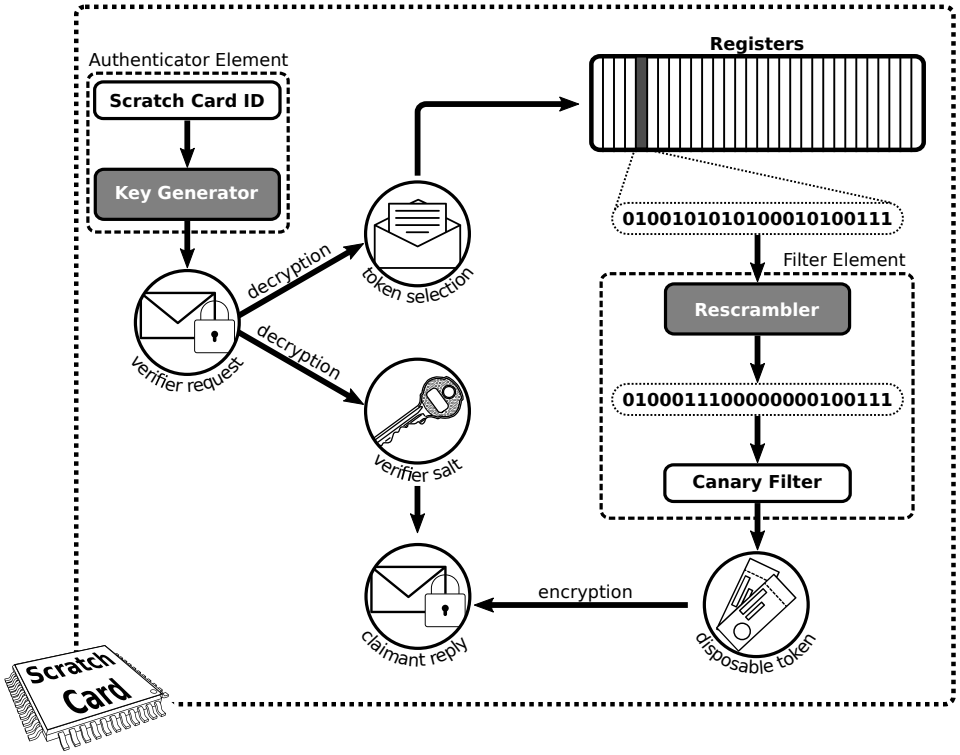

Figure 5.9: Authentication Protocol Overview

1. The CD sends an authentication request to the VD asking to be logged into the system or to be given access to a limited service. Within this request the claimant also embeds the indexes of all the disposable tokens that are still available in its scratch card. If the i^{th} index number is present in this message, it means that the i^{th} register within the scratch card is available to be read in order to retrieve the i^{th} disposable token;

2. Once the verifier receives the above message, it first creates a random salt value. Then, for each disposable token that will be involved in the authentication step, a single request is created by encrypting the token layout together with the random salt. Then, this request is again encrypted with the public key of the scratch card, thus rendering the claimant involved in the actual authentication process the only one able to read it;

3. When the claimant receives the verifier's message, the scratch card private key is computed by the authenticator and it is used to decrypt the message received, thus obtaining both the salt value and the encrypted request computed by the verifier. Values found within the encrypted request are then used to challenge the scratch memory and to read the content of the disposable token. Once the content of the token has been extracted from the read once memory, it is again encrypted twice before it is sent back to the verifier. The first encryption is made with the random salt value received by the VD and its purpose is to guarantee the freshness of the response. Then the second encryption is made with the scratch card private key to guarantee the authenticity and integrity of the response;

4. Once the verifier receives the claimant's reply to its request, it first verifies the digital signature (see Chapter 2 for details) on the message and then it decrypts the message with the random salt created at the beginning of the authentication step. If the signature is verified and if the content of the response is meaningful than the content of the message is validated. If everything is OK, than the claimant is allowed to access the service/device/resource and a new entry is appended in the verifier's log.

If all the above steps are accomplished without errors and the execution flow goes through all the operations depicted in Figure 5.9, then the authentication procedure is accomplished and the claimant is given the required access.

Key Rollover

As for the majority of the real-world PKI-based authentication approaches, FORCE assumes that, in case of card issuer private key renewal, a time-window is adequately chosen to let customers decide whether to use their last disposable tokens or to exchange them with new ones. This standard procedure is widely accepted in the real world and, as such, no custom key rollover protocol has been designed for FORCE.

Security Analysis

In this section the robustness of our solution is discussed. FORCE uses both symmetric and asymmetric cryptographic primitives in order to guarantee the most important security principles as follow:

- **Authenticity**: it is guaranteed by FORCE both in the pairing step that in the authentication step. In the former, the authenticity is ensured by the SSP protocol while in the latter it is ensured by the authenticator element embedded in the scratch card;
- **Non Repudiation**: the storage device kept physically safe by the verifier prevents the adversary from deleting past logs thus avoiding malicious repudiation requests in which a claimant may deny to have already used one or more disposable tokens. Furthermore, the content of the storage device can be exported to an external device (such as a pen drives) on a timely basis, thus making our assumption even more acceptable;
- **Integrity**: disposable tokens' integrity is ensured by signing each token with the private key of the scratch card issuer. Furthermore, message integrity is ensured in FORCE thanks to the on-the-fly computation of the scratch card private key that is never stored within the device but always computed at run-time;
- **Confidentiality**: responses are first encrypted with the random salt generated by the verifier at the beginning of the authentication step and then signed with the scratch card private key. This second step ensures that the response was originated by a specific scratch card while the encryption layer built upon the salt, guarantees confidentiality and freshness of the response generated by the card;
- **Availability**: the fully offline scenario completely removes any type of external communication requirement and makes it possible to use offline disposable tokens also in extreme situations with no network coverage. Furthermore, the implementation with a passive card makes the proposed scratch card able to be used by different devices.

As regards physical security properties provided in FORCE, scratch cards used in our solution share the assumption, as in other solutions based on the same hardware approach [167], that the card is tamper-evident. This assumption is based on the size of nowadays integrated circuits and on the unfeasibility for a casual adversary to open the device and to play with the card without causing an alteration in the PUF behavior. This assumption is no longer valid if an expert adversary with access to highly sophisticated and expensive tools, such as scanning electron microscopes or focused ion beams [160], is taken into account. However, such tools can be worth thousands

of dollars and applying this kind of attack on each single device is assumed here to be overkill and inconvenient for the attacker.

Attack Mitigation

In this section, FORCE resiliency is discussed against all the attacks listed in Table 5.2:

- **Double Usage**: the read once property of the scratch memory prevents an adversary from reading the same token twice. Even if a malicious claimant creates a fake verifier device and reads all the disposable tokens, it will not be able to use them with other verifiers due to the inability in decrypting their messages. In fact, the private key of the scratch card is needed to decrypt the request generated by the verifier and can be obtained only within the claimant device. The fake verifier forged by the malicious claimant could then try to produce or emulate a new scratch card with a private/public key pair. However, the public key of such a card will not be considered a trusted one by other verifiers as not signed by a known card issuer. Thus, any message received by such an unconfirmed scratch card will be immediately rejected;

- **Forgery**: each disposable token is signed with the private key of the card issuer and thus it is not possible for an adversary to forge new tokens;

- **Data Dump**: opening the scratch card to copy the content of the memory will alter the behavior of the PUF, thus making the whole scratch card useless;

- **Data Poisoning**: each completed authentication log entry is kept in the verifier's storage device. If a disposable token has been corrupted by a memory poisoning attack, such token will not be accepted. Such corrupted and unused tokens can be claimed back to the card/token issuer that will check for verifiers' logs. If such token is not present in any of the logs, a new disposable token will be given back to the victim;

- **Data Deletion**: this is a special case of the Memory Poisoning attack in which all disposable tokens are corrupted;

- **Hardware Emulation**: PUFs, by design, cannot be neither forged nor emulated as the responses computed by such fake PUFs will be different from the original ones;
- **Software Emulation**: it is not possible, by design, to emulate PUFs without opening them and, thus, corrupting them. Furthermore, hardware speed is usually faster than software speed. Hence, PUFs are normally designed to be easy to stimulate but complex to emulate thus making the gap in the execution time even higher. This makes the verifier able to use *out-of-time* thresholds to identify emulated scratch cards;
- **Information Stealing**: the private key of the CD and the real layout of each disposable token is computed on-the-fly as needed. No sensitive information is kept in the scratch card;
- **Reverse Engineering**: by design, any attempt to tweak the scratch card in order to steal any useful information alters the behavior of the PUF thus rendering the whole scratch card no longer usable;
- **Man-In-the-Middle**: disposable tokens are signed and shuffled by the card issuer and contain, among all other things, the scratch card ID. As a consequence, an adversary cannot use disposable tokens which belong to other claimants by simply copy them from the scratch card of the victim. Even changing the content of the victim's disposable token by replacing the victim's ID with the ID of the adversary is not possible. After such alteration of the token, the adversary would not be able to sign again the token with the private key of the card issuer, thus rendering the malicious token useless. This can be better understood thinking to PKI scenarios. Changing the ID is like changing the public key thus pretending to be another user. However by having access to the public key will not make us capable to impersonate other users as we will not have their private keys;
- **Replay**: each challenge is different due to the random salt generated by the verifier;
- **HW Modification**: by design, it is not possible for an adversary to add/modify/remove any element belonging to the scratch card without changing its behavior;
- **HW Eavesdropping**: nowadays photon counting APD modules [168] and photon emission microscope with InGaAs image sensors are used

with focused ion beam [160] systems in order to locate faults within integrated circuits. However, as explained at the beginning of this section, we consider this kind of attack overkill.

In general, attacks that try to infer information from a device can be categorized as *passive* or *intrusive*. In *passive* attacks the system interface is probed for either timing or electrical differences. In *intrusive* attacks the adversary is able to breach the physical boundary of the package by scanning, probing or altering the hardware.

In FORCE, on the one hand, intrusive attacks are not feasible as they alter the functionality of the scratch card. On the other hand, passive attacks have been analyzed by subdividing them into *powered* and *unpowered* attacks. In powered attacks the device is monitored while running whilst, in unpowered attacks, information is extracted from the device while the hardware is not powered on. In FORCE no value used by the protocol is permanently stored in the CD. As such, unpowered attacks are mitigated. On the contrary, a run-time attack using extremely complex monitoring tools could have access to the values being computed during each step of the protocol. However, stealing information on-the-fly at run-time is considered overkill in this scenario (as already explained in Section 5.3) thus providing also resiliency against casual run-time attackers.

So far, we have discussed the resiliency of our authentication system against attacks targeting the involved devices and all the messages exchanged between the claimant and the verifier. In the following, other considerations are shared based on the different adversary models introduced in Section 5.2:

- **Malicious Claimant**: as shown at the beginning of this section, forgery, dump and reply attacks are mitigated by the architecture and physical nature of the PUF elements and the read once memory embedded in the scratch card. Hence, even though a malicious claimant has complete access to its scratch card, he cannot play with it without making it invalid;
- **Malicious Verifier**: the only feasible attack for a malicious verifier is the deletion of log entries from the storage device. However, this is not possible as the storage device has been assumed to be append-only and kept physically secure. Furthermore, it makes no sense for a verifier to delete past log entries as they are the only evidence attesting that the

verifier has correctly accomplished the authentication procedure (see Chapter 6 for a practical example in mobile payments);

- **Ubiquitous**: this attacker is the most powerful one. Hence he is assumed to be able to steal information from both the VD and the CD and to reconstruct the semantic of the scratch card memory content. However, in order to steal such information the adversary has to physically tweak the scratch card, thus invalidating it thanks to the PUF tamper-evidence feature.

The robustness of our solution is mainly based on PUF features but also on the high unpredictability of token layout within the scratch memory. As regards physical attacks to PUFs, Integrated Circuits (for short, IC) or other hardware in general, some relevant results are discussed in [169] and [167]. The first one aims at protecting IC integrity as each manufactured IC is rendered inoperative unless a unique per-chip unlocking key is applied. After manufacturing, the response of each chip to specially generated test vectors is used to construct the correct per-chip unlocking key. As concerns [167], Choi and Kim focused on the protection of the keys inside TPMs using a PUF. In fact, when the keys are stored in memory and when they are moved through the bus, their value is changed by the PUF, thus rendering eavesdropping out of the PUF IC useless. When the keys are needed for the cryptographic module, they are retrieved from outside the PUF IC and decrypted by the same PUF. However, the values of the keys could be revealed through side-channel attacks, e.g. non-invasive forms of physical attack measuring timings, power consumption, and electromagnetic radiation. Most cryptographic modules are known to be vulnerable to side-channel attacks, and these attacks would be effective against the TPM; thus, countermeasures against side-channel attacks are necessary and will be analyzed in our future works.

5.4 Memory-less Erasable Tokens

The solution proposed in this section is inspired by our previous result and is aimed at enhancing offline authentication security. As in FORCE, this new approach has been applied and published as a Fraud Resilient Device for Off-line micro-payments to show how the mobile payment ecosystem could benefit from it (see Chapter 6). However, as for FORCE, the protocol presented here is aimed at solving open issues on secure machine to machine offline disposable authentication and can be applied to any other use case.

The improvements over FORCE regard both the architecture and the protocol, that have been completely revised, as well as other major substantial novel contributions, as detailed in the following:

1. **Architecture**: differently from FORCE that uses a single hardware element, in FRoDO a token element is used to read disposable tokens in a trusted way, whilst an identity element is used to tie a specific token to a specific user or device. This new design provides a two factor authentication to the claimant. In fact, by linking a disposable token to an identity element, it is not possible for a malicious user to steal and to use tokens that belong to others. A specific disposable token can be read only by a specific identity element (i.e. by a specific device). Furthermore, whilst in FORCE the PUFs are used only to authenticate accesses to the scratch card, FRoDO is also capable to leverage multiple PUFs to authenticate both the identity element and the token element and, last but not least, to make them communicate to each other in a secure way;

2. **Protocol**: while in FORCE the verifier has to communicate directly with the scratch card, in FRoDO the verifier needs to communicate just with the identity element. The identity element identifies a user or a device and has the burden to communicate with the token element thus allowing seamless and faster communication between the involved actors/entities;

3. **Security Properties**: the two-steps layered communication protocol proposed in FRoDO goes from the verifier to the claimant's identity element and then to the token element. This new approach allows the token issuer to design disposable tokens that can be read only by a certain identity element, i.e. by a specific device. This means that even though the token element is lost or it is stolen by an attacker, such element will not work without the associated identity element. On the other hand, the identity element used to improve the security of the claimants can be used to block malicious claimants as well. In fact, if an identity element is considered malicious and is blacklisted, regardless of the token element used in the authentication protocol, all the requests will be rejected. Unlike FORCE, FRoDO does not use a read once memory to store our disposable tokens but they are rather computed on-the-fly by challenging an erasable PUF. In fact, while FORCE is able to avoid the majority of the offline attacks, it leverages

the assumption that advanced attacks are overkill even though they might be practically unleashed to extract meaningful information from the scratch card without changing its behavior. However, thinking about the IoT, such an assumption looks too strong to hold as tiny devices cannot afford advanced shield technologies thus being targeted by introspection attacks that leverage, as an example, on focused ion beam [160].

As for FORCE, FRoDO is based on strong physical unclonable functions [170; 171] without relying on any pre-computed CRDB [53] and can be applied to any scenario composed by a CD, a VD and a storage device that is assumed to be kept physically protected by the verifier. FRoDO does not require any special hardware component apart from the identity element and the token element that can be either plugged into the claimant's device or directly embedded in it. Similarly to secure elements, both the identity and the token elements are assumed tamper-evident, with a secure storage and execution environment for sensitive data thanks to the underlying PUF technology. Thus, as defined in the ISO7816-4 standard, both of them can be accessed via some APIs while maintaining the desired security and privacy level. Such software components (i.e. APIs) are not central to the security of our solution and can be easily and constantly updated thus rendering infrastructure maintenance easier.

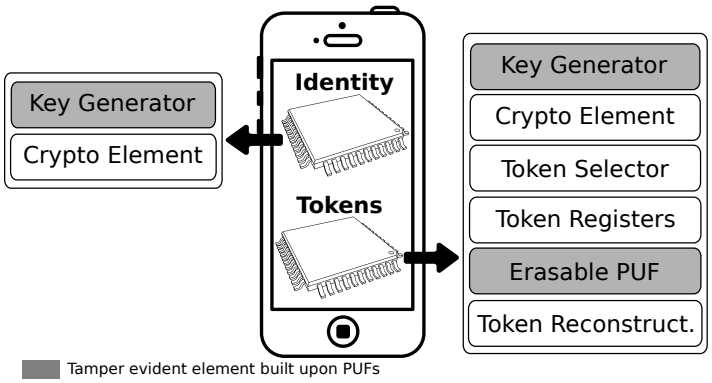

Figure 5.10: FRoDO main architecture

Architecture

As depicted in Figure 5.10, the architecture of FRoDO is composed by two main elements: an identity element and a token element. The token element,

depicted in Figure 5.11, can be any hardware built upon a PUF and it is used to read disposable tokens in a trusted way. The identity element has to be embedded into the claimant device and it is used to tie a token element to a specific device. This new design provides a two factor authentication to the claimant. In fact, the relationship between the token and the identity element prevents an attacker from stealing token elements that belong to other users. The whole FRoDO system architecture can then be decomposed as follows:

- **Identity Element**:
  - **Key Generator**: used to compute on-the-fly the private key of the identity element;
  - **Cryptographic Element**: used for symmetric and asymmetric cryptographic primitives applied to data received in input and sent as output by the identity element;
- **Token Element**:
  - **Key Generator**: used to compute on-the-fly the private key of the token element;
  - **Cryptographic Element**: used for symmetric and asymmetric cryptographic primitives applied to data received in input and sent as output by the token element;
  - **Token Selector**: it is responsible for the selection of the right registers used together with the PUF to obtain the disposable token;
  - **Token Registers**: used to store values required to reconstruct disposable tokens. Such values differ in seeds values (used as input to the PUF) and helper values (used in order to reconstruct stable token);
  - **Erasable PUF[172]**: it is a read-once PUF [172]. After the first challenge, even if the same input is used, the output is random;
  - **Token Reconstructor**: it is responsible to use the PUF, seed values and helper values in order to reconstruct the final content of disposable tokens.

Both identity and token elements are built upon PUFs thus inheriting the following features:

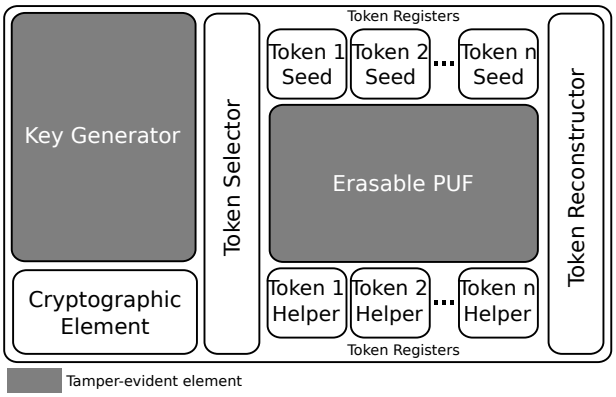

Figure 5.11: Token element architecture

- **Clone Resiliency**: it must be extremely hard to physically clone a strong PUF;
- **Emulation Resiliency**: due to the very large number of possible challenges and the PUF's finite read-out rate, a complete measurement of all CRPs within a limited time frame must be extremely hard to achieve;
- **Unpredictability**: it must be difficult to numerically predict the response of a strong PUF to a randomly selected challenge even if many other challenge-response pairs are known.

In the remainder of this section, each element of the FRoDO architecture is analyzed more in detail.

Key Generator

As depicted in Figure 5.10, the key generator element is used both within the identity and the token element. The main responsibility of such an element is to compute on-the-fly the private key, used during the authentication process by the cryptographic elements to decrypt the requests and to encrypt the replies. In order to compute each private key, as already adopted in FORCE, a publicly known ID is used as input to the PUF. Thus, both the identity and the token element are shipped with such a hard-coded ID signed by the element issuer in order to avoid forgery attacks. This allows the claimant to broadcast the public key of both the identity and the token element to verifiers that are not required to know all of them in advance.

As in FORCE, FRoDO adopts a lightweight key extraction algorithm to compute stable cryptographic keys (see Chapter 5.3 for more details). By using such on-the-fly stable value generation process, the identity/token elements' private keys are not stored anywhere within the claimant device. Hence, they are much better protected from attackers trying to steal them.

Erasable Tokens

At the heart of FRoDO lies a read-once strong physical unclonable function [172]. Such PUF, used to compute on-the-fly each token, has the property that reading one value destroys the original content by changing the behavior of the PUF that will response with random data in further challenges.

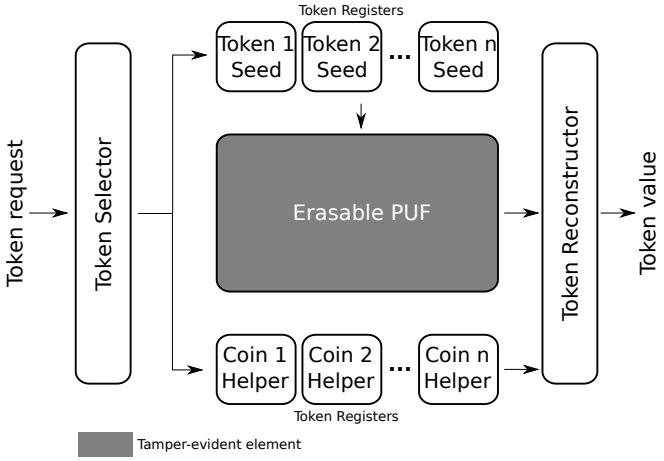

Figure 5.12: Token reconstruction based on an erasable strong PUF.

Disposable tokens, computed by the erasable PUF, do not have to follow any specific format and are not directly written within the claimant's token element but are rather reconstructed on-the-fly. As depicted in Figure 5.12, verifier requests do not contain the erasable-PUF challenge by themselves, but are rather used as input to the token selector. This latter one has information about available tokens and is responsible for the selection of the seed registers (one or more) that will be involved in the authentication. The selected seed registers are then used as inputs to the erasable PUF and the output is XORed together with helper registers in order to reconstruct the final value.

The scheme of a token reconstruction is given in Figure 5.13. As in FORCE, tokens are signed by their issuer and then modified in order to create a chunk

of bytes that are written within seed registers. In addition, helper data are written in helper registers in order to provide stable PUF output [173].

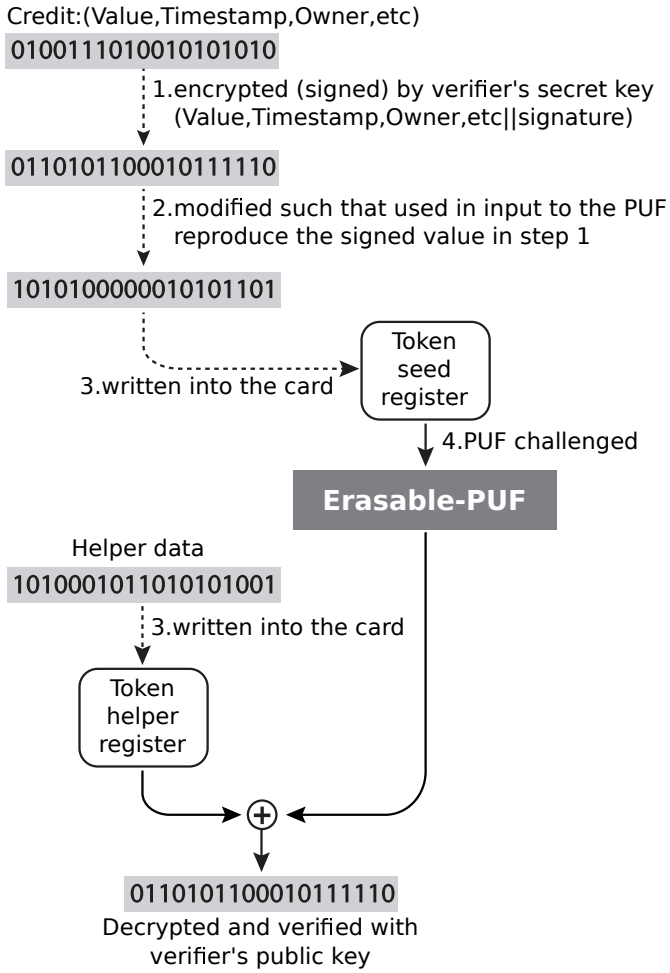

Figure 5.13: Token reconstruction

The Protocol

This section describes the authentication protocol being used in FRoDO (a more detailed description of the protocol applied to offline payments is given in Chapter 6.1). Similar to FORCE, here we introduce and analyze only the two main protocol steps involved in the offline authentication procedure, i.e. the pairing step and the authentication step. Other additional steps such as

Transaction Dispute and *Redemption* are given in Chapter 6.1 as they are not needed in a general purpose authentication.

Pairing Step

The pairing step in FRoDO is exploited to verify the physical presence of the claimant thus avoiding remote attacks. For this purpose, we have not designed any custom pairing algorithm but we have leveraged on the standard Bluetooth SSP pairing approach [174].

At the end of the pairing step, both the claimant and the verifier devices share their public keys that is used for message integrity and authenticity. Furthermore, in order to avoid brute force pairing attacks during the pairing step, FRoDO adopts a *fail-to-ban* approach, already adopted in FORCE, where malicious activities can be detected and malicious claimants blacklisted.

Authentication Step

For the sake of clarity and completeness, the authentication protocol designed for FRoDO is now described from two different standpoints, i.e. the interaction between the verifier and the claimant identity elements and then the interaction between the claimant's identity element and the claimant's token element.

The interaction between the claimant and the verifier is composed by the following operations:

1. The verifier first creates a random salt value. Then, it encrypts the request with the salt and with the public key of the identity element. Furthermore, it signs such encrypted request with its private key to provide authenticity and integrity;

2. When the claimant receives the encrypted requested from the verifier, it first verifies the signature and then computes the private key of its identity element. With the private key it can remove the first encryption layer thus obtaining the salt and the verifier's encrypted request. The result is a token request that is used in the internal protocol between the identity element and the token element in order to retrieve the disposable token;

3. Once the disposable token has been extracted from the erasable-PUF, it is encrypted once again with the salt and the public key of the verifier and last but not least it is signed with the claimant's private key;

4. When the verifier receives the reply from the claimant, it first decrypts it using its private key, the with the salt and one last time with the claimant's public key. If all these operations are accomplished without errors and if the disposable token obtained by the verifier is a valid one (i.e. it has been signed by the token issuer) then the authentication is completed and the claimant is granted to the required services/resources;

So far we have covered the protocol executed between the verifier and the claimant. However, once the verifier request is received by the claimant, the identity and token elements have to execute another protocol between them. The main protocol is composed by the following operations (for a complete and detailed protocol description see the payment application in Chapter 6.1) :

1. Once the identity element has decrypted the token request received by the verifier, such a decrypted message has to be sent to the token element. Hence, it has to be encrypted once again as if it should have been sent to another device. As such, the first operation that has to be done is to sign the token request with the private key of the identity element. Then, the request is encrypted with the public key of the token element;

2. Once the token request has been received by the token element, the first operation is the retrieval of the token element's private key. As for the identity element, the token element uses its embedded ID as a challenge to the PUF that produces in output its private key. Once the token element private key has been computed, the message received by the identity element is decrypted and its integrity and authenticity is verified by checking its signature. If everything is OK, than the erasable-PUF is challenged with one or more seed registers and the output is XORed with the respective helper registers in order to retrieve the token values. Such tokens are then encrypted again with the private key of the identity element and finally signed by the token element;

Solution / Resiliency	Data in Transit	Data at Rest	Data in Memory
VWKP09 [175]	✔	•	•
CYC11 [176]	✔	•	•
KK11 [177]	✔	•	•
CL12 [178]	✔	•	•
CHHZ13 [179]	✔	•	•
CVY13 [180]	✔	•	•
LLQ13 [181]	✔	•	•
KK13 [182]	✔	•	•
YNS13 [183]	✔	•	•
WJ13 [184]	✔	•	•
FORCE [185]	✔	•	•
FRoDO	✔	✔	✔

Table 5.3: Data breach resiliency.

3. When the identity element receives the encrypted disposable token from the token element, if the decryption process is accomplished without errors then the identity element is ready to send everything back to the verifier in order to accomplish the authentication process.

As in FORCE, in FRoDO additional procedures for *disputes* or *token redemption* were not designed as they do not take part in the protocol and go beyond the goal of a disposable authentication approach. However, for a real-world application such as mobile offline payments, such procedures are required to design a practical and feasible solution. As such, all the details regarding the two protocols described here as well as transaction dispute and transaction redemption can be found in Chapter 6.1.

Security Analysis

In this section the robustness of FRoDO is discussed. FRoDO uses both symmetric and asymmetric cryptographic primitives in order to guarantee the following security principles:

- **Authenticity**: it is guaranteed in FRoDO by the on-the-fly computation of private keys. Furthermore, public keys used by both the verifier and the claimant's elements are signed by their issuers. As such, their authenticity can always be verified;

- **Non-Repudiation**: the storage device that is kept physically safe by the verifier prevents the adversary from being able to delete past log entries, thus protecting against malicious repudiation requests. Furthermore, the content of the storage device can be backed up and exported to a secondary equipment, such as pen drives, in order to make it even harder for an adversary to tamper with it;
- **Integrity**: it is ensured with the signature of each disposable token by the identity/token element issuer. Token seeds and token helpers are written into the token element registers by the token issuer such that the final token value given as output corresponds to an encrypted version of the real token. As such, by using the public key of the token element issuer, it is always possible to verify the integrity of each token. Furthermore, the integrity of each message exchanged in the protocol is provided as well. In fact, both the identity and the token elements use their private/public keys. The private key is not stored anywhere within the identity/token element but it is computed each time as needed;
- **Confidentiality**: both the communications between the claimant and the verifier and those between the identity element and the token element leverage asymmetric cryptographic primitives to achieve message confidentiality;
- **Availability**: the availability of the proposed solution is guaranteed mainly by the fully offline scenario that completely removes any type of external communication requirement and makes it possible to use offline disposable tokens also in extreme situations with no network coverage. Furthermore, the lack of any registration or withdrawal phase, makes FRoDO able to be used by different devices.

As in FORCE, FRoDO shares the assumption that each element built on top of a PUF is tamper-evident. This assumption is based on the size of nowadays integrated circuits and on the impossibility for a casual attacker to open the device without causing an alteration in the PUF behavior.

Blacklists

FRoDO uses two different elements: an identity element and a token element, in order to improve the security of the whole authentication system. In fact, the verifier's device does not directly communicate with the token

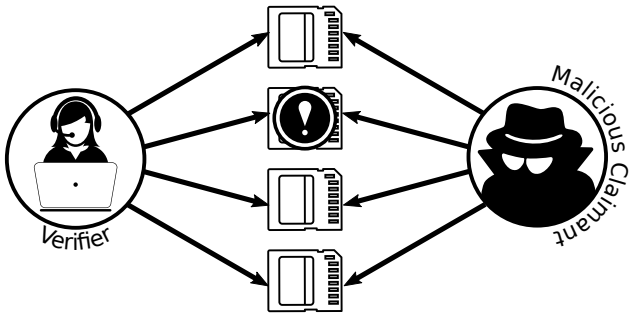

Figure 5.14: Malicious claimant playing with multiple cards

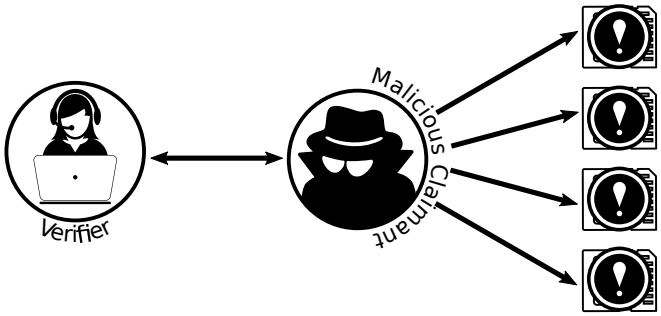

Figure 5.15: Blacklisted malicious claimant

element but has to go through the identity element. On the one hand this allows the token element issuer to design all the disposable tokens to belong to a specific device or user and thus to be read only by a certain identity element. This means that even though the token element is lost or it is stolen by an attacker, such element will not work without the associated identity element. As such, the identity element can be considered as a second factor aimed at improving the security of claimant tokens. On the other hand, the identity element can be used to fight against attackers. In fact, whilst usually an attacker could play with different scratch cards without ever being recognized (as shown in Figure 5.14), if an identity element is considered malicious and is blacklisted, no matter what is the device used by the user, any token coming from any token element will never be accepted and processed by the verifier (as depicted in Figure 5.15). As usually the identity element is per-device specific, this second factor approach enables verifiers in blocking malicious devices thus mitigating attacks based on the brute force.

Attack Mitigation

In this section, an analysis on the resiliency of each individual attack is provided:

- **Double Usage**: the read-once property of the erasable PUF [172] used in this solution prevents an attacker from computing the same token twice;
- **Token Forgery**: each token is encrypted by the token element issuer and thus it is not possible for an attacker to forge new tokens;
- **Emulation**: PUFs, by design, can be neither dumped nor forged, neither in hardware nor in software. Responses computed by emulated/fake PUFs will be different from the original ones;
- **Information Stealing**: the private key of each element is computed on-the-fly as needed. No sensitive information is kept neither in the identity nor in the token element. Token seeds and token helpers do not provide by themselves any information about tokens and physical access to the hardware will cause the PUFs to change their behavior;
- **Replay**: each transaction is different due to the random salt generated each time by the verifier;
- **Man In the Middle**: digital tokens are encrypted by the token element issuer and contain, among all other things, the ID of the token element. Furthermore, as in FRoDO disposable tokens are computed at run-time rather than being written within the device, an attacker cannot *dump* tokens from another claimant. Last but not least, an attacker cannot pretend to be another claimant with a different ID because it will not be able to sign any message;
- **Reverse Engineering**: by design, any attempt to tweak and steal any useful information from either the identity or the token element will alter the behavior of the PUFs thus rendering both the identity and token elements no longer usable;
- **Denial of Services**: FRoDO uses the standard Bluetooth SSP pairing process with a fail to ban approach to stop denial of service attacks;
- **HW Modification**: by design, it is not possible for an attacker to either add or modify or remove any element belonging to identity/token element without changing its behavior;

Taking into account all the different kind of attackers defined in Chapter 5.2, FRoDO showed to be resilient to (a) malicious claimants, (b) malicious verifiers and (c) ubiquitous attackers. The way in which those attackers are mitigated is the same previously achieved in FORCE (for more details see Section 5.3).

Data Breach Resiliency

As already introduced in Chapter 5.2, offline authentication systems can be attacked to steal private and sensitive claimant's information. However, devices that belong to authentication systems are usually kept physically and digitally secure. Hence, attacks against authentication systems in mature environments are typically multi-staged and, in offline scenarios where there is no connection to the outside world, stolen data has to be kept hidden within the authentication system waiting for the attacker to be back.

The scenario is completely different in the IoT where usually there are not specific and dedicated authentication devices and where each claimant usually can also serve as a verifier depending on the use case. In this new scenario, all the attacks that have been introduced in Chapter 5.2 are even more dreadful, since claimant devices, are continuously threatened by cyber-attacks at any time. This means that an attacker does not need anymore to infiltrate and traverse dedicated authentication systems but just needs to compromise sybil things that will later represent verifiers or even use forensic tools [186] in order to steal claimants' information and keep them hidden within the device itself, ready for an exfiltration.

Solutions proposed so far and focused on offline authentication protocols try to guarantee security requirements such as double usage resiliency, forgery resiliency, and anonymity. However, regardless of the trustworthiness assumptions they make, all of them completely lack a data breach analysis. In fact, as shown in Table 5.3, they suffer from both *data at rest* and *data in memory* vulnerabilities. This is mainly due to the fact that, to the best of our knowledge, all current offline solutions adopt a withdrawal phase. In such a phase, tokens are precomputed and precached within the device. Later, during the authentication protocol, such tokens are used as identity or possession evidences to authenticate the claimant in a way that the verifier can validate even without connecting to an external TTP. However, token precaching allows attackers to extract them from the claimant's device. Furthermore, even for those offline solutions that do not make use of withdrawal procedures, such as in our FORCE approach, persistent memo-

ries are usually exploited either to store or to reconstruct tokens at run-time (see Figure 5.16). Hence, the exposure of sensitive information that can be extracted from the hardware makes those solutions vulnerable to data at rest and data in memory vulnerabilities.

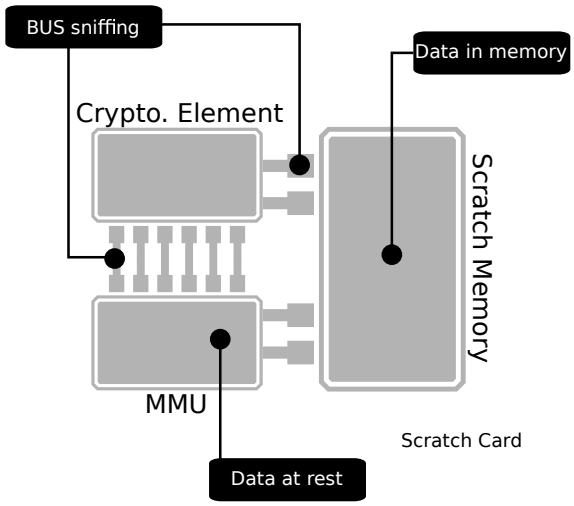

Figure 5.16: FORCE data breach vulnerabilities

FRoDO architecture has been designed to be immune from any data breach threat. In fact, all the information required in the authentication protocol, such as private keys or disposable tokens, are computed on-the-fly and are never stored anywhere within the device thus mitigating data at rest breaches. Furthermore, any value required during the authentication process is directly computed in hardware by challenging PUFs. As such, all the data are never going to be loaded into the device memory thus mitigating also data in memory breaches.

The analysis of FRoDO security against physical attacks and the protocol adopted for identity and token keys rollover will not be discussed here as it leverages the same techniques used in FORCE and both have been already analyzed in Chapter 5.3 and Chapter 5.3.

Chapter 6

Real World Application

6.1 Offline Mobile Payments

The research of electronic cash has long history and has been a hot research topic in cryptography for years [187]. Usually the banks needs to be involved in the payment process in order to prevent specific attacks such as double spending, however this not only limits the privacy of both the payee and the payer but also prevents such approaches from being used in offline/private environments.

The first solution that tried to solve this challenge and to provide a Bank-free payment protocol was proposed by Chaum, Fiat and Naor (for short, CFN) back in 1988 [188]. Following, many other solutions were proposed, all based on the CFN paradigm where a customer withdraws electronic-cash (for short, e-cash) and later sends this e-cash to a shop without the need of involving the bank. If the e-cash is spent only once nothing happens and both the payer and the transaction remain anonymous but if the same e-cash is spent twice then the bank can extract the identity of the payer and block any further payment. In CFN, to enable anonymous transactions and to detect double spending, the identity of the payee is split into multiple pieces that are then written within each e-cash. To reconstruct the identity of the payee, two pieces from the same e-cash are needed, thus by avoiding to double spend the same e-cash, the payee can be sure that his/her transactions will remain anonymous. However, e-cash are shipped by the bank and payee identity information are written within e-cash by the bank as well. This means that the bank has complete access to identity information of its customers. This scenario is usually accepted as the bank or any other payment gateway is

assumed to be trusted and insider attacks by untrusted authorities are not taking into account. However, the bank is not an abstract entity. It is made by people and those people might suddenly turn malicious. It is then of paramount importance to design e-cash protocols for the real world to be resilient against both outsider and insider threats.

The first work that analyzed and discussed the security of an e-cash system against insider threats was published by Miyazaki and Sakurai back in 1999 [188]. Later on, many other solutions were proposed and all of them based on the CFN paradigm [189; 190; 191; 192; 193]. In such a paradigm, three entities take part in order to enable offline payments:

- **A bank (B)**: this entity has a pair of public and private keys respectively P_B and S_B. A signature generated by S_B can be considered to be e-cash representing some money w. In order to provide anonymous transactions for its customers, B generates the above signatures (i.e. the e-cash) with a blind signature technique [194];
- **A customer (C)**: this is the entity willing to spend e-cash that has been withdrawn from B, to a shop. This entity as well owns a public and private key pair P_C and S_C;
- **A shop (S)**: this entity accepts e-cash from customers. E-cash is then sent back to B in order to check for malicious transactions such as double spending. This entity as well owns a public and private key pair P_S and S_S.

By using the above three entities we can then define three different protocols (see Figure 6.1):

- **Withdrawal Protocol**: this is the first step required for a digital transaction. In such a protocol, C generates a message m_C signed with its private key S_C. By relying on a blind signature scheme, C proves to B that the message m_C has been signed with S_C without revealing S_C to B. At this point B as well generates a signature on m_C by using S_B with a blind signature and then withdraws money corresponding to w from the bank account owned by C which now computes $\sigma_B(m_C)$ as the B signature on m_C;
- **Payment Protocol**: C sends $(m_C, \sigma_B(m_C))$ to a shop S. S verifies the signature $\sigma_B(m_C)$ and if it is correct, S sends a random challenge

c_S to C. C computes the response r_C and sends it back to S that finally verifies r_C and if it is correct exchange goods for the e-cash provided by C;

- **Deposit Protocol**: S sends $(m_C, \sigma_B(m_C), c_S, r_C)$ to B that verifies its signature $(m_C, \sigma_B(m_C))$ and the challenge-response pair (c_S, r_C). B then stores $(m_C, \sigma_B(m_C), c_S, r_C)$ in a database for future detection of double spending attacks and credits w to the shop S. If a double spending is detected, B can extract the identity of C by the two transactions (c_S, r_C) and (c'_S, r'_C)

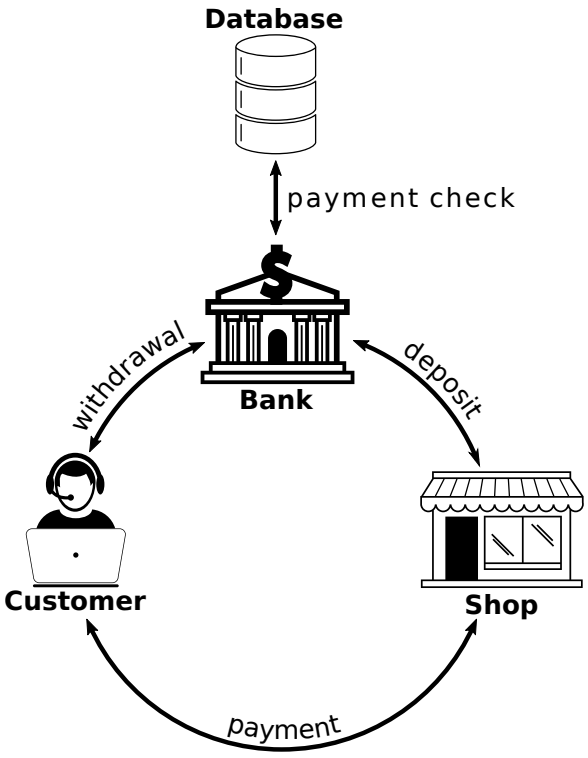

Figure 6.1: Chaum, Fiat and Naor (CFN) general payment scheme

Usually, all the payment protocols adopt the above three steps. However, depending on the environment or special use cases, those three steps can be changed depending on the constraints of the environment. Hence, depending on their connection capabilities, all the solutions proposed so far for mobile payments can be classified according to the following taxonomy:

- **Online**: solutions such as [195; 196; 197] that require the customer's mobile device to be connected to a network (e.g. 3G) in order to communicate with a bank, a payment-gateway or a TTP;

- **Semi Offline**: solutions such as [198; 199] that require an active connection only from the vendor side;

- **Weak Offline**: solutions such as [200; 187] that require a connection either to a shared data-set or to a peer-to-peer network. Such approaches, by allowing access to past transactions, enable vendors to check for customer's account validity, thus preventing fraudulent behaviors. Other solutions belonging to the weak-offline category work with digital cash designed to be accepted either by specific vendors (known as digital vouchers) or within a specific short time window like in [201; 202];

- **Fully Offline**: solutions that do not require any external connection but either assume involved devices to be trusted [184; 203] or are limited to transactions that can leverage on a bank account.

Furthermore, based on customer's information known by the bank, offline solutions can be further grouped into four distinct types (see Figure 6.2):

- **Type I**: any customer's information, either private or public, is kept by the bank in its internal databases and written within e-cash during the withdrawal phase. Here an insider adversary can easily impersonate the user if it has access to the database;

- **Type II**: customer's private information (such as the private key) is kept by the customer whilst public information is stored within the bank database. In this approach a malicious bank employee could impersonate a customer only if it is able to steal the private key paired with the public one;

- **Type III**: the customer register his public key as his identity information and then makes purchases with the private key and the public certificate linked to the public key. For an insider adversary to succeed with an impersonation attack, the certificate has to be obtained and the secret key has to be extracted from the public key;

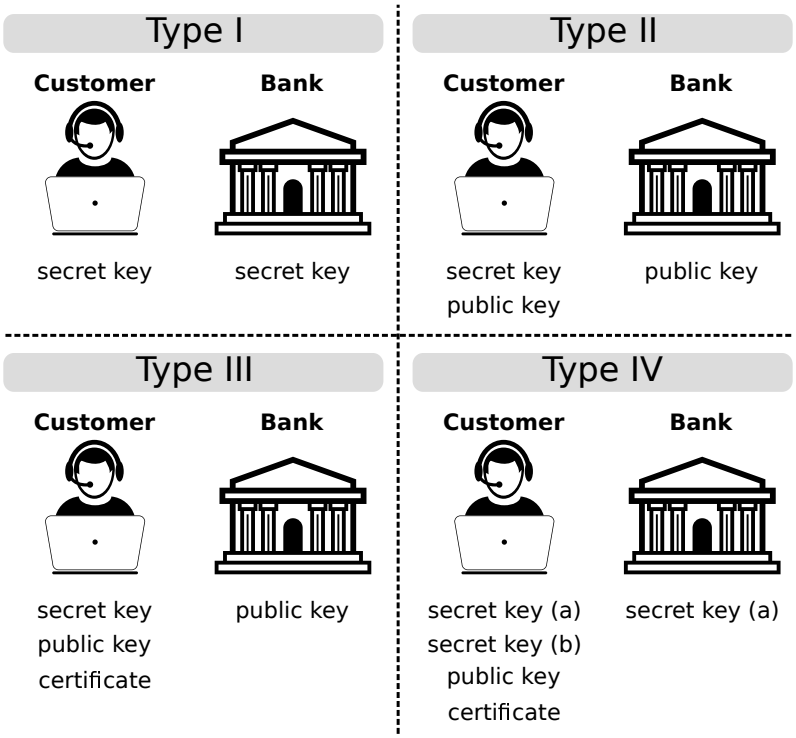

Figure 6.2: Customer data distribution

- **Type IV**: in this type of payment, each secret key is split into two different components. One is known to both the bank and the customer whilst the other one is only known by the customer. Furthermore, the public key and the certificate linked to the secret key are also known only to the customer. In systems adopting this type of payments, customer's information stored within the bank database can only produce little knowledge of the corresponding customers' secret keys. Therefore, this incomplete information makes the impersonation attack harder and the security/privacy of customers better.

As an general consideration, payments solutions proposed so far which belong to type IV proved to be the most secure ones thanks to the limited information provided to the bank. Some of this solutions have also adopted hardware-based countermeasures to make the payment process more secure. As an example, in [204] a tamper-proof device was exploited to store customer identification information known only to the bank. Such a device was

provided to customers with the first half of their secret key embedded in the devices whilst the second half given to them. In this case, instead of keeping part of the secret key within the bank, thus allowing insider attacks, the tamper-proof device was designed as a trusted third party acting as an extension of the bank. However, by taking into account a malicious customer and by considering unfeasible the practical verification of the authenticity of the tamper-proof device, such a solution falls into Type II. Other solutions have been designed during the last years aimed at both protecting customers from insider threats while at the same time providing countermeasures against fraudsters. Blazy et al. [193] investigated offline transferability, i.e. the capability of directly exchanging money without involving any TTP. In their solution, a trusted authority called judge was introduced, aimed at providing double-spending detection mechanisms without giving any personal and secret customer's information to the bank. In this approach, the customer by itself generates the public and private key and then obtains a certificate from the trusted authority.

It is therefore possible to conclude that, fighting against insider threats require the less customer's information to be stored within the bank or any other TTP that might turn malicious now and in the future. Furthermore, in fully offline scenarios, there is no bank connection at all, thus no banks no TTPs can be involved. This represents a perfect chance for insider threats as no information is ever exchanged with others and it also makes hard for the vendor to detect double-spending attacks as there are no logs on the customer's past activity. In fact, the main issue of a fully offline scenario is that keeping track of past transactions can be hard, as it is difficult for a vendor to check if some digital credits have already been spent. This is the main reason why the solutions proposed so far in the literature (see Table 6.1) require some kind of TTP to store past transactions within a list and check such a list, or demand for such a validation process, each time a new transaction is made [197]. Alternatively, offline solutions that do not rely on TTPs either assume a tamper proof/resistant smart card (such as [184; 203]) where to store sensitive information, or just check for customer's identity [179] whereas security checks, are verified and validated by the bank at a later time (such solutions are classified as postponed as there is no way to check the validity of the transaction at run-time).

Following in this section, we will introduce a practical use case that can benefit from a secure offline authentication. This application is about micropayments and we will later see how the hardware intrinsic security approach enables two devices to authenticate to each other and to accomplish pay-

System	TTP	Note
CFN88 [188]	Bank	The first offline e-cash system
FY93 [205]	Bank	Provable Security Argument
Sch95 [206]	Bank	Scheme withstanding parallel attacks
BGK95β [207]	Bank	Franklin-Yung based [208]
JY96 [209]	Bank	Resistant against bank robberies attacks
Pai92 [210]	Bank	Based on Guillou-Quisquarter scheme
Fer93 [211]	Bank	Discussion on framing by a malicious bank
ASM11 [212]	Bank	Anonymous customer suspension
Bra93 [213]	Bank	Based on the technique of restrictive blinding
DdC94 [214]	DB	Transferable e-cash w/o any increase in size
BGK95α [207]	Bank	Brands-based [213]
CMS96 [215]	Bank	Passive anonymity-revoking trustee
FTY96 [216]	Bank	Anonymous revoking via ElGamal decryption
NMV97 [217]	Bank	Based on Nyberg-Rueppel signature
dST98 [218]	Bank	Based on modified restrictive blinding scheme
BCFG11 [193]	Bank	Transferable e-cash w/ Groth-Sahai proofs
CGT08 [219]	Bank	Transferable e-cash w/ unconditional anonymity
CG08 [191]	Bank	Transferable e-cash w/ perfect anonymity
CG10 [192]	Bank	Multiple denomination in e-cash
OO91 [220]	Bank	The first divisible e-cash
EO94 [221]	Bank	Single-term divisible e-cash
Oka95 [222]	Bank	Improving efficiency of scheme [221]
PP97 [223]	TTP	Extortion-tracing under offline payment
HKOK07 [190]	PKI	Keys of customers and bank are based on PKI
FO96 [224]	TTP	Anonymous channels or distributed structure
Yac94 [225]	Bank	GMR-ZKP at initial certificate and withdrawal
Bra95 [204]	Bank	Tamper-proof device for customer ID
MS98 [226]	RC	Partially blind signature based on DLP
XAG06 [227]	P2P	One-way hash functions for e-coin encryption
PO07 [228]	DB	Offline e-cash based on bilinear pairings
XuZu08 [229]	P2P	Fair offline e-cash scheme based on ECC
AbASA08 [230]	DB	Hidden Markov Model (HMM) frauds detection
ChRo08 [231]	DB	ID-based fair and transferable offline e-cash
WaWa09 [232]	DB	Offline e-cash scheme based on ECDLP
VaHV10 [197]	CA	Anonymous subscription schemes
SEH11 [203]	DB	Mobile agent based electronic cash system
CHZ13 [179]	DB	Blind signatures-based rescue scheme
WJ13 [184]	DB	Anonymous exchange using bilinear pairings
BrS14 [233]	DB	VANET toll collection-based payment

Table 6.1: Offline payment schemes with double-spending prevention

ment transactions without relying on any external TTP or money accounts. Micro-payments is the best application to show the importance of secure offline authentication approaches. In fact, mobile devices can suffer from temporary disconnections from the network as for payments during flights or for network maintenance. Furthermore, privacy can also benefits from offline transactions as neither TTPs nor bank account are involved.

Mobile payments nowadays allow banks to offer additional and easy-to-use services to their customers. However, mobile payment technology is still at its early stages of evolution, albeit already supported by recent hardware. As an example, NFC [234] is present on most recent smartphones. As such, advanced short-range communication and a computing power similar to that of a recent netbook are nowadays quite common. This scenario is producing a shift in purchase methods from classic credit cards to new approaches such as mobile-based payments. The possibility to use smartphones or other mobile devices as digital wallets allows to shop pervasively without having to carry any credit or debit card [235; 236] and this is going to improve over time with the IoT allowing users to shop with their smart watch rather than with their Google glass etc. However designing a mobile payment system is a challenging task requiring consideration of several factors such as customer preferences, technological environments, social cultures, legal and regulatory requirements and standardization.

The acceptance of mobile payment systems mostly depends on the platform on which payments rely, the most important of which are now introduced here:

- **Pure SMS Platform**: in this platform (see Figure 6.3) SMS are used for all the communications between the user and the payment network. Standard formats for SMS messages are used for the communications. Such messages can contain, among all other things, timestamps, random numbers, user account information and so on as needed by the protocol being used. User authentication is based on the mobile number being used for the communication. Possible real world application for this platform are bill payments or financial operations like account history and funds transfer [237; 238];

- **USSD Platform**: USSD is a protocol used by GSM capable devices in order to communicate with the service provider computers. Users can request provider services by entering short codes on the mobile.

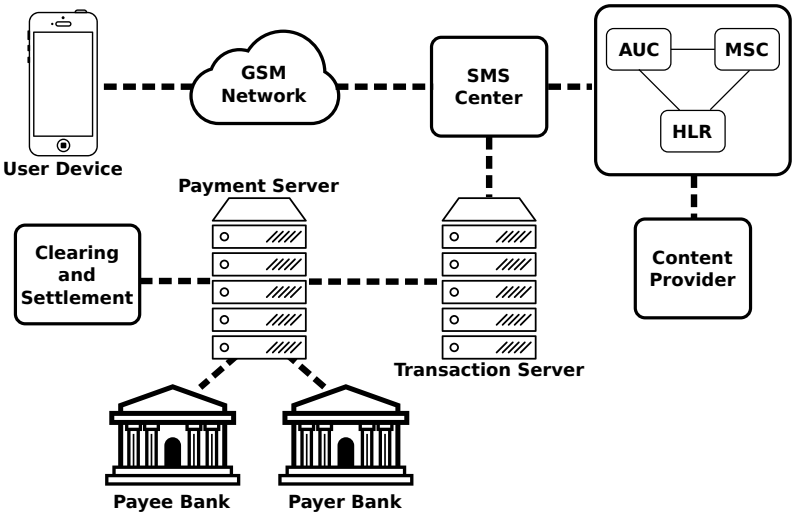

Figure 6.3: SMS platform

Such codes are standardized and their content is defined for each service. As shown in Figure 6.4, users send USSD request codes to the USSD gateway trough the GSM network. USSD gateways create a session and route session's information to suitable applications. Applications send back this information to the USSD gateways that will take care to forward such response to the users whilst the payment server will take care of bank interaction. Some of mobile services which can be provided by this platform include electronic content purchase and reservation [239; 240];

- **WAP/GPRS Platform**: this platform can be used to make payments through mobile Internet connections. In this case the authentication of the user is done by digital certificate, mobile phone numbers and secret PIN codes. As shown in Figure 6.5, communications are routed by WAP gateways that convert WML requests into HTML thus allowing WAP enabled mobile devices to browse the Internet. Mobile services which can be provided by this platform include financial operations and web store purchase [237; 238];

- **Phone-Application Based Platform**: in this platform, payment operations are done through payment software installed directly within the mobile device. A complex communication scheme composed by SMS, USSD and WAP is needed to transfer payment information.

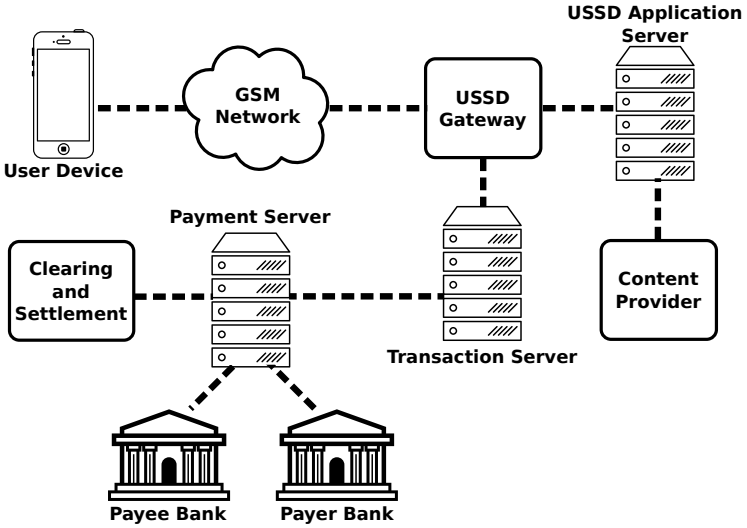

Figure 6.4: USSD platform

Details on SMS, USSD and WAP communication platform scheme can be seen in Figure 6.3, Figure 6.4 and Figure 6.5. Depending on the channel being used in a specific payment transaction, costs, security and accessible services could be different. Main drawbacks of this platform are manual installation/update of mobile applications and duplicate installation in case of phone exchange. Nonetheless, the advantages are end to end security, content encryption and improvement of network bandwidth usage [241; 242];

- **SIM-Application Based Platform**: this platform works with applications installed directly into the SIM card of the user. Such payment software and other services can be downloaded over the air (for short, OTA). When such software are successfully installed, the user can use them by sending requests for supported services to the operator. These requests are processed in OTA servers and recorded on transaction servers. Requests can be encrypted for a higher security and privacy. Then, OTA servers decrypt such requests by HSM which include encryption keys. It is important to note that binary SMS are different from normal SMS. With binary SMS it is possible to use rich content messages encrypted in text messages. Furthermore, such messages are no stored into the phone but kept in the SIM. SIM-application tool-kits enable SIM to provide value-added services [243].

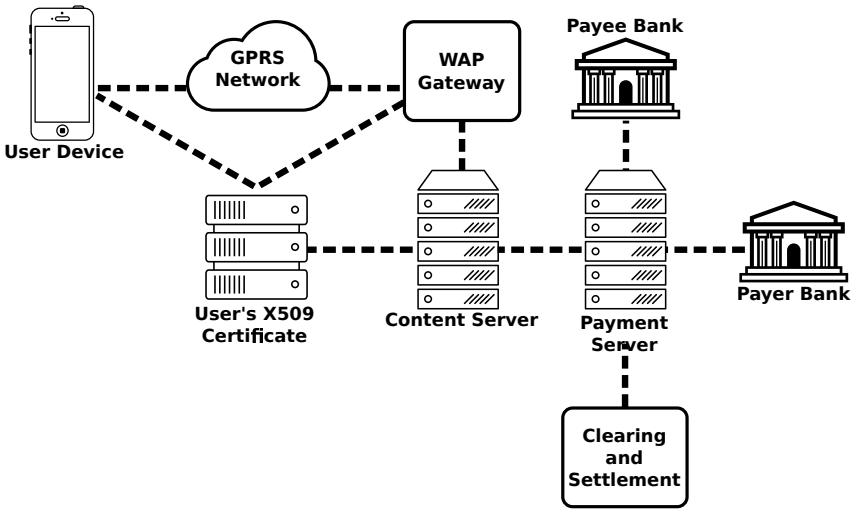

Figure 6.5: WAP/GPRS platform

Some mobile services which can benefit from platform include financial operations, electronic and physical good purchases and reservations. The architecture of this platform has been shown in Figure 6.6;

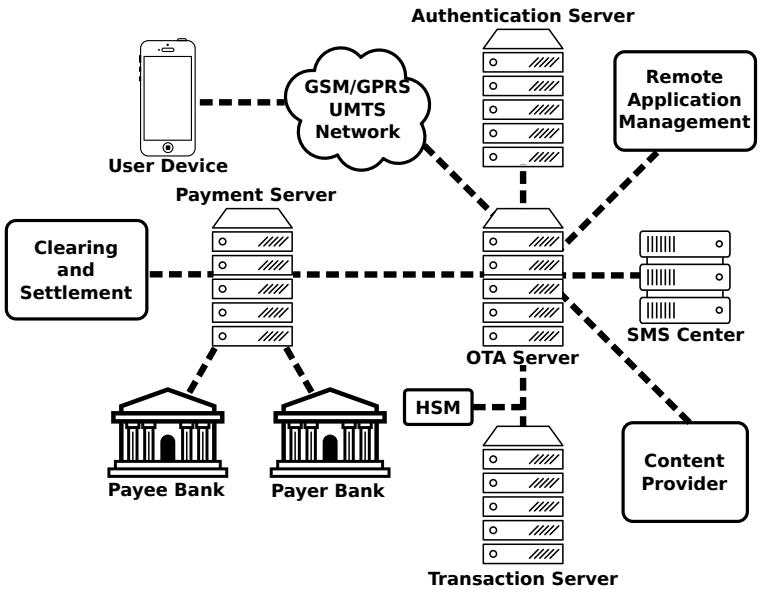

Figure 6.6: SIM-application Based platform

- **Dual Chip Phone Platform**: some phones are built with two chip slots. Usually, one of them is for the SIM-card and the other one is for a payment card. Some pilot projects have been proposed with this features such as the Europay, MasterCard, and Visa (for short, EMV) by Visa [244]. With such solution the user should place the second card on the phone and start the transaction by entering the related PIN number. Some mobile services that can benefit from this platform are financial operations, electronic content purchases and reservations. The architecture of this platform has been shown in Figure 6.7;

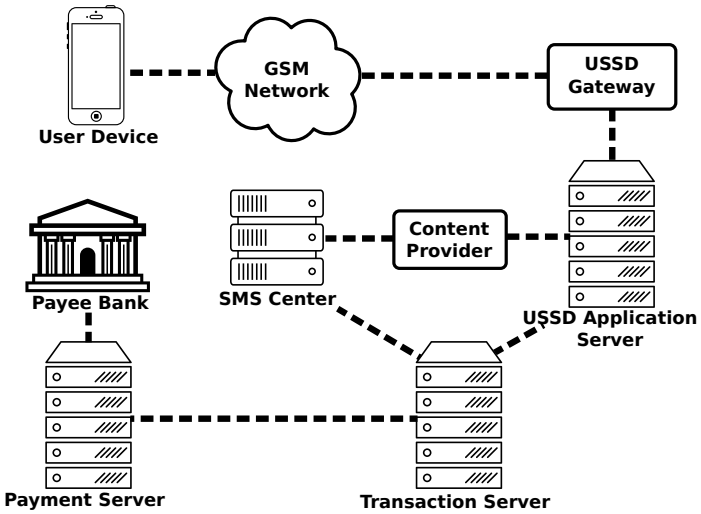

Figure 6.7: Dual-chip phone platform

- **Short Range Communication Network Platform**: in this platform purchase requests can be sent through short range communication networks like Bluetooth, NFC technology and infrared data association (for short, IrDA) to the vendor's machine. As shown in Figure 6.8 either the short range or the network connection can be removed. If no short range connection is available, the user can communicate to the bank through the vendor's machine. Instead, if the user device does not have any short range technology capability, payment requests can be sent through the network directly to either the bank or any other payment gateway.

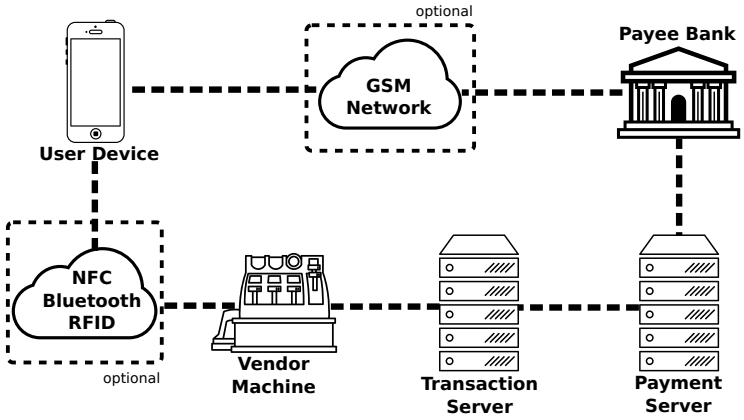

Figure 6.8: Short range communications platform

Threat Model

In this section we are now going to introduce and to describe the most common attack techniques that can thwart the security of any of the above listed payment platforms.

Malicious Insiders

Insider threats are all those malicious exploitation of the payment protocol from within the bank. Banks are responsible to manage money accounts, to verify inflows and outflows and to enable offline payments before the double spending attacks is detected. However, the bank or any of its employees, can turn malicious and cheat over both customers and merchants. In this section we are going to introduce classic and well known insider threats.

Impersonation. This is the attack in which a malicious bank employee steals customer credentials in order to accomplish payments on his/her behalf. This attack is feasible as customers and merchants key-pairs are stored, and usually generated, by the bank. Therefore, the malicious employee could withdrawal e-cash from a customer's account by using its private key or even worse, the malicious bank employee could even try to double spend the same e-cash without compromising himself/herself. It is then clear that by storing customer's private keys within the bank database it is nearly impossible to prevent the insider attack.

Framing Attack. In this attacks malicious banks or bank employees misuse customer's key pairs to forge fake double spending evidences. This kind of attack is possible in all those payment protocols such as [207; 213; 204; 218; 221] where ElGamal and Okamoto-Schnorr signatures are used in the payment process to identify double-spending attacks. Framing attacks require multiple double-spending transactions to reconstruct customer identities but this can be easily achieved by software flaws, malware injections, etc. In the Yacobi's scheme [225], the ElGamal signature is used similarly, but the attack is slightly different as the bank does not know the customer's secret key. In such a scenario, banks can still exploit framing attacks on double-spenders by creating additional fake evidences.

Bank-Shop Collusion. In payment systems such as [225; 226; 224] the customer uses the same signing key and public key certificate in every payment transaction making them exposed to insider attacks as shown above. In such a scenario, the shop is able to recognize each customer and to distinguish between them thus being able to pair each customer with the public certificate used while shopping. Therefore, if the bank and the shop collude, the bank can know the purchase history of each costumer due to the re-use of the public certificate in each different payment. Peterson and Poupard [223] solved this issues and made harder for banks and shops to trace and profile customers' purchases by leveraging on multiple payment keys.

System Breaches

While in the previous section we have introduced different kinds of attackers and attacks, in this section we are going to describe how those attacks are usually accomplished.

Attacks against points of sale (for short, PoS) systems are typically multi-staged [245] and can leverage on vulnerabilities exposed by different devices. First, attackers must gain access to the victim's network (this step is called *infiltration*), then attackers have to traverse the network (this step is called *propagation*) and ultimately they have to break into PoS systems. Once into the payment systems, they have to install malicious software in order to steal data from the compromised devices (this step is called *aggregation*). As the PoS system is unlikely to have external network access, the stolen data is then typically sent to an internal back-office server (see Figure 6.10) waiting for the attacker to be back (this step is called *exfiltration*). Figure 6.9 depicts all the devices usually involved in payment systems of different

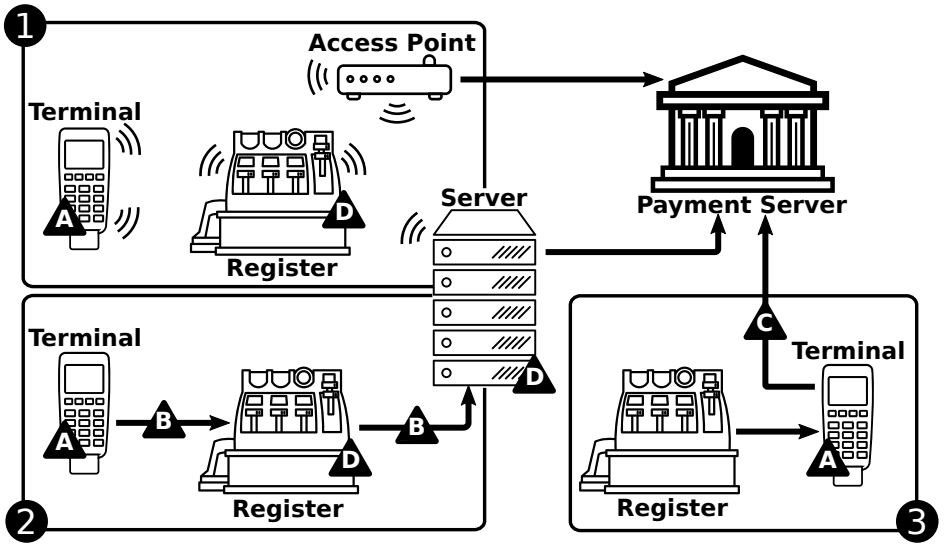

Figure 6.9: PoS system threats

nature. In that figure, (1) shows an online payment system with direct access to the bank, (2) shows a payment system where each transaction is first collected within a server and then sent to the bank whilst (3) shows an offline payment system where each transaction is stored locally within the PoS device and then sent to the bank when the connection is available. In all the three scenarios, all the involved devices can be hacked and thus all of them cannot be considered trusted in a payment protocol. In particular, (A) represents attacks against the PoS device, (B) attacks against sensitive data while in transit within the vendor's network, (C) attacks against sensitive data exiting the vendor's network and directed toward the bank and (D) attacks against data collected from the vendor.

PoS network hacking can be achieved by exploiting shared connections, open networks, or by cracking the password of the merchant's network. However, networks can be monitored and protected against malicious activities [246]. Network infiltration is just one of the many sophisticated attack methods. In addition, a successful server breach would give attackers not only the access to a single PoS system or to a network of PoS systems in a single location but, depending on the architecture, possibly to all PoS systems controlled by the retailer, even in multiple locations.

Regardless of the adopted electronic payment system model (for short, EPS), the payment process is composed of two main steps, *authorization* and *set-*

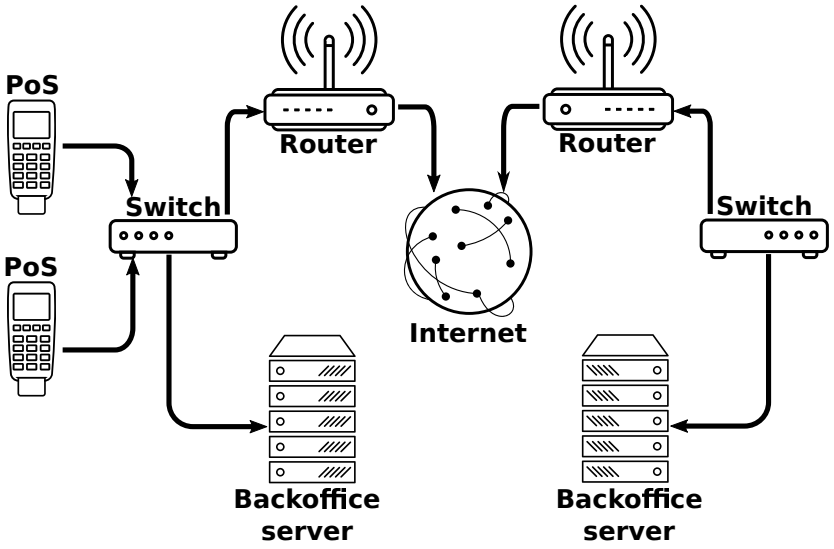

Figure 6.10: Point of Sale architecture

tlement. On one hand, the authorization is the state of the payment process where the purchase is verified and finalized. On the other hand, the settlement comprises all actions happening after the authorization stage. Even though data processed at this stage is not as valuable as data processed during the authorization stage, it still contains sensitive information such as the amount of money spent within the transaction. Such information is relevant to customer's privacy and thus it has to be protected.

Device Breaches

PoS devices are the key elements in electronic payment systems and are normally *guarded* by employees during operating hours. However, it is still possible for an attacker to inject malware within the PoS or even to replace it with a fake one. Furthermore, many all-in-one PoS systems are based on general purpose operating systems and, as such, they are susceptible to a wide variety of attacks which can lead to large scale data breaches.

All the attacks described so far require the PoS to be connected to a network in order for the attacker to break into the payment system and infect either the PoS itself or a specific component within the EPS. However, EPS can also be fully offline. In this scenario, no data is going to leave the PoS and there is no way to remotely infect the PoS. As such, breaches based

Symbol	Meaning
Enc()/Dec()	Symmetric encryption/decryption
Enc()/Dec()	Asymmetric encryption/decryption
Salt	Salt value
CreditIdx	Credit memory addresses
CreditVal	Credit memory content
Req	Credit request built by VD
Res	Response built by CD
CPK	Card public key
CSK	Card secret (private) key
BPK	Bank/Card Issuer public key
BSK	Bank/Card Issuers secret (private) key
VPK	Vendor public key
VSK	Vendor secret (private) key
EReq	Encrypted request
ERes	Encrypted response
FRes	Final response
RReq	Redemption request
Log	Log entry
ELog	Encrypted log entry

Table 6.2: Symbols used in all the phases of the transaction protocol

on network-level hacking are not feasible in this scenario. However, data processed by the PoS can still be eavesdropped by having physical access to it or by exploiting device vulnerabilities. It is then important to design self-enforced devices resilient to both outsiders and insiders threats.

FORCE: Payment Protocol

In this section we are going to analyze the FORCE protocol when applied to mobile payments. In Section 5.3, we have already provided an overview of the authentication protocol leveraged by FORCE. In this section we are now going to analyze the same protocol in the presence of a payer and a payee. In this scenario, disposable tokens described in Section 5.3 are substituted by digital credits where a digital credit is the virtual representation of a digital coin. As such, it is important that digital credit are not spent more than once.

Payment Step

FORCE payment step is composed by the following operations (symbols in Table 6.2):

1. The customer sends a purchase request to the VD asking for some goods;
2. The vendor computes the total amount and sends it back to the customer;
3. The customer checks for the amount and either confirms or denies the transaction. If the transaction is confirmed, the CD creates a reply for the VD with the indexes of all the credits that are still available in the card. If the i^{th} index number is present in the reply, it means that the i^{th} credit register can be read in order to retrieve the i^{th} digital credit within the card;
4. The vendor first creates a random salt value. Then, for each credit that will be involved in the transaction, a request is created by encrypting the credit index with the random salt obtaining *Req*

$$Enc_{Salt}(CreditIdx) = Req \tag{6.1}$$

5. Such encrypted request along with the salt just created are encrypted once again with the public key of the scratch card, thus rendering the customer the only one able to read it

$$\underline{Enc}_{CPK}(Req, Salt) = EReq \tag{6.2}$$

6. When the customer receives such a request, the private key is computed by the authenticator and it is used to decrypt the message received thus obtaining the salt value and the request

$$\underline{Dec}_{CSK}(EReq) = (Req, Salt) \tag{6.3}$$

7. The salt is then used to decrypt the request *Req*

$$Dec_{Salt}(Req) = CreditIdx \tag{6.4}$$

8. CreditIdx is used by the MMU to read the scratch card digital credit value;
9. The credit value is sent back to the authenticator;
10. The salt is used once again to create an encrypted response for the vendor

$$Enc_{Salt}(CreditVal) = Res \tag{6.5}$$

11. The response is encrypted with the private key of the card thus providing authenticity and integrity

$$\underline{Enc}_{CSK}(Res) = ERes \tag{6.6}$$

12. The encrypted response is then sent back to the vendor;

13. The vendor decrypts the $ERes$ in two steps

$$\underline{Dec}_{CPK}(ERes) = Res \tag{6.7}$$

$$Dec_{Salt}(Res) = CreditVal \tag{6.8}$$

14. Finally the content of the credit is decrypted with the public key of the bank/card issuer

$$\underline{Dec}_{BPK}(CreditVal) = FRes \tag{6.9}$$

15. If the credit value is correct, a new entry is stored in the storage device of the vendor after having being encrypted with the private key of the vendor.

If all the steps are accomplished without errors the transaction is authorized and the purchase is allowed. It is important to highlight that, as already described in Chapter 5.4, FORCE has been designed as a secure and reliable offline machine to machine authorization scheme rather than as an e-cash system. As such, problems affecting digital currencies, such as digital change, are beyond the scope of the proposed solution and will not be analyzed in this thesis.

Transaction Dispute

Due to its fully offline nature, FORCE does not provide a transaction dispute protocol to better protect both the customer and the vendor as disputes can be turned malicious and thwart the system. Indeed, a malicious customer could simulate an error in the transaction, thus requesting a direct refund to the vendor, whilst a malicious vendor could simulate an invalid transaction, even if digital credits were successfully read from the customer's scratch card. Hence, direct transaction disputes between vendors and customers are avoided whilst online transaction disputes are allowed. In fact, since a further online redemption phase is allowed, the correctness and completeness of each offline transaction can be easily verified by the bank/card issuer thus rendering fake dispute attempts too risky.

Redemption Step

Vendors accepting FORCE scratch cards from their customers can verify digital credits at run-time without relying on any TTP. This is due to the fact that what is actually exchanged between the customer and the vendor is not a promissory note (as with credit cards and all other postponed payment schemes that claim to be offline) but it is a digital value, representing real money and signed by the bank/card issuer. As such, each FORCE payment transaction just needs the pairing and the payment phases in order to be accomplished and evaluated by the vendor. However, for the sake of completeness, the redemption phase is introduced here.

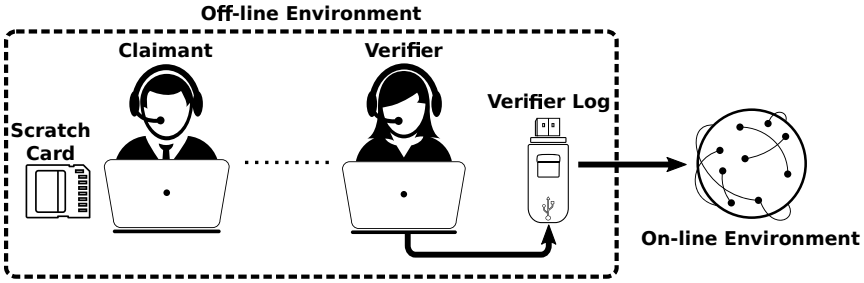

Figure 6.11: Possible uses of digital credit obtained in past transactions

As shown in Figure 6.11, once the offline transaction has been completed, the vendor owns the digital credit just received from the customer. Such credit is encrypted by the bank/card issuer and, as such, it can be easily verified by everyone using the public key of the bank/card issuer. Thus, once the credit has been verified, the vendor can use the digital coin (encapsulated within the credit) either to send it back to the bank/card issuer in exchange for real money or to use it as a common cryptographic currency. If the vendor chooses to send it back to the bank/card issuer, the credit and the coins will be stored in the bank database. On the contrary, if the vendor decides to use the credit as an e-cash digital coin, the credit will be broadcast over the network depending on the payment platform being used.

Each time the vendor decides to spend a credit, it first encrypts it with his private key and then with the public key of the targeted vendor in order to provide authenticity and confidentiality. Once the credit has been received by the other vendor, this latter can easily verify that the credit has been created by the bank/card issuer and that nobody else spent it in the past. However, this *second-step* payment process relies on common online payment protocols and will not be discussed here as it is beyond the aim of this thesis.

Symbol	Meaning
Enc()/Dec()	Symmetric encryption/decryption
Enc()/Dec()	Asymmetric encryption/decryption
Salt	Salt value
IePK	Identity element public key
IeSK	Identity element secret (private) key
CePK	Coin element public key
CeSK	Coin element secret (private) key
BPK	Bank/Element Issuer public key
BSK	Bank/Element Issuers secret (private) key
VPK	Vendor public key
VSK	Vendor secret (private) key

Table 6.3: Symbols used in the transaction protocol

FRoDO: Payment Protocol

We will now describe FRoDO authentication protocol when applied to mobile payments. FRoDO as FORCE stores digital credits within its disposable tokens but, unlike the read-once memory exploited by FORCE, in FRoDO digital credits are never stored within the customer device but are always computed on-the-fly by challenging an erasable-PUF. Unlike FORCE, here the payment protocol is split in two procedures as we have two secure elements belonging to the customer. Hence, while an initial communication is carried by the customer and the vendor, the second one is internal to the customer's device.

Payment Phase

The first step in the payment protocol is between the customer and the vendor and is composed by the following operations (symbols in Table 6.3):

1. The customer sends a purchase request to the vendor asking for some goods;

2. The vendor first creates a random salt value. Then, it encrypts the coin request for three times. The first time with the salt itself. The second time with the public key of the identity element (i.e. the public key of the customer device that is going to receive this request), and the last time with the private key of the vendor itself. Thus, operations performed by the vendor are the following:

$$Enc_{Salt}(Req) = CReq \tag{6.10}$$

$$\underline{Enc}_{IePK}(CReq, Salt) = EncReq \tag{6.11}$$

$$\underline{Enc}_{VSK}(EncReq) = PrivateReq \tag{6.12}$$

3. Once the private request has been built, it is sent to the customer;

4. When the customer receives such a request, first the private key of the identity element is computed by the key generator in the identity element. Then, all the encryption layers computed by the vendor are removed. As such, the customer computes three decryption operations. The first one with the public key of the vendor. The second one with the private key of the identity element and the last one with the salt value. Details follow:

$$\underline{Dec}_{VPK}(PrivateReq) = EncReq \tag{6.13}$$

$$\underline{Dec}_{IeSK}(EncReq) = (CReq, Salt) \tag{6.14}$$

$$Dec_{Salt}(CReq) = Req \tag{6.15}$$

5. Once the coin request is in plain-text, the value of the coin is retrieved from the coin element. Then, such a value computed by the erasable PUF and the coin reconstructor is first encrypted with the salt, then with the private key of the identity element and at the end with the public key of the vendor – to ensure that only the right vendor device can decrypt it. That is:

$$Enc_{Salt}(CoinValue) = CValue \tag{6.16}$$

$$\underline{Enc}_{IeSK}(CValue) = EncValue \tag{6.17}$$

$$\underline{Enc}_{VPK}(EncValue) = PrivateResponse \tag{6.18}$$

6. When the vendor finally receives the *PrivateResponse* value, the last step only requires the coin just read to be validated. Then, the whole payment transaction can be authorized and committed. This is done by firstly decrypting the received response with the private key of the vendor and then by decrypting the value obtained with the public key of the identity element. Then, the salt is used to obtain the value read from the erasable PUF and, as a final step, the public key of the

bank/coin element issuer is used to decrypt the CoinValue thus obtaining the coin raw data built by the bank/card issuer at manufacturing time:

$$\underline{Dec}_{VPK}(PrivateResponse) = EncValue \tag{6.19}$$

$$\underline{Dec}_{IePK}(EncValue) = CValue \tag{6.20}$$

$$Dec_{Salt}(CValue) = CoinValue \tag{6.21}$$

$$\underline{Dec}_{BPK}(CoinValue) = RawValue \tag{6.22}$$

7. If the raw value of the read coin is correct, a new entry is stored in the vendor's storage device after being encrypted with the vendor's private key. It is important to stress that *CoinValue* is not a raw representation of the coin, but it is encrypted at manufacturing time by the bank with its private key. This means that it is not possible to forge digital coins. Indeed, the whole transaction will be validated if and only if the decryption of *CoinValue* with the public key of the bank is successful. If, and only if, this signature will be validated by the vendor then the whole transaction will be validated as well.

Now that all messages exchanged between the customer and the vendor device have been introduced, it is possible to show how the identity and the coin elements interact with each other:

1. Once the identity element has decrypted the coin request received by the vendor, it has to start a customer device internal protocol that allows the identity element to read a coin from the coin element. The first operation is the encryption of the coin request with the private key of the identity element. This provides authenticity for the message that will be received by the coin element. Then, such a private request (for short, PrReq) is encrypted with the public key of the coin element in order to mitigate MITM attacks between the identity element and the coin element as follow:

$$\underline{Enc}_{IeSK}(Req) = PrReq \tag{6.23}$$

$$\underline{Enc}_{CePK}(PrReq) = SecureRequest \tag{6.24}$$

2. The newly encrypted coin request is sent to the coin element;

3. Once the coin request is received by the coin element, the first operation is the retrieval of the coin element private key. As for the identity element, the coin element uses its embedded ID as a challenge to the PUF that will response with the private key in output;

4. When the private key of the coin element has been computed, it is possible to first decrypt the request received by the identity element and then decrypt the obtained output using the public key of the identity element. This ensures message authenticity and integrity:

$$\underline{Dec}_{CeSK}(SecureRequest) = PrReq \tag{6.25}$$

$$\underline{Dec}_{IePK}(PrReq) = Req \tag{6.26}$$

5. Such a request is then used to challenge the erasable PUF embedded into the coin element, as described in Chapter 5.4. All the involved operations are the following:

$$SelectCoinSeed(Req) = PUFChallenge \tag{6.27}$$

$$ReadCoin(PUFChallenge) = PartialCoin \tag{6.28}$$

$$Reconstruct(PartialCoin, CoinHelper) = CoinValue \tag{6.29}$$

6. The coin value has now to be encrypted twice. The first encryption layer is needed in order to prove the authenticity of the coin. The second encryption layer is needed such that only the right identity element will be able to read it:

$$\underline{Enc}_{CeSK}(CoinValue) = EncCoin \tag{6.30}$$

$$\underline{Enc}_{IePK}(EncCoin) = FinalCoin \tag{6.31}$$

7. When the encrypted coin has been received by the identity element, these two encryption layers are removed:

$$\underline{Dec}_{IeSK}(FinalCoin) = EncCoin \tag{6.32}$$

$$\underline{Dec}_{CePK}(EncCoin) = CoinValue \tag{6.33}$$

8. Now the identity element owns the coin value read from the erasable PUF. In order to complete the transaction, this value is sent back to the vendor device as shown in the previous description of the protocol.

If all the above steps are accomplished without errors, the transaction is authorized and the purchase is allowed. It is important to highlight that FRoDO, as already specified for FORCE, has been designed as a secure and reliable encapsulation scheme rather than as an e-cash system. As such, problems affecting digital currencies, such as digital change, are beyond the scope of the proposed solution.

Redemption Phase

As in FORCE, even in FRoDO digital credits can be either sent back to the bank or used in online circuits. However, as the redemption step involves the vendor rather than the customer, and that there is no difference in the vendor's architecture between FORCE and FRoDO, the redemption step is exactly the same and thus, it is not discussed here (see Section 6.1).

Part IV

Conclusions

Chapter 7

Conclusions and Future Work

The final goal of this thesis has been the analysis and improvement of *trust* in the upcoming Internet of Things. Our analysis was particularly focused on the concepts of *identity* and *authenticity* in order to design new protocols and systems capable of ensuring a self-enforced environment without the need of any human control. In fact, the main change in the Internet of Things will be the lack of the human factor. We are nowadays used to control our devices by ourselves. Each time we need to connect or to interact with devices we use passwords, PIN codes or any other factor that puts us in the position of control. However, in the Internet of Things devices ranging from powerful machines to sensors will need to cooperate and to interact with each other in an autonomous way, thus avoiding the human control.

Machines able to decide for themselves scare people mainly due to the fact that in the digital world it is hard to recognized what is authentic and what is fake. This aspect thwart the whole concept of trust as being unable to detect counterfeiting objects makes any security policy useless. To fight against the problem of counterfeiting, in the past, many different approaches have been proposed such as public key infrastructures, trust anchors, trusted computing platform, etc. All of them have shown excellent qualities in strengthening the security of certain systems. However, none of them has proven to be perfect thus requiring additional efforts in the study and improvement of this topic.

In this thesis we have analyzed the authentication problem from both an

online and offline perspective. On one hand, cloud-based online authentication approaches have been improved here by adding a *context* factor and a *continuity* factor. In fact, the solutions introduced in this thesis for online authentication leverage on the surrounding environment to prove their authenticity and are able to authenticate each device even with the presence of sybil devices. On the other hand, offline authentication protocols have been improved as well. While we already have different offline approaches for the authentication of devices, the state of the art lacks a solution for disposable authentication, i.e. the process in which each interaction has to be unique and unrepeatable. This kind of authentication is of paramount importance for payment applications or e-health scenarios where each interaction between the claimant and the verifier has to be controlled.

Nowadays few solutions are already able to provide such an authentication approach but either leverage on trusted third parties or make strong assumptions thus rendering their solution not feasible in the real world. In this thesis we have proposed two versions of a fully offline, disposable authentication protocol that is based on hardware intrinsic properties thus providing secure and reliable one-time interactions even in the presence of an ubiquitous adversary.

Regardless of the specific approach being used, authentication is the process in which a claimant willing to be recognized sends evidences that can be used to verify its identity. Hence, even though an authentication protocols guarantees to a verifier that the claimant it is interacting with is the claimant X rather than the claimant Y this does not say anything about the trustworthiness of X and Y. It is then needed to design and to assign trusted identities to our devices. Usually this is accomplished with cryptographic keys, digital fingerprints and in the last few years on a promising technology called blockchain. These approaches have pros and cons but all of them usually rely on *physical security* assumptions. As such, as long as our devices are physically protected against adversaries (as in the Cloud) we can rely on these approaches but in the Internet of Things, device tampering will be easy to accomplish and we will soon need better solutions.

In this thesis we have proposed a new identity approach based on the *smart contracts* technology and aimed at providing a context aware and attribute-based solution. Different from other approaches, our solution does not force manufacturers in changing their ID syntax but provides a name and discovery process that is distributed and self-enforced.

7.1 Future Work

The identification and authentication solutions provided in this thesis improve over the state of the art and show how an Internet of Trust is possible. However, these solutions are not perfect and, since attackers get more powerful over time as their knowledge increase, our solutions still need some improvements and other approaches have to be analyzed.

The main goal of giving identities to our devices is to be able to recognize them against malicious devices. Our solution based on CONtextual NamE disCovery and resolving with Transactional security is, to the best of our knowledge, the first solution that solve the problem of name and discovery within the Internet of Things. CONNECT leverage on a context-aware and attribute-based solution to assign dynamic names to our devices. However, as also explained in this thesis, to avoid the overloading of the network, things do not broadcast their messages each time they need to interact with other things but rather use a local routing table. However, in the Internet of Things and in the following Internet of Everything each device will be surrounded and will need to interact with billions of other devices thus requiring the analysis of other forms of interaction.

As regards device authentication, we have proposed here different solutions that improve the state of the art from both an online and offline perspective. On one hand, for online scenarios we have proposed a solution based on the context and another one based on meta-data synchronization. However both can be improved as:

- **CONNECT**: our context-aware approach makes it possible to detect sybil attackers when the majority of the network is not malicious. However, when the majority of the surrounding things is malicious or when we need to move a device within a new network for which we do not have any information, our context-aware approach might not work. Last but not least, CONNECT is now based on a two step authentication protocol where the first step is manual whilst the second is autonomous. As a future work we are working on a single step authentication protocol that completely removes the human factor;
- **SUF**: our Software-based Unclonable Functions showed to be able to bring hardware intrinsic security properties such as unclonability, tamper evidence and unpredictability to a software-based authentication protocol. However, it relies on a standard client-server model and as-

sumes an ubiquitous and persistent attacker as practically unfeasible. As a future work we are now working on the design of a distributed model where multiple clients and server can cooperate and where each client is assumed to be malicious.

On the other hand, we have also proposed in this thesis two offline authentication approaches. Our solutions named FORCE and FRoDO, to the best of our knowledge, are the first ones able to exploit hardware intrinsic properties for disposable and event-based authentication protocols. However, as for the other solutions based on the same technology, FORCE and FRoDO are static in nature. The static nature of physical unclonable functions has been studied in the last few years and solutions have already been published aimed at designing *reconfigurable* physical unclonable functions. However, the reconfiguration of such functions can be achieved in three different ways by leveraging on (a) a software layer, (b) a hypervisor or by (c) designing hardware reconfigurable functions. We are actually working on build virtualized physical functions.

Bibliography

Each reference indicates the pages where it appears.

[1] M. Conti, D. Sajal K., B. Chatschik, K. Mohan, N. Lionel M., P. Andrea, R. George, T. Gerhard, T. Gene, and Z. Franco, "Looking ahead in pervasive computing: Challenges and opportunities in the era of cyber-physical convergence," *Pervasive and Mobile Computing*, vol. 8, no. 1, pp. 2 – 21, 2012. 3

[2] R. Poovendran, "Cyber-physical systems: Close encounters between two parallel worlds [point of view]," *Proceedings of the IEEE*, vol. 98, no. 8, pp. 1363–1366, Aug 2010. 3

[3] P. Kyeong, "Importance of staratification on mixing and transport in a shallow, micro-tidal northern gulf of Mexico estuary," in *OCEANS, 2012 - Yeosu*, May 2012, pp. 1–7. 3

[4] H. Junbeom and K. Kyungtae, "Dependable and secure computing in medical information systems," *Computer Communications*, vol. 36, no. 1, pp. 20 – 28, 2012. 3

[5] K. Sampigethaya, R. Poovendran, and L. Bushnell, "Secure operation, control, and maintenance of future e-enabled airplanes," *Proceedings of the IEEE*, vol. 96, no. 12, pp. 1992–2007, Dec 2008. 3

[6] B.-T. Santiago, F.-C. Tiago M., P.-I. Héctor J., and E. Carlos J., "Real-time personal protective equipment monitoring system," *Computer Communications*, vol. 36, no. 1, pp. 42 – 50, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366412000060 3

[7] M. James, C. Michael, B. Jacques, D. Richard, B. Peter, and M. Alex, "Disruptive technologies: Advances that will transform life, business, and the global economy," McKinsey Global Institute, May 2013. 3

[8] E. Borgia, "The Internet of Things vision: Key features, applications and open issues," *Computer Communications*, vol. 54, pp. 1 – 31, 2014. 4

[9] G. Kiev, T. Lionel, and D. Didier, "Combining heterogeneous service technologies for building an Internet of Things middleware," *Computer Communications*, vol. 35, no. 4, pp. 405 – 417, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366411003586 4

[10] M. Conti, "Computer communications: Present status and future challenges," *Computer Communications*, vol. 37, pp. 1 – 4, 2014. 4

[11] C. Mack, "Keynote: Moore's law 3.0," in *IEEE Workshop on Microelectronics and Electron Devices*, April 2013, pp. xiii–xiii. 4

[12] A. Sangiovanni-Vincentelli, "Let's get physical: adding physical dimensions to cyber systems," Internet of Everything Summit, Rome, 2014. 4

[13] E. Dave, "How the next evolution of the Internet is changing everything," in *The Internet of Things*, Apr 2011, White Paper. 4

[14] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update, 2014 & 2019," in *Visual Networking Index (VNI)*, 2015, White Paper. 5

[15] Z. Khan, A. Anjum, and S. Kiani, "Cloud based Big Data analytics for smart future cities," in *IEEE/ACM 6th International Conference on Utility and Cloud Computing*, Dec 2013, pp. 381–386. 5

[16] A.-T. Fadi M., H. Hossam S., and I. Mohamed A., "Efficient deployment of wireless sensor networks targeting environment monitoring applications," *Computer Communications*, vol. 36, no. 2, pp. 135 – 148, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366412003106 5

[17] S. Sina Tayarani-Bathaie, Z. N. Sadough Vanini, and K. Khorasani, "Dynamic neural network-based fault diagnosis of gas turbine engines," *Neurocomputing*, vol. 125, pp. 153–165, Feb. 2014. 5

[18] R. Chakravorty, "A programmable service architecture for mobile medical care," in *Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops*, 2006, pp. 5 pp.–536. 5

[19] Z. M. Aljazzaf, M. Perry, and M. A. M. Capretz, "Online trust: Definition and principles," in *Proceedings of the 2010 Fifth International Multi-conference on Computing in the Global Information Technology*, ser. ICCGI '10. IEEE Computer Society, 2010, pp. 163–168. [Online]. Available: http://dx.doi.org/10.1109/ICCGI.2010.17 6

[20] B. Michele and A. Karpow, "Watch and be watched: Compromising all smart TV generations," in *11th IEEE Consumer Communications and Networking Conference*, Jan 2014, pp. 351–356. 7

[21] A. Al-Anwar, Y. Alkabani, M. El-Kharashi, and H. Bedour, "Hardware trojan protection for third party IPs," in *Euromicro Conference on Digital System Design*, Sept 2013, pp. 662–665. 8

[22] M. Myers, "Can Yahoo recycle your username and protect your data?" CNET - Tech Culture, Online, 2013. 11

[23] C. Johnston, "Time's up for IPv4 as North America runs out of addresses," The Guardian - Online, July 2015. 11

[24] L. Meonghun, H. Jeonghwan, and Y. Hyun, "Agricultural production system based on IoT," in *IEEE 16th International Conference on Computational Science and Engineering*, Dec 2013, pp. 833–837. 11

[25] D. Conzon, P. Brizzi, P. Kasinathan, C. Pastrone, F. Pramudianto, and P. Cultrona, "Industrial application development exploiting IoT vision and model driven programming," in *18th International Conference on Intelligence in Next Generation Networks*, Feb 2015, pp. 168–175. 11

[26] Y. Geng, X. Li, M. Mantysalo, Z. Xiaolin, P. Zhibo, X. Li Da, S. Kao-Walter, C. Qiang, and Z. Li-Rong, "A health-IoT platform based on the integration of intelligent packaging, unobtrusive bio-sensor, and intelligent medicine box," *Industrial Informatics, IEEE Transactions on*, vol. 10, no. 4, pp. 2180–2191, Nov 2014. 11

[27] X. Teng, J. Wendt, and M. Potkonjak, "Security of IoT systems: Design challenges and opportunities," in *IEEE/ACM International Conference on Computer-Aided Design*, Nov 2014, pp. 417–423. 11

[28] J. Granjal, E. Monteiro, and J. Silva, "Security for the Internet of Things: A survey of existing protocols and open research issues," *IEEE Communications Surveys Tutorials*, 2015. 11

[29] A. Riahi, E. Natalizio, Y. Challal, N. Mitton, and A. Iera, "A systemic and cognitive approach for IoT security," in *International Conference on Computing, Networking and Communications*, Feb 2014, pp. 183–188. 11

[30] H. Pohls, V. Angelakis, S. Suppan, K. Fischer, G. Oikonomou, E. Tragos, R. Diaz Rodriguez, and T. Mouroutis, "RERUM: Building a reliable IoT upon privacy- and security- enabled smart objects," in *IEEE Wireless Communications and Networking Conference Workshops*, April 2014, pp. 122–127. 11

[31] S. Sahraoui and A. Bilami, "Compressed and distributed host identity protocol for end-to-end security in the IoT," in *Fifth International Conference on Next Generation Networks and Services*, May 2014, pp. 295–301. 11

[32] J. Shah and J. Parvez, "An examination of next generation IP migration techniques: Constraints and evaluation," in *International Conference on Control, Instrumentation, Communication and Computational Technologies*, July 2014, pp. 776–781. 12, 15

[33] Z. Kuan, L. Xiaohui, L. Rongxing, and S. Xuemin, "Sybil attacks and their defenses in the Internet of Things," *Internet of Things Journal, IEEE*, vol. 1, no. 5, pp. 372–383, Oct 2014. 13, 16, 76, 96

[34] J. Edwards. (2013) Facebook targets 76 million fake users in war on bogus accounts. [Online]. Available: http://www.businessinsider.com/facebook-targets-76-million-fake-users-in-war-on-bogus-accounts-2013-2?IR=T 16

[35] D. Coppersmith, D. B. Johnson, and S. Matyas, "A proposed mode for triple-DES encryption," *IBM Journal of Research and Development*, vol. 40, no. 2, pp. 253–262, 1996. 21

[36] A. Boukerche, Y. Ren, and L. Mokdad, "Applying symmetric and asymmetric key algorithms for the security in wireless networks: proof of correctness," in *Proceedings of the 6th ACM workshop on QoS and security for wireless and mobile networks*, ser. Q2SWinet '10. New York, NY, USA: ACM, 2010, pp. 33–40. 21

[37] H. Orman and P. Hoffman, "Determining strengths for public keys used for exchanging symmetric keys," United States, 2004. 21

[38] M. E. Hellman, "An overview of public key cryptography," *IEEE Communications Magazine*, vol. 16, no. 6, pp. 24–32, November 1978. 21

[39] R. C. Merkle, "Secure communications over insecure channels," *Commun. ACM*, vol. 21, no. 4, pp. 294–299, apr 1978. 21

[40] C. Decker and R. Wattenhofer, "Information propagation in the Bitcoin network," in *IEEE Thirteenth International Conference on Peer-to-Peer Computing*, Sept 2013, pp. 1–10. 24

[41] G. Zyskind, O. Nathan, and A. Pentland, "Decentralizing privacy: Us-

ing blockchain to protect personal data," in *IEEE Security and Privacy Workshops (SPW)*, May 2015, pp. 180–184. 24, 47

[42] S. Martins and Y. Yang, "Introduction to Bitcoins: a pseudo-anonymous electronic currency system," in *Proceedings of the 2011 Conference of the Center for Advanced Studies on Collaborative Research*, ser. CASCON '11. Riverton, NJ, USA: IBM Corp., 2011, pp. 349–350. 24

[43] A. Shoufan and N. Huber, "A fast hash tree generator for Merkle signature scheme," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, May 2010, pp. 3945–3948. 25

[44] I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, "Proof of activity: Extending Bitcoin's proof of work via proof of stake," *SIGMETRICS Perform. Eval. Rev.*, vol. 42, no. 3, pp. 34–37, Dec 2014. 28, 48

[45] A. Beikverdi and S. JooSeok, "Trend of centralization in Bitcoin's distributed network," in *16th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, June 2015, pp. 1–6. 31

[46] S. Omohundro, "Cryptocurrencies, smart contracts, and artificial intelligence," *AI Matters*, vol. 1, no. 2, pp. 19–21, Dec 2014. 31

[47] M. Youssfi, O. Bouattane, J. Bakkoury, and M. Bensalah, "A new massively parallel and distributed virtual machine model using mobile agents," in *International Conference on Multimedia Computing and Systems*, April 2014, pp. 407–414. 32

[48] A.-R. Sadeghi and D. Naccache, Eds., *Towards Hardware-Intrinsic Security - Foundations and Practice*, ser. Information Security and Cryptography. Springer, 2010. 34, 39

[49] K. M. Tolk, *Reflective particle technology for identification of critical components*. Sandia National Laboratories, Jan 1992. [Online]. Available: http://www.osti.gov/scitech/servlets/purl/7116334 35

[50] N. R. Council, *Counterfeit Deterrent Features for the Next-Generation Currency Design*. The National Academies Press, 1993. 35

[51] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical one-way functions," *Science*, vol. 297, no. 5589, pp. 2026–2030, 2002. 35

[52] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," in *Proceedings of the 9th ACM conference on Computer and communications security*, ser. CCS '02. New York, NY, USA: ACM, 2002, pp. 148–160. 35

[53] P. S. Ravikanth, "Physical one-way functions," Ph.D. dissertation,

Massachusetts institute of technology, 2001, aAI0803255. 35, 118

[54] P. Koeberl, L. Jiangtao, A. Rajan, and W. Wei, "Entropy loss in PUF-based key generation schemes: The repetition code pitfall," in *IEEE International Symposium on Hardware-Oriented Security and Trust*, May 2014, pp. 44–49. 39

[55] S. Katzenbeisser, U. Kocabaş, V. Rožić, A.-R. Sadeghi, I. Verbauwhede, and C. Wachsmann, "PUFs: myth, fact or busted? a security evaluation of physically unclonable functions (PUFs) cast in silicon," in *Proceedings of the 14th international conference on Cryptographic Hardware and Embedded Systems*, ser. CHES'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 283–301. 40

[56] J. Herbert and A. Litchfield, "A novel method for decentralised peer-to-peer software license validation using cryptocurrency blockchain technology," in *38th Australasian Computer Science Conference*, ser. CRPIT, D. Parry, Ed., vol. 159. Sydney, Australia: ACS, 2015, pp. 27–35. 47

[57] F. Conner, V. Dragos, and Y. Sophia, "A decentralized public key infrastructure with identity retention," *IACR Cryptology ePrint Archive*, vol. 2014, p. 803, 2014. 48

[58] G. Bela, M. Norman, and G. Andre, "Decentralized trusted timestamping using the crypto currency Bitcoin," *CoRR*, vol. abs/1502.04015, 2015. 48

[59] G. Zyskind, O. Nathan, and A. Pentland, "Enigma: Decentralized computation platform with guaranteed privacy," *CoRR*, vol. abs/1506.03471, 2015. 48

[60] J. van den Hooff, M. F. Kaashoek, and N. Zeldovich, "VerSum: Verifiable Computations over Large Public Logs," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '14, 2014, pp. 1304–1316. 48

[61] K. Crary and M. J. Sullivan, "Peer-to-peer affine commitment using Bitcoin," in *Proceedings of the 36th ACM SIGPLAN Conference on Programming Language Design and Implementation*, ser. PLDI 2015, 2015, pp. 479–488. 48

[62] IBM, "ADEPT: An IoT practitioner perspective," 2015, online Resource. 48, 49

[63] A. White, A. Tickle, and A. Clark, "Overcoming reputation and proof-of-work systems in botnets," in *4th International Conference on Network and System Security*, Sept 2010, pp. 120–127. 48

[64] M. Blackstock and R. Lea, "IoT interoperability: A hub-based approach," in *International Conference on the Internet of Things*, Oct 2014, pp. 79–84. 49

[65] J. H. Ziegeldorf, F. Grossmann, M. Henze, N. Inden, and K. Wehrle, "Coinparty: Secure multi-party mixing of Bitcoins," in *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy*, ser. CODASPY. New York, NY, USA: ACM, 2015, pp. 75–86. 49

[66] S. Sevilla, P. Mahadevan, and J. Garcia-Luna-Aceves, "FERN: A unifying framework for name resolution across heterogeneous architectures," *Computer Communications*, vol. 56, no. 0, pp. 14 – 24, 2015. 50

[67] F. Yang, N. Matthys, R. Bachiller, S. Michiels, W. Joosen, and D. Hughes, "μPnP: plug and play peripherals for the Internet of Things," in *Proceedings of the Tenth European Conference on Computer Systems*, ser. EuroSys, L. Réveillère, T. Harris, and M. Herlihy, Eds. New York, NY, USA: ACM, 2015, pp. 25:1–25:14. 50

[68] F.-C. Chang and D.-K. Chen, "The design of an XMPP-based service integration scheme," in *Proceedings of the 2011 Seventh International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, ser. IIH-MSP. Washington, DC, USA: IEEE Computer Society, 2011, pp. 33–36. 50

[69] S. Cai, M. Bakhouya, M. Becherif, J. Gaber, and M. Wack, "An in-vehicle embedded system for can-bus events monitoring," *J. Mob. Multimed.*, vol. 10, pp. 128–140, May 2014. 50

[70] C. Paetz, *Z-Wave Basics: Remote Control in Smart Homes*. USA: CreateSpace Independent Publishing Platform, 2013. 50

[71] L. Sejun, J. Jaehoon, and P. Jungsoo, "DNS name autoconfiguration for IoT home devices," in *IEEE 29th International Conference on Advanced Information Networking and Applications Workshops*, March 2015, pp. 131–134. 51

[72] L. Yi, W. He, W. Junyu, Q. Kan, K. Ning, W. Kaijiang, S. Yiwei, and Z. Lirong, "Enterprise-oriented IoT name service for agriculture product supply chain management," in *International Conference on Identification, Information and Knowledge in the Internet of Things*, Oct 2014, pp. 237–241. 51

[73] Y. Zhiwei, K. Ning, T. Ye, and P. Yong-Jin, "A universal object name resolution scheme for IoT," in *IEEE International Conference*

on Green Computing and Communications and Internet of Things and Cyber, Physical and Social Computing, Aug 2013, pp. 1120–1124. 51

[74] R. Pozza, M. Nati, S. Georgoulas, A. Gluhak, K. Moessner, and S. Krco, "CARD: Context-aware resource discovery for mobile Internet of Things scenarios," in *IEEE 15th International Symposium on A World of Wireless, Mobile and Multimedia Networks*, June 2014, pp. 1–10. 51

[75] M. Antonini, S. Cirani, G. Ferrari, P. Medagliani, M. Picone, and L. Veltri, "Lightweight multicast forwarding for service discovery in low-power IoT networks," in *22nd International Conference on Software, Telecommunications and Computer Networks*, Sept 2014, pp. 133–138. 51

[76] C. Sejin, S. Seungmin, O. Byungkook, and L. Kyong-Ho, "Semantic description, discovery and integration for the Internet of Things," in *IEEE International Conference on Semantic Computing*, Feb 2015, pp. 272–275. 51

[77] G. Tanganelli, E. Mingozzi, C. Vallati, and C. Cicconetti, "A distributed architecture for discovery and access in the Internet of Things," in *IEEE Conference on Computer Communications Workshops*, April 2013, pp. 45–46. 51

[78] R. Kolcun and J. McCann, "Dragon: Data discovery and collection architecture for distributed IoT," in *International Conference on the Internet of Things*, Oct 2014, pp. 91–96. 51

[79] B. Villaverde, R. De Paz Alberola, A. Jara, S. Fedor, S. Das, and D. Pesch, "Service discovery protocols for constrained machine-to-machine communications," *Communications Surveys Tutorials, IEEE*, vol. 16, no. 1, pp. 41–60, First 2014. 51

[80] A. Jara, P. Lopez, D. Fernandez, J. Castillo, M. Zamora, and A. Skarmeta, "Mobile digcovery: A global service discovery for the Internet of Things," in *27th International Conference on Advanced Information Networking and Applications Workshops*, March 2013, pp. 1325–1330. 51

[81] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the Internet of Things: A survey," *Communications Surveys Tutorials, IEEE*, vol. 16, no. 1, pp. 414–454, First 2014. 51

[82] IRM WG, Kantara Initiative discussion group, "The Design Principles of Relationship Management," Feb 2015. 52

[83] C.-J. M. Liang, B. F. Karlsson, N. D. Lane, F. Zhao, J. Zhang, Z. Pan, Z. Li, and Y. Yu, "SIFT: Building an Internet of Safe Things," in *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*, ser. IPSN. New York, NY, USA: ACM, 2015, pp. 298–309. 54

[84] A. Gianluca, B. Luca, B. Luciano, B. Orazio, D. F. Marco, L. Valeria, P. Pasquale, P. Fabio, R. Giuseppe, and T. Angelo, "STEM-NET: How to deploy a self-organizing network of mobile end-user devices for emergency communication," *Computer Communications*, vol. 60, no. 0, pp. 12 – 27, 2015. 55

[85] C. Boyd and A. Mathuria, *Protocols for Authentication and Key Establishment*, 1st ed. Springer Publishing Company, Incorporated, 2010. 61

[86] A. J. Menezes, S. A. Vanstone, and P. C. V. Oorschot, *Handbook of Applied Cryptography*, 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 1996. 61

[87] B. Guttman and E. A. Roback, "Sp 800-12. an introduction to computer security: The nist handbook," National Institute of Standards and Technology, Gaithersburg, MD, United States, Tech. Rep., 1995. 61

[88] M. Kakihara, "Grasping a global view of smartphone diffusion: An analysis from a global smartphone study," in *2014 International Conference on Mobile Business*, 2014. 63

[89] J. Tang, V. Terziyan, and J. Veijalainen, "Distributed pin verification scheme for improving security of mobile devices," *Mob. Netw. Appl.*, vol. 8, no. 2, pp. 159–175, Apr. 2003. [Online]. Available: http://dx.doi.org/10.1023/A:1022289231864 63

[90] I. Jermyn, A. Mayer, F. Monrose, M. K. Reiter, and A. D. Rubin, "The design and analysis of graphical passwords," in *Proceedings of the 8th Conference on USENIX Security Symposium - Volume 8*, ser. SSYM'99. Berkeley, CA, USA: USENIX Association, 1999, pp. 1–1. [Online]. Available: http://dl.acm.org/citation.cfm?id=1251421.1251422 63

[91] S. Gold, "Android insecurity," *Network Security*, vol. 2011, no. 10, pp. 5–7, 2011. [Online]. Available: http://dx.doi.org/10.1016/S1353-4858(11)70104-0 64

[92] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith, "Smudge attacks on smartphone touch screens," in *Proceedings of the*

4th USENIX Conference on Offensive Technologies, ser. WOOT'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 1–7. [Online]. Available: http://dl.acm.org/citation.cfm?id=1925004.1925009 64

[93] S. Kwang Il, P. Ji Soo, L. Jae Yong, and P. Jong Hyuk, "Design and implementation of improved authentication system for Android smartphone users," in *26th International Conference on Advanced Information Networking and Applications Workshops*, March 2012, pp. 704–707. 64

[94] Y.-L. Zhang, J. Yang, and H.-T. Wu, "Sweep fingerprint sequence reconstruction for portable devices," *Electronics Letters*, vol. 42, no. 4, pp. 204–205, Feb 2006. 64

[95] D. M. Monro, S. Rakshit, and D. Zhang, "Dct-based iris recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 4, pp. 586–595, Apr 2007. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2007.1002 64

[96] O. Mazhelis, J. Markkula, and J. Veijalainen, "An integrated identity verification system for mobile terminals," *Inf. Manag. Comput. Security*, vol. 13, no. 5, pp. 367–378, 2005. [Online]. Available: http://dx.doi.org/10.1108/09685220510627269 64

[97] N. Clarke, S. Karatzouni, and S. Furnell, "Flexible and transparent user authentication for mobile devices," in *Emerging Challenges for Security, Privacy and Trust*, ser. IFIP Advances in Information and Communication Technology, D. Gritzalis and J. Lopez, Eds. Springer Berlin Heidelberg, 2009, vol. 297, pp. 1–12. 64

[98] M. Derawi, P. Bours, and K. Holien, "Improved cycle detection for accelerometer based gait authentication," in *Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, Oct 2010, pp. 312–317. 64

[99] M. Conti, I. Zachia-Zlatea, and B. Crispo, "Mind how you answer me!: Transparently authenticating the user of a smartphone when answering or placing a call," in *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, ser. ASIACCS '11. New York, NY, USA: ACM, 2011, pp. 249–259. [Online]. Available: http://doi.acm.org/10.1145/1966913.1966945 64, 66

[100] S. Weidong, Y. Jun, J. Yifei, Y. Feng, and X. Yingen, "Senguard: Passive user identification on smartphones using multiple sensors," in *IEEE 7th International Conference on Wireless and Mobile Comput-*

ing, Networking and Communications, Oct 2011, pp. 141–148. 64

[101] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "User evaluation of lightweight user authentication with a single tri-axis accelerometer," in *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, ser. MobileHCI '09. New York, NY, USA: ACM, 2009, pp. 15:1–15:10. [Online]. Available: http://doi.acm.org/10.1145/1613858.1613878 67

[102] L. Chien-Cheng, L. Deron, C. Chin-Chun, and Y. Ching-Han, "A new non-intrusive authentication method based on the orientation sensor for smartphone users," in *IEEE Sixth International Conference on Software Security and Reliability*, June 2012, pp. 245–252. 67, 68

[103] F. Steven, C. Nathan, and K. Sevasti, "Beyond the pin: Enhancing user authentication for mobile devices," *Computer Fraud and Security*, vol. 2008, no. 8, pp. 12 – 17, 2008. 67

[104] C. Nickel, T. Wirtl, and C. Busch, "Authentication of smartphone users based on the way they walk using k-NN algorithm," in *Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, July 2012, pp. 16–20. 68

[105] C. Nickel, H. Brandt, and C. Busch, "Benchmarking the performance of SVMs and HMMs for accelerometer-based biometric gait recognition," in *IEEE International Symposium on Signal Processing and Information Technology*, Dec 2011, pp. 281–286. 68

[106] C. Stein, C. Nickel, and C. Busch, "Fingerphoto recognition with smartphone cameras," in *Proceedings of the International Conference of the Biometrics Special Interest Group*, Sept 2012, pp. 1–12. 69

[107] P. Fahmi, E. Kodirov, D.-J. Choi, G.-S. Lee, A. Mohd Fikri Azli, and S. Sayeed, "Implicit authentication based on ear shape biometrics using smartphone camera during a call," in *IEEE International Conference on Systems, Man, and Cybernetics*, 2012, pp. 2272–2276. 69

[108] S. Azenkot, K. Rector, R. Ladner, and J. Wobbrock, "PassChords: Secure multi-touch authentication for blind people," in *Proceedings of the 14th International ACM SIGACCESS Conference on Computers and Accessibility*, ser. ASSETS '12, 2012, pp. 159–166. 70

[109] S. Azenkot, J. O. Wobbrock, S. Prasain, and R. E. Ladner, "Input finger detection for nonvisual touch screen text entry in perkinput," in *Proceedings of Graphics Interface 2012*, ser. GI '12, 2012, pp. 121–129. 70

[110] R. Fantacci, L. Maccari, T. Pecorella, and F. Frosali, "Analysis of

secure handover for IEEE 802.1x-based wireless ad hoc networks," *Wireless Communications, IEEE*, vol. 14, no. 5, pp. 21–29, October 2007. 71, 73

[111] C. Jyh-Cheng and W. Yu-Ping, "Extensible authentication protocol (EAP) and IEEE 802.1x: tutorial and empirical experience," *Communications Magazine, IEEE*, vol. 43, no. 12, pp. supl.26–supl.32, Dec 2005. 71

[112] B. Aboba and P. Calhoun, "RADIUS (remote authentication dial in user service) support for extensible authentication protocol (EAP)," Microsoft, United States, Tech. Rep., 2003. 72

[113] IEEE, "Ieee standard for local and metropolitan area networks part 16: Air interface for fixed and mobile broadband wireless access systems amendment 2: Physical and medium access control layers for combined fixed and mobile operation in licensed bands and corrigendum 1," *IEEE Std 802.16e-2005 and IEEE Std 802.16-2004/Cor 1-2005 (Amendment and Corrigendum to IEEE Std 802.16-2004)*, 2006. 72

[114] G. Kambourakis, S. Gritzalis, and J. H. Park, "Device authentication in wireless and pervasive environments," *Intelligent Automation & Soft Computing*, vol. 16, no. 3, pp. 399–418, 2010. 72

[115] A.-R. Sadeghi, M. Selhorst, C. Stüble, C. Wachsmann, and M. Winandy, "Tcg inside?: A note on tpm specification compliance," in *Proceedings of the First ACM Workshop on Scalable Trusted Computing*, ser. STC '06, 2006, pp. 47–56. [Online]. Available: http://doi.acm.org/10.1145/1179474.1179487 72, 73

[116] D. Q. Liu and M. Coslow, "Extensible authentication protocols for ieee standards 802.11 and 802.16," in *Proceedings of the International Conference on Mobile Technology, Applications, and Systems*, ser. Mobility '08, 2008, pp. 47:1–47:9. 73

[117] J. Wayne, G. Serban I., S. Clement, and V. Korolev, "Smart cards and mobile device authentication: an overview and implementation," in *NIST Manuscript Publication- Internal Report (NISTIR) - 7206*, NIST, Ed., 2005. 73

[118] N. Sastry, U. Shankar, and D. Wagner, "Secure verification of location claims," in *Proceedings of the 2Nd ACM Workshop on Wireless Security*, ser. WiSe '03, 2003, pp. 1–10. 73

[119] K. Remley, C. Grosvenor, R. Johnk, D. Novotny, P. Hale, M. Mckinley, A. Karygiannis, and E. Antonakakis, "Electromagnetic signatures of WLAN cards and network security," in *Proceedings of the Fifth*

IEEE International Symposium on Signal Processing and Information Technology, Dec 2005, pp. 484–488. 73

[120] C. Hasan, M. Adibuzzaman, F. Kawsar, M. Haque, and S. Ahamed, "PryGuard: A secure distributed authentication protocol for pervasive computing environment," in *Modern Approaches in Applied Intelligence*, ser. Lecture Notes in Computer Science, K. Mehrotra, C. Mohan, J. Oh, P. Varshney, and M. Ali, Eds. Springer Berlin Heidelberg, 2011, vol. 6703, pp. 135–145. 73

[121] S. Moon, P. Skelly, and D. Towsley, "Estimation and removal of clock skew from network delay measurements," in *IEEE Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 1, Mar 1999, pp. 227–234 vol.1. 73

[122] N. Nam Tuan, Z. Guanbo, H. Zhu, and Z. Rong, "Device fingerprinting to enhance wireless security using nonparametric bayesian method," in *IEEE Proceedings*, April 2011, pp. 1404–1412. 74

[123] N. Patwari and S. K. Kasera, "Robust location distinction using temporal link signatures," in *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking*, ser. MobiCom '07. ACM, 2007, pp. 111–122. [Online]. Available: http://doi.acm.org/10.1145/1287853.1287867 74

[124] D. Sanorita, R. Nirupam, X. Wenyuan, R. C. Romit, and N. Srihari, "AccelPrint: Imperfections of accelerometers make smartphones trackable," in *21st Annual Network and Distributed System Security Symposium*, 2014. 74

[125] L. C. C. Desmond, C. C. Yuan, T. C. Pheng, and R. S. Lee, "Identifying unique devices through wireless fingerprinting," in *Proceedings of the First ACM Conference on Wireless Network Security*, ser. WiSec '08. ACM, 2008, pp. 46–55. [Online]. Available: http://doi.acm.org/10.1145/1352533.1352542 74

[126] J. Franklin, D. McCoy, P. Tabriz, V. Neagoe, J. Van Randwyk, and D. Sicker, "Passive data link layer 802.11 wireless device driver fingerprinting," in *Proceedings of the 15th Conference on USENIX Security Symposium*, ser. USENIX-SS'06, vol. 15. USENIX Association, 2006. [Online]. Available: http://dl.acm.org/citation.cfm?id=1267336.1267348 74

[127] F. Guo and T.-c. Chiueh, "Sequence number-based MAC address spoof detection," in *Proceedings of the 8th International Conference on Recent Advances in Intrusion Detection*, ser. RAID'05. Springer-Verlag,

2006, pp. 309–329. 74

[128] J. Pang, B. Greenstein, R. Gummadi, S. Seshan, and D. Wetherall, "802.11 user fingerprinting," in *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking*, ser. MobiCom '07. ACM, 2007, pp. 99–110. [Online]. Available: http://doi.acm.org/10.1145/1287853.1287866 74

[129] P. Eckersley, "How unique is your web browser?" in *Proceedings of the 10th International Conference on Privacy Enhancing Technologies*, ser. PETS'10. Springer-Verlag, 2010, pp. 1–18. [Online]. Available: http://dl.acm.org/citation.cfm?id=1881151.1881152 74

[130] T. Yen, Y. Xie, F. Yu, R. P. Yu, and M. Abadi, "Host fingerprinting and tracking on the web: Privacy and security implications," in *19th Annual Network and Distributed System Security Symposium, NDSS 2012, San Diego, California, USA, February 5-8, 2012*, 2012. 74

[131] G. Acar, M. Juarez, N. Nikiforakis, C. Diaz, S. Gürses, F. Piessens, and B. Preneel, "FPDetective: Dusting the web for fingerprinters," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, ser. CCS '13. ACM, 2013, pp. 1129–1140. [Online]. Available: http://doi.acm.org/10.1145/2508859.2516674 74

[132] H. Ding-Jie, Y. Kai-Ting, T. Wei-Chung, and C. Ge-Ming, "Design of client device identification by clock skew in clouds," in *IEEE International Conference on Automation Science and Engineering*, Aug 2014, pp. 1133–1138. 74

[133] N. Nikiforakis, A. Kapravelos, W. Joosen, C. Kruegel, F. Piessens, and G. Vigna, "Cookieless Monster: Exploring the Ecosystem of Web-Based Device Fingerprinting," in *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, ser. SP '13, 2013, pp. 541–555. 74

[134] A. Das, N. Borisov, and M. Caesar, "Do you hear what i hear?: Fingerprinting smart devices through embedded acoustic components," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '14. New York, NY, USA: ACM, 2014, pp. 441–452. [Online]. Available: http://doi.acm.org/10.1145/2660267.2660325 74

[135] J. François, R. State, T. Engel, and O. Festor, "Enforcing Security with Behavioral Fingerprinting," in *Proceedings of the 7th International Conference on Network and Services Management*, ser. CNSM '11, 2011, pp. 64–72. 74

[136] H. Bojinov, Y. Michalevsky, G. Nakibly, and D. Boneh, "Mobile device

identification via sensor fingerprinting," *CoRR*, vol. abs/1408.1416, 2014. 75

[137] N. Tran, B. Min, J. Li, and L. Subramanian, "Sybil-resilient online content voting," in *Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation*, ser. NSDI'09. Berkeley, CA, USA: USENIX Association, 2009, pp. 15–28. [Online]. Available: http://dl.acm.org/citation.cfm?id=1558977.1558979 76

[138] Y. Reddy, "A game theory approach to detect malicious nodes in wireless sensor networks," in *Third International Conference on Sensor Technologies and Applications*, June 2009, pp. 462–468. 76

[139] G. Wang, T. Konolige, C. Wilson, X. Wang, H. Zheng, and B. Y. Zhao, "You are how you click: Clickstream analysis for sybil detection," in *Proceedings of the 22Nd USENIX Conference on Security*, ser. SEC'13. Berkeley, CA, USA: USENIX Association, 2013, pp. 241–256. [Online]. Available: http://dl.acm.org/citation.cfm?id=2534766.2534788 76

[140] F. Reid and M. Harrigan, "An analysis of anonymity in the Bitcoin system," in *IEEE Third International Conference on Privacy, Security, Risk and Trust (PASSAT) and IEEE Third International Conference on Social Computing (SocialCom)*, Oct 2011, pp. 1318–1326. 80

[141] L. Jun-Ya, L. Wei-Cheng, and H. Yu-Hung, "A lightweight authentication protocol for Internet of Things," in *International Symposium on Next-Generation Electronics*, May 2014, pp. 1–2. 81

[142] V. Roussev, "An evaluation of forensic similarity hashes," *Digital Investigation*, vol. 8, pp. 34–41, 2011. 84

[143] J. Kornblum, "Identifying almost identical files using context triggered piecewise hashing," *Digit. Investig.*, vol. 3, pp. 91–97, Sep 2006. 84

[144] R. Di Pietro, L. Mancini, A. Villani, and D. Vitali, "Uniqueness of the file systems genome: Supporting arguments and massive experimental measurements," in *International Conference on Risks and Security of Internet and Systems*, Oct 2013, pp. 1–8. 89

[145] S. Agafin and A. Krasnopevtsev, "Memory access time as entropy source for rng," in *Proceedings of the 7th International Conference on Security of Information and Networks*, ser. SIN '14. New York, NY, USA: ACM, 2014, pp. 176:176–176:179. [Online]. Available: http://doi.acm.org/10.1145/2659651.2659695 89

[146] A. W. Min, R. Wang, J. Tsai, M. A. Ergin, and T.-Y. C. Tai, "Improving energy efficiency for mobile platforms by exploiting low-power sleep states," in *Proceedings of the 9th Conference on Computing*

Frontiers, ser. CF '12. New York, NY, USA: ACM, 2012, pp. 133–142. [Online]. Available: http://doi.acm.org/10.1145/2212908.2212928 90

[147] H. Will, K. Schleiser, and J. Schiller, "A real-time kernel for wireless sensor networks employed in rescue scenarios," in *IEEE 34th Conference on Local Computer Networks*, 2009, pp. 834–841. 90

[148] Google. (2015) Cloud messaging. https://developers.google.com/cloud-messaging. [Online]. Available: https://developers.google.com/cloud-messaging 91

[149] Y. S. Yilmaz, B. I. Aydin, and M. Demirbas, "Google cloud messaging (GCM): an evaluation," in *IEEE Global Communications Conference*, 2014, pp. 2807–2812. [Online]. Available: http://dx.doi.org/10.1109/GLOCOM.2014.7037233 91

[150] S. Mingoo, S. Hanson, J. Seo, D. Sylvester, and D. Blaauw, "Robust ultra-low voltage ROM design," in *Custom Integrated Circuits Conference*, 2008, pp. 423–426. 96

[151] E. Ebrard, B. Allard, P. Candelier, and P. Waltz, "Review of fuse and antifuse solutions for advanced standard cmos technologies," *Microelectron. J.*, vol. 40, no. 12, pp. 1755–1765, Dec 2009. 96

[152] J.-H. Yoon, "Memory properties of al-based nanoparticle floating gate for nonvolatile memory applications," *Journal of the Korean Physical Society*, vol. 61, no. 5, pp. 799–802, 2012. 96

[153] M. Wu and W. Zwaenepoel, "envy: a non-volatile, main memory storage system," *SIGPLAN Not.*, vol. 29, no. 11, pp. 86–97, Nov 1994. 96, 97

[154] V. Delgado-Gomes, J. Oliveira-Lima, C. Lima, J. Martins, R. Jardim-Goncalves, and V. Fernao Pires, "Energy consumption evaluation to reduce manufacturing costs," in *Fourth International Conference on Power Engineering, Energy and Electrical Drives*, May 2013, pp. 1012–1016. 96

[155] D. Prochnow, *Experiments with EPROMS*. McGraw-Hill Professional, 1988. 96

[156] M. Cappelletti, *Flash Memories*, P. Cappelletti and C. Golla, Eds. Norwell, MA, USA: Kluwer Academic Publishers, 1999. 97

[157] B. E. Kratochvil, L. Dong, and B. J. Nelson, "Real-time rigid-body visual tracking in a scanning electron microscope," *Int. J. Rob. Res.*, vol. 28, no. 4, pp. 498–511, Apr 2009. 97

[158] M. Korosec, J. Duhovnik, and N. Vukasinovic, "Identification and optimization of key process parameters in noncontact laser scanning for

reverse engineering," *Comput. Aided Des.*, vol. 42, no. 8, pp. 744–748, Aug 2010. 97

[159] M. S. N. Murthy, M. G. Jones, J. Kulka, J. D. Davies, M. Halliwell, P. C. Jackson, D. R. Bull, and P. N. T. Wells, "Infrared confocal microscope," in *IEEE Colloquium on New Microscopies in Medicine and Biology*, 1994, pp. 1–2. 97

[160] J. Melngailis, "Focused ion beam technology and applications," *Journal of Vacuum Science Technology B: Microelectronics and Nanometer Structures*, vol. 5, no. 2, pp. 469–495, 1987. 97, 112, 115, 118

[161] G. Hong and J. Bo, "Forensic analysis of skimming devices for credit fraud detection," in *2nd IEEE International Conference on Information and Financial Engineering*, Sept 2010, pp. 542–546. 101

[162] S. Gomzin, *Hacking Point of Sale: Payment Application Secrets, Threats, and Solutions*, 1st ed. Wiley Publishing, 2014. 102

[163] V. Coskun, K. Ok, and B. Ozdenizci, *Near Field Communication: From Theory to Practice*, 1st ed. Wiley Publishing, 2012. 103

[164] B. J. E. V. Rens, "Authentication using a read-once memory," June 2006, accessed: 2013-07-30. 104

[165] M.-D. M. Yu, D. M'Raihi, R. Sowell, and S. Devadas, "Lightweight and secure PUF key storage using limits of machine learning," in *Proceedings of the 13th international conference on Cryptographic hardware and embedded systems*, ser. CHES'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 358–373. 108

[166] D. Lim, J. W. Lee, B. Gassend, G. E. Suh, M. van Dijk, and S. Devadas, "Extracting secret keys from integrated circuits," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 13, no. 10, pp. 1200–1205, oct 2005. 108

[167] P. Choi and D. K. Kim, "Design of security enhanced TPM chip against invasive physical attacks," in *IEEE ISCAS '12*, 2012, pp. 1787–1790. 112, 116

[168] J. Krämer, D. Nedospasov, A. Schlösser, and J.-P. Seifert, "Differential photonic emission analysis," in *Constructive Side-Channel Analysis and Secure Design*, ser. COSADE'13. Berlin, Heidelberg: Springer-Verlag, 2013, pp. 1–16. 114

[169] W. P. Griffin, A. Raghunathan, and K. Roy, "CLIP: Circuit level IC protection through direct injection of process variations," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 20, no. 5, pp. 791–803, may 2012. 116

[170] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas, and J. Schmidhuber, "Modeling attacks on physical unclonable functions," in *Proceedings of the 17th ACM Conference on Computer and Communications Security*, ser. ACM CCS '10. New York, NY, USA: ACM, 2010, pp. 237–249. 118

[171] U. Rührmair, H. Busch, and S. Katzenbeisser, "Strong PUFs: Models, Constructions, and Security Proofs," in *Towards Hardware-Intrinsic Security*, ser. Information Security and Cryptography, A.-R. Sadeghi and D. Naccache, Eds. Springer Berlin Heidelberg, 2010, pp. 79–96. 118

[172] U. Rührmair, C. Jaeger, and M. Algasinger, "An attack on PUF-based session key exchange and a hardware-based countermeasure: Erasable PUFs," in *Financial Cryptography and Data Security*, ser. LNCS, G. Danezis, Ed. Springer, 2012, vol. 7035, pp. 190–204. 119, 121, 128

[173] R. Maes, P. Tuyls, and I. Verbauwhede, "Low-overhead implementation of a soft decision helper data algorithm for SRAM PUFs," in *Proceedings of the 11th International Workshop on Cryptographic Hardware and Embedded Systems*, ser. CHES '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 332–347. 122

[174] P. John, S. Karen, and C. Lily, "Guide to Bluetooth Security," in *NIST Special Publication 800-121, Revision 1*, June 2012. 123

[175] G. Van Damme, K. Wouters, H. Karahan, and e. al, "Offline NFC payments with electronic vouchers," *... applications for mobile ...*, 2009. 125

[176] F. Chun-I, L. Yu-Kuang, and W. Chien-Nan, "An anonymous fair offline micropayment scheme," in *International Conference on Information Society*, June 2011, pp. 377 –381. 125

[177] N. Kiran and G. Kumar, "Building robust m-commerce payment system on offline wireless network," in *IEEE 5th International Conference on Advanced Networks and Telecommunication Systems*, dec. 2011, pp. 1 –3. 125

[178] C.-L. Chen and J.-J. Liao, "Fair offline digital content transaction system," *Information Security, IET*, vol. 6, no. 3, pp. 123–130, Sept 2012. 125

[179] W. Cong, S. Hongxiang, Z. Hua, and J. Zhengping, "An improved off-line electronic cash scheme," in *Fifth International Conference on Computational and Information Sciences*, June 2013, pp. 438–441.

125, 136, 137

[180] C.-I. Fan, V. S.-M. Huang, and Y.-C. Yu, "User efficient recoverable off-line e-cash scheme with fast anonymity revoking," *Mathematical and Computer Modelling*, vol. 58, no. 1ÂŰ2, pp. 227 – 237, 2013. 125

[181] J. Liu, J. Liu, and X. Qiu, "A proxy blind signature scheme and an off-line electronic cash scheme," *Wuhan University Journal of Natural Sciences*, vol. 18, no. 2, pp. 117–125, 2013. 125

[182] N. Kiran and G. Kumar, "Reliable OSPM schema for secure transaction using mobile agent in micropayment system," in *Fourth International Conference on Computing, Communications and Networking Technologies*, July 2013, pp. 1–6. 125

[183] B. Yahid, M. Nobakht, and A. Shahbahrami, "Providing security for e-wallet using e-cheque," in *7th Intenational Conference on e-Commerce in Developing Countries: With Focus on e-Security*, April 2013, pp. 1–14. 125

[184] J. Wen-Shenq, "An efficient and practical fair buyer-anonymity exchange scheme using bilinear pairings," in *Proceedings of the 2013 Eighth Asia Joint Conference on Information Security*, July 2013, pp. 19–26. 125, 134, 136, 137

[185] V. Daza, R. Di Pietro, F. Lombardi, and M. Signorini, "FORCE - Fully Off-line secuRe CrEdits for Mobile Micro Payments," in *11th Intl. Conf. on Security and Cryptography*, SCITEPRESS, Ed., 2014. 125

[186] E. S. Canlar, M. Conti, B. Crispo, and R. Di Pietro, "Windows mobile LiveSD forensics," *J. Netw. Comput. Appl.*, vol. 36, no. 2, pp. 677–684, Mar 2013. 129

[187] T. Nishide and K. Sakurai, "Security of offline anonymous electronic cash systems against insider attacks by untrusted authorities revisited," in *Proceedings of the 2011 Third International Conference on Intelligent Networking and Collaborative Systems*, ser. INCOS '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 656–661. 131, 134

[188] S. Miyazaki and K. Sakurai, "Classification of Chaum-Fiat-Naor paradigm based anonymous electronic cash systems according to vulnerability against insider-attacks from untrusted authorities," in Cryptographic Techniques and E-Commerce, pp. 262–271, 1999. 131, 132, 137

[189] Y. Hanatani, Y. Komano, K. Ohta, and N. Kunihiro, "Provably secure

electronic cash based on blind multisignature schemes," in *Proceedings of the 10th International Conference on Financial Cryptography and Data Security*, ser. FC'06, 2006, pp. 236–250. 132

[190] ——, "Provably secure untraceable electronic cash against insider attacks," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. E90-A, no. 5, pp. 980–991, May 2007. 132, 137

[191] S. Canard and A. Gouget, "Anonymity in transferable e-cash," in *Applied Cryptography and Network Security*, ser. Lecture Notes in Computer Science, S. Bellovin, R. Gennaro, A. Keromytis, and M. Yung, Eds. Springer Berlin Heidelberg, 2008, vol. 5037, pp. 207–223. 132, 137

[192] ——, "Multiple denominations in e-cash with compact transaction data," in *Financial Cryptography and Data Security*, ser. Lecture Notes in Computer Science, R. Sion, Ed. Springer Berlin Heidelberg, 2010, vol. 6052, pp. 82–97. 132, 137

[193] O. Blazy, S. Canard, G. Fuchsbauer, A. Gouget, H. Sibert, and J. Traoré, "Achieving optimal anonymity in transferable e-cash with a judge," in *Progress in Cryptology*, ser. Lecture Notes in Computer Science, A. Nitaj and D. Pointcheval, Eds. Springer Berlin Heidelberg, 2011, vol. 6737, pp. 206–223. 132, 136, 137

[194] X. Hu and S. Huang, "A provably secure blind signature scheme," in *Proceedings of the 4th International Conference on Theory and Applications of Models of Computation*, ser. TAMC'07, 2007, pp. 171–180. 132

[195] W. Chen, G. Hancke, K. Mayes, Y. Lien, and J.-H. Chiu, "Using 3G network components to enable NFC mobile transactions and authentication," in *IEEE International Conference on Progress in Informatics and Computing*, vol. 1, Dec 2010, pp. 441 –448. 134

[196] S. Golovashych, "The technology of identification and authentication of financial transactions. from smart cards to NFC-terminals," in *Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications.*, Sep 2005, pp. 407–412. 134

[197] M. G. Vasco, S. Heidarvand, and J. Villar, "Anonymous subscription schemes: A flexible construction for on-line services access," in *Proceedings of the 2010 International Conference on Security and Cryptography*, July 2010, pp. 1–12. 134, 136, 137

[198] K. S. Kadambi, J. Li, and A. H. Karp, "Near-field communication-based secure mobile payment service," in *ICEC '09: Proceedings of*

the 11th International Conference on Electronic Commerce. ACM Request Permissions, Aug. 2009. 134

[199] V. C. Sekhar and S. Mrudula, "A complete secure customer centric anonymous payment in a digital ecosystem," in *2012 International Conference on Computing, Electronics and Electrical Technologies (ICCEET)*, 2012, pp. 1049–1054. 134

[200] S. Dominikus and M. Aigner, "mCoupons: An application for near field communication (NFC)," in *21st International Conference on Advanced Information Networking and Applications Workshops*, 2007, pp. 421–428. 134

[201] V. Patil and R. K. Shyamasundar, "An efficient, secure and delegable micro-payment system," in *International Conference on e-Technology, e-Commerce and e-Service*, 2004, pp. 394–404. 134

[202] M. Aigner, S. Dominikus, and M. Feldhofer, "A system of secure virtual coupons using NFC technology," in *Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops*. IEEE, 2007, pp. 362–366. 134

[203] M. A. Salama, N. El-Bendary, and A. E. Hassanien, "Towards secure mobile agent based e-cash system," in *1st Intl. Workshop on Security and Privacy Preserving in e-Societies*. New York, NY, USA: ACM, 2011, pp. 1–6. 134, 136, 137

[204] S. Brands, "Off-line electronic cash based on secret-key certificates," in *LATIN '95: Theoretical Informatics*, ser. Lecture Notes in Computer Science, R. Baeza-Yates, E. Goles, and P. Poblete, Eds. Springer Berlin Heidelberg, 1995, vol. 911, pp. 131–166. 135, 137, 144

[205] M. Franklin and M. Yung, "Secure and efficient off-line digital money," in *In Proceedings of ICALP'93, (LNCS 700*. Springer-Verlag, 1993, pp. 265–276. 137

[206] L. A. Schoenmakers, "An efficient electronic payment system withstanding parallel attacks," Centre for Mathematics and Computer Science, Amsterdam, The Netherlands, The Netherlands, Tech. Rep., 1995. 137

[207] B. Ernie, G. Peter, and K. David, "Trustee-based tracing extensions to anonymous cash and the making of anonymous change," in *In Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, 1995, pp. 457–466. 137, 144

[208] M. Franklin and M. Yung, "Towards provably secure efficient electronic cash (extended abstract)," IBM Research, Tech. Rep., 1992. 137

[209] M. Jakobsson and M. Yung, "Revokable and versatile electronic money (extended abstract)," in *Proceedings of the 3rd ACM Conference on Computer and Communications Security*, ser. CCS '96, 1996, pp. 76–87. 137

[210] J. C. Pailles, "New protocols for electronic money," in *Advances in Cryptology*, vol. 718, 1992, pp. 263–274. 137

[211] N. Ferguson, "Single term off-line coins," University of Bergen, Tech. Rep., 1994. 137

[212] M. H. Au, W. Susilo, and Y. Mu, "Electronic cash with anonymous user suspension," in *Proceedings of the 16th Australasian Conference on Information Security and Privacy*, ser. ACISP'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 172–188. 137

[213] S. Brands, "Untraceable off-line cash in wallet with observers," in *Proceedings of the 13th Annual International Cryptology Conference on Advances in Cryptology*, ser. CRYPTO '93, 1994, pp. 302–318. 137, 144

[214] S. D'Amiano and G. Di Crescenzo, "Methodology for digital money based on general cryptographic tools," in *Advances in Cryptology*, ser. Lecture Notes in Computer Science, A. De Santis, Ed. Springer Berlin Heidelberg, 1995, vol. 950, pp. 156–170. 137

[215] C. Jan, M. Ueli, and S. Markus, "Digital payment systems with passive anonymity-revoking trustees," in *Computer Security*. Springer Verlag, 1996, pp. 33–43. 137

[216] Y. Frankel, Y. Tsiounis, and M. Yung, "Indirect discourse proof: Achieving efficient fair off-line e-cash," in *Proceedings of the International Conference on the Theory and Applications of Cryptology and Information Security: Advances in Cryptology*, ser. ASIACRYPT '96, K. Kim and T. Matsumoto, Eds. London, UK, UK: Springer-Verlag, 1996, pp. 286–300. 137

[217] K. Q. Nguyen, Y. Mu, and V. Varadharajan, "A new digital cash scheme based on blind nyberg-rueppel digital signature," in *Proceedings of the First International Workshop on Information Security*, ser. ISW '97. London, UK, UK: Springer-Verlag, 1998, pp. 313–320. 137

[218] A. de Solages and J. TraorÃĺ, "An efficient fair off-line electronic cash system with extensions to checks and wallets with observers," in *Financial Cryptography*, ser. Lecture Notes in Computer Science, R. Hirchfeld, Ed. Springer Berlin Heidelberg, 1998, vol. 1465, pp. 275–295. 137, 144

[219] S. Canard, A. Gouget, and J. Traoré, "Improvement of efficiency in (unconditional) anonymous transferable e-cash," in *Financial Cryptography and Data Security*, ser. Lecture Notes in Computer Science, G. Tsudik, Ed. Springer Berlin Heidelberg, 2008, vol. 5143, pp. 202–214. 137

[220] T. Okamoto and K. Ohta, "Universal electronic cash," in *Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*, ser. CRYPTO '91, London, UK, UK, 1992, pp. 324–337. 137

[221] T. Eng and T. Okamoto, "Single-term divisible electronic coins," in *Advances in Cryptology âĂŤ EUROCRYPT'94*, ser. Lecture Notes in Computer Science, A. De Santis, Ed. Springer Berlin Heidelberg, 1995, vol. 950, pp. 306–319. 137, 144

[222] T. Okamoto, "An efficient divisible electronic cash scheme," in *Proceedings of the 15th Annual International Cryptology Conference on Advances in Cryptology*, ser. CRYPTO '95. London, UK, UK: Springer-Verlag, 1995, pp. 438–451. 137

[223] H. Peterson and G. Poupard, "Efficient scalable fair cash with off-line extortion prevention," in *Proceedings of the First International Conference on Information and Communication Security*, ser. ICICS '97, 1997, pp. 463–477. 137, 144

[224] E. Fujisaki and T. Okamoto, "Practical escrow cash system," in *Proceedings of the International Workshop on Security Protocols*, 1997, pp. 33–48. 137, 144

[225] Y. Yacobi, "Efficient electronic money (extended abstract)," in *Proceedings of the 4th International Conference on the Theory and Applications of Cryptology: Advances in Cryptology*, ser. ASIACRYPT '94, 1995, pp. 153–163. 137, 144

[226] S. Miyazaki and K. Sakurai, "A more efficient untraceable e-cash system with partially blind signatures based on the discrete logarithm problem," in *Proceedings of the Second International Conference on Financial Cryptography*, ser. FC '98, 1998, pp. 296–308. 137, 144

[227] D. Xiaoling, O. Ayoade, and J. Grundy, "Off-line micro-payment protocol for multiple vendors in mobile commerce," in *Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies.*, 2006, pp. 197–202. 137

[228] C. Popescu and H. Oros, "An off-line electronic cash system based on bilinear pairings," in *14th International Workshop on Systems, Signals*

and Image Processing, 2007 and 6th EURASIP Conference focused on Speech and Image Processing, Multimedia Communications and Services., 2007, pp. 438–440. 137

[229] Z. Xuanwu, "Threshold cryptosystem based fair off-line e-cash," in *Second International Symposium on Intelligent Information Technology Application*, vol. 3, 2008, pp. 692–696. 137

[230] S. Abhinav, K. Amlan, S. Shamik, and M. Arun, "Credit card fraud detection using hidden markov model," *IEEE Transactions on Dependable and Secure Computing*, vol. 5, no. 1, pp. 37–48, 2008. 137

[231] W. Changji and L. Rongbo, "An ID-based transferable off-line e-cash system with revokable anonymity," in *International Symposium on Electronic Commerce and Security*, 2008, pp. 758–762. 137

[232] Z.-g. Wang and Z.-k. Wan, "A secure off-line electronic cash scheme based on ECDLP," in *First International Workshop on Education Technology and Computer Science*, vol. 2, 2009, pp. 30–33. 137

[233] K. C. Brijesh and V. Shekhar, "Secure pay while on move toll collection using VANET," *Computer Standards & Interfaces*, vol. 36, no. 2, pp. 403–411, 2014. 137

[234] G. Madlmayr, J. Langer, C. Kantner, and J. Scharinger, "NFC devices: Security and privacy," in *Third International Conference on Availability, Reliability and Security.*, 2008, pp. 642–647. 138

[235] W. Frisby, B. Moench, B. Recht, and T. Ristenpart, "Security analysis of smartphone point-of-sale systems," in *Proceedings of the 6th USENIX conference on Offensive Technologies*, ser. WOOT'12. Berkeley, CA, USA: USENIX Association, 2012, pp. 123–134. 138

[236] Y. Chen, J.-S. Chou, H.-M. Sun, and M.-H. Cho, "A novel electronic cash system with trustee-based anonymity revocation from pairing," *Electron. Commer. Rec. Appl.*, vol. 10, no. 6, pp. 673–682, nov 2011. 138

[237] H. Ho, S. Fong, and Z. Yan, "User acceptance testing of mobile payment in various scenarios," in *e-Business Engineering, 2008. ICEBE '08. IEEE International Conference on*, 2008, pp. 341–348. 138, 139

[238] C. Kim, S. Choe, C. Choi, and Y. Park, "A systematic approach to new mobile service creation," *Expert Syst. Appl.*, vol. 35, no. 3, pp. 762–771, Oct 2008. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2007.07.044 138, 139

[239] A. Sarajlic and D. Omerasevic, "Access channels in m-commerce services," in *29th International Conference on Information Technology*

Interfaces., 2007, pp. 507–512. 139

[240] D. McKitterick and J. Dowling, "State of the art review of mobile payment technology," 2003. 139

[241] D. McKitterick, *A Web Services Framework for Mobile Payment Services.* Trinity College, 2003. [Online]. Available: http://books.google.it/books?id=rPaJMwEACAAJ 140

[242] A. Shivani, K. Mitesh, M. Bernard, and U. Nirav, "Security issues in mobile payment systems," in *Towards Next Generation E-government*, J. Bhattacharya, Ed., 2007. 140

[243] N. Seema, L. Chang-Tien, and L. Liang, "Analysis of payment transaction security in mobile commerce," in *Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration*, 2004, pp. 475–480. 140

[244] T. D.llori, "Emv chip technology @ONLINE," 2013. 142

[245] Trustwave, "2013 global security report," Trustwave, Technical Report, 2013. [Online]. Available: http://www2.trustwave.com/rs/trustwave/images/2013-Global-Security-Report.pdf 144

[246] R. Battistoni, A. D. Biagio, R. Di Pietro, M. Formica, and L. V. Mancini, "A live digital forensic system for windows networks," in *Proceedings of The Ifip Tc 11 23rd International Information Security Conference*, S. Jajodia, P. Samarati, and S. Cimato, Eds. Springer US, 2008, vol. 278, pp. 653–667. 145

[247] V. Patil and R. Shyamasundar, "An efficient, secure and delegable micro-payment system," in *IEEE International Conference on e-Technology, e-Commerce and e-Service*, March 2004, pp. 394–404.

[248] W. Michael, C. Ismail, M. Malgorzata, P. Aneto, P. W. Patrick, and H. Gökhan, "In-card access control and monotonic counters for offline payment processing system," Australian Patent PCT/US2013/028 466, 2013.

[249] F. Jiang, M. Lisowiec, M. Springer, A. Okonkwo, and P. Leung, "Presence-of-card code for offline payment processing system," U.S. Patent US9 020 858 B2, 2015, uS Patent 9,020,858. [Online]. Available: http://www.google.com/patents/US9020858

[250] F. Jiang, M. Springer, M. Lisowiec, G. Bakir, P. Leung, and A. Okonkwo, "Transaction signature for offline payment processing system," U.S. Patent US8 959 034 B2, 2015, uS Patent 8,959,034. [Online]. Available: http://www.google.com/patents/US8959034

[251] N. Richard H.H., M. Mirkazemi-Moud, J. Barrowman, and C. Schulz,

"Point of sale terminal having enhanced security," U.S. Patent US20 130 106 606 A1, 2013, uS Patent App. 13/717,957. [Online]. Available: http://www.google.com.ar/patents/US20130106606