Security bootstrapping for Internet of Things

Sameh Khalfaoui

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2022IPPAT023

Thèse de doctorat

Secure Bootstrapping for Internet of Things

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Télécom Paris

École doctorale n°626 Ecole doctorale de l'Institut Polytechnique de Paris (ED IP Paris)
Spécialité de doctorat : Informatique

Thèse présentée et soutenue à Palaiseau, le 20 juin 2022, par

SAMEH KHALFAOUI

Maryline LAURENT Professeure, Télécom SudParis	Présidente, Examinatrice
Samia SAAD-BOUZEFRANE Professeure, Conservatoire National des Arts et Métiers (CNAM)	Rapporteure
Myungchul KIM Professeur, Korea Advanced Institute of Science and Technology (KAIST)	Rapporteur
Youssef LAAROUCHI Chef de projet, EDF R&D	Examinateur
Marc-Oliver PAHL Directeur de recherche, IMT Atlantique Campus Rennes	Examinateur
Pascal URIEN Professeur, Télécom Paris	Directeur de thèse
Jean LENEUTRE Maître de conférences, Télécom Paris	Co-Directeur de thèse
Jingxuan MA Ingénieure de recherche, EDF R&D	Encadrante

Abstract

The demand for Internet of Things (IoT) services is increasing exponentially, and consequently a large number of devices are being deployed. These connected objects are, nowadays, considered as an essential part of the business processes of numerous industry sectors. The easiness of adoption of these devices and their compact nature make them a cost-efficient tool that can perfectly adapt to numerous use-cases. However, these devices can represent a serious threat to the security of the deployment network and a potential entry-point when exploited by the adversaries. Thus, there is an imminent need to perform a secure association approach of the IoT objects before being rendered operational on the network of the user. This procedure is referred to as secure bootstrapping and it primarily guarantees the confidentiality and the integrity of the data exchanges between the user and the devices. Secondly, this process provides an assurance on the identity and the origin of these objects.

Due to scalability limitations, the first phase of the bootstrapping process cannot be efficiently conducted using pre-shared security knowledge such as digital certificates. This step is referred to as secure device pairing and it ensures the establishment of a secure communication channel between the use and the object. The pairing phase uses an ad-hoc symmetric key agreement protocol that is suitable to the resource-constrained nature of these devices. The use of auxiliary channels has been proposed as a way to authenticate the key exchange but they require a relatively long time and an extensive user involvement to transfer the authentication bits. However, the context-based schemes use the ambient environment to extract a common secret without an extensive user intervention under the requirement of having a secure perimeter during the extraction phase, which is considered a strong security assumption.

The second phase of the bootstrapping process is referred to as secure device enrollment and it aims at avoiding the associating of a malicious IoT object by authenticating its identity. The use of hardware security elements, such as the Physical Unclonable Function (PUF), has been introduced as a promising solution that is suitable for the resource-constraint nature of these devices. A growing number of PUF architectures has been demonstrated mathematically clonable through Machine Learning (ML) modeling techniques. The use of ML PUF models has been recently proposed to authenticate the IoT objects. This procedure facilitates the scalability of the authentication process by reducing the storage space required for each device. Nonetheless, the leakage scenario of the PUF model to an adversary due to an insider threat within the organization is not supported by the existing solutions. Hence, the security of these PUF model-based enrollment proposals can be compromised.

In this thesis, we study the secure bootstrapping process of resource-constrained devices and we introduce two security schemes:

- A hybrid ad-hoc pairing protocol, called COOB, that efficiently combines a state-of-the-art fast context-based scheme with the use of an auxiliary channel. This protocol exploits a nonce exponentiation of the Diffie-Hellman public keys to achieve the temporary secrecy goal needed for the key agreement. Our method provides

security even against an attacker that can violate the safe zone requirement, which is not supported by the existing contextual schemes. This security improvement has been formally validated in the symbolic model using the TAMARIN prover.

- An enrollment solution that exploits a ML PUF model in the authentication process, called Water-PUF. Our enrollment scheme is based on a specifically designed black-box watermarking technique for PUF models with a binary output response. This procedure prevents an adversary from relying on the watermarked model in question or another derivative model to bypass the authentication. Therefore, any leakage of the watermarked PUF model that is used for the enrollment does not affect the correctness of the protocol. The Water-PUF design is validated by a number of simulations against numerous watermark suppression attacks to assess the robustness of our proposal.

Résumé

La demande de services qui se basent sur l'Internet des objets (IoT) augmente de manière exponentielle, ce qui entraîne le déploiement d'un grand nombre de dispositifs. Ces objets connectés sont, de nos jours, considérés comme une partie essentielle des processus industriels de nombreux secteurs d'activités. La facilité d'adoption de ces dispositifs et leur nature compacte en font un outil rentable qui s'adapte parfaitement à de nombreux cas d'utilisation. Cependant, ces dispositifs peuvent représenter une menace pour la sécurité du réseau de déploiement et un point d'entrée potentiel pour des adversaires. Il existe donc un besoin imminent de réaliser une approche d'association sécurisée des objets connectés avant qu'ils ne soient rendus opérationnels sur le réseau de l'utilisateur. Cette procédure, appelée "amorçage de la sécurité", garantit en premier lieu la confidentialité et l'intégrité des échanges de données entre l'utilisateur et les dispositifs. Ensuite, ce processus fournit une assurance sur l'identité et l'origine de ces objets.

En raison des limites d'évolutivité, la première phase du processus d'amorçage ne peut pas être menée efficacement en utilisant des connaissances de sécurité pré-partagées telles que des certificats numériques. Cette étape d'appairage assure l'établissement d'un canal de communication sécurisé entre l'utilisation et l'objet. La phase d'appairage utilise un protocole d'accord de clé symétrique qui est adapté à la nature de ces dispositifs à ressources limitées. L'utilisation de canaux auxiliaires a été proposée comme moyen d'authentifier l'échange de clés, mais elle nécessite un temps relativement long et une participation importante de l'utilisateur pour transférer les bits d'authentification. Cependant, les systèmes basés sur le contexte utilisent l'environnement ambiant pour extraire un secret commun sans intervention importante de l'utilisateur, à condition d'avoir un périmètre sécurisé pendant la phase d'extraction, ce qui est considéré comme une hypothèse de sécurité forte.

La deuxième phase du processus d'amorçage est appelée "enrôlement sécurisée" et vise à éviter l'association d'un objet IoT malveillant en authentifiant son identité et son origine. L'utilisation d'éléments de sécurité matériels, tels que les fonctions physiques non clonables (PUF), a été présentée comme une solution prometteuse adaptée à la nature limitée des ressources de ces dispositifs. Un nombre croissant d'architectures PUF ont été démontrées mathématiquement clonables grâce à des techniques de modélisation par apprentissage automatique. L'utilisation de modèles de PUF a été récemment proposée pour authentifier les objets IoT. Cette procédure facilite l'évolutivité du processus d'authentification en réduisant l'espace de stockage requis pour chaque dispositif. Néanmoins, le scénario de fuite du modèle PUF vers un adversaire en raison d'une menace interne au sein de l'organisation n'est pas pris en charge par les solutions existantes. Par conséquent, la sécurité de ces propositions d'inscription basées sur le modèle PUF peut être compromise.

Dans cette thèse, nous étudions le processus d'amorçage de la sécurité des dispositifs à ressources limitées et nous introduisons deux protocoles :

- Un protocole hybride d'appairage, appelé COOB, qui combine d'une manière efficace un schéma d'appairage contextuel avec l'utilisation d'un canal auxiliaire. Ce protocole exploite une technique d'exponentiation spécifiques des clés publiques Diffie-

Hellman en utilisant des nonces pour atteindre l'objectif de secret temporaire nécessaire à l'accord de clé. Notre méthode assure la sécurité même contre un attaquant qui peut contrôler la zone de sécurité (un environnement hostile), ce qui n'est pas pris en charge par les schémas contextuels existants. Cette amélioration de la sécurité a été formellement validée dans le modèle symbolique en utilisant l'outil de vérification formelle TAMARIN.

- Une solution d'enrôlement qui exploite un modèle de PUF dans le processus d'authentification, appelé Water-PUF. Notre protocole est basé sur une technique de tatouage numérique spécialement conçue pour les modèles PUF. Cette procédure empêche un adversaire de s'appuyer sur le modèle tatoué ou sur un autre modèle dérivé pour contourner l'authentification. Par conséquent, toute fuite du modèle PUF filigrané utilisé pour l'enrôlement n'affecte pas l'exactitude du protocole. La conception du Water-PUF est validée par un certain nombre de simulations contre de nombreuses attaques de suppression de tatouage numérique afin d'évaluer la robustesse de notre proposition.

Acknowledgments

First and foremost I am extremely grateful to my team of supervisors for their invaluable advice, continuous support, and patience during my PhD study. Their immense knowledge and plentiful experience have encouraged me in all the time of my academic research and daily life. I would like to thank all the members of the cybersecurity research group in EDF R&D. It is their help and support that have made my study and my life within the group a memorable time. Finally, I would like to express my gratitude to my family for having my back since day one, my fiancee for bringing the best in myself and my friends for being there for me when I needed them. Without their tremendous understanding and encouragement in the past few years, it would be impossible for me to complete my study.

You can never cross the ocean unless you have the courage to lose sight of the shore.

Christopher Columbus

This work was conducted within the SEIDO lab that is the joint research laboratory for Security and Internet of Things between EDF R&D and Télécom Paris. The thesis was supervised by:

- Jean LENEUTRE, Assistant-Professor, Télécom Paris
- Pascal URIEN, Professor, Télécom Paris
- Arthur VILLARD, Research Engineer, EDF R&D
- Ivan GAZEAU, Research Engineer, EDF R&D
- Jingxuan MA, Research Engineer, EDF R&D

Contents

List of Figures

List of Tables

1 | Introduction

Contents

The Internet of Things refers to the network of physical devices that are connected to the Internet in order to collect and share data. These devices have recently become essential in our everyday life: connected vehicles, home automation facilities, smart factory sensors, and fitness trackers. Altogether, they create a massive ecosystem of billions of deployed devices that has been valued at approximately 100 billion USD in 2017 with a growth estimation that is expected to reach 1.6 trillion USD by 2025 [136]. This technology has recently become more industry-specific with a wide adoption in sectors such as agriculture [129], healthcare [42], retail [22], manufacturing [148], automotive [156] and energy [85]. Consequently, according to [121], the Industrial Internet of Things (IIoT) market is expected to grow at a compound annual growth rate of 16.7% from 2021 to 2027 in order to reach a total valuation of 263.4 billion USD by 2027. The increasing growth rates reflect the positive impact of the IoT technology on these sectors.

The increasing popularity of the IoT products pressures the manufacturers to opt for a rush-to-market behavior in order to comply with the needs of their clients. Thus, they tend to overlook the importance of ensuring the security of these resource-constraint devices which might create a potential attack vector once they are deployed. In the first half of 2021, the Kaspersky honeypots have detected more than 1.5 billion attacks against IoT devices that aim at stealing data, mining cryptocurrency or compromising these systems to create botnets.

As a consequence, the National Institute of Standards and Technology (NIST) has recently introduced new regulations, NISTIR 8259A [64], for the United States (US) IoT market regarding the security of the introduced devices and the collected data. On the other hand, the European Telecommunications Standards Institute (ETSI) has released similar cybersecurity guidelines, in the ETSI EN 303 645 report [62], for the IoT consumer market in Europe and in the United Kingdom (UK). For these reasons, the manufacturers need a more comprehensive and easy-to-adopt security solution in order to keep pace with these regulations. Therefore, the appliance of a secure association procedure that is suitable for the IoT context is crucial to communicate securely with the object in question. Thus, we eliminate the risk related to associating a malicious object into the network of the user. This secure association process ensures that the communicating IoT nodes are, indeed, the ones intended to.

1.1 Background on the Internet of Things

The increasing impact of the IoT technology on our lives is due to its capability to adapt to the different application scenarios imposed by the end-users. The high adoption rate and the lack of security considerations with respect to the nature of these devices can create the perfect attack vector to gain access to the network of the user. A prominent example of these threats is the data breach, reported by the cybersecurity firm Darktrace [141], through the exploitation of a vulnerability in an Internet-connected thermostat of an aquarium in a casino. This vulnerable IoT object has been used as an entry point to the network of the casino which gave the attacker access to the high-roller database of gamblers which he has pulled back across the network and out through the thermostat. Thus, in the upcoming parts, we aim at determining the security challenges related to the usage of the IoT technology in the different application scenarios.

1.1.1 Applications

To better highlight the importance of ensuring the security of these devices, we start by describing their most relevant applications along with the degree of dependency of the major sectors on the IoT services. These applications can be classified into four major categories:

I. **Consumer IoT:** The consumer IoT solutions focus on the personal use of these services such as wearables, smart home functionalities and personal monitoring systems. A suitable example is the personal home assistant, such as Google Home or Amazon Echo, that manages the different communicating IoT devices placed around the home. Another common example is the self-monitoring wearables such as the smart watches and the smart bands that provide ongoing care and monitoring by collecting data and by displaying relevant information about the user's well-being.

II. **Commercial IoT:** These devices are deployed in businesses in order to enhance the client experience through a more efficient monitoring for sectors such as the hospitality, healthcare and retail industries. For instance, the cashier-less grocery store Amazon Go [196] is a prime example that displays the use of commercial IoT devices to reduce queuing time at cashiers. This process is conducted through the use of cameras and sensors that detect the selected products and automate the billing operations.

III. **Industrial IoT:** The industrial IoT solutions aim at improving the efficiency and the productivity of the existing automated industrial systems within a large scale factory or a manufacturing plant. This category differs from the commercial IoT class because the integration process with the existing legacy infrastructure is generally more complex to perform. The massive deployment of these devices provides an important amount of data that is analyzed through big-data and machine learning techniques to derive the appropriate business decisions. The enormous amount of data that is collected by every sensors inside the factory can serve as a source of information to create a digital twin that corresponds to a virtual representation of the different manufacturing processes. Thus, it can enhance the entire industrial ecosystem by monitoring the performance of the production lines and the prediction of future maintenance operations. According to [72], the digital twin market was valued at 2.6 billion USD in 2019 and is estimated to reach 73.2 billion dollars by 2030.

IV. **Military IoT:** Similar to the application of the IoT devices in the civilian use-cases, this technology also seems to appeal to the military field since it offers the

opportunity to gather battlefield awareness in advance through the drones and the connected cameras. Thus, it is easier to perform the autonomous target recognition. Furthermore, the attached sensors on the deployed troops can help track and alert about their health conditions which helps to rapidly provide the necessary medical intervention. In addition, the gathered field data from earlier missions constitutes a valuable tool to create a virtual training simulator in association with the augmented reality techniques. The Lockheed Martin defense corporation is currently developing military tools that exploit the IoT technology in order to create a revolutionary war-fighting network that is called Command, Control Battle Management and Communications System (C2BMC) [50].

1.1.2 Security Challenges

The IoT devices within the target network of the user can be an ideal entry point for the attackers. This is mainly due to the lack of security considerations by the manufacturer and the negligence of the security best practices by the end-user. Thus, we conclude the existence of two evaluation metrics that are crucial to the resiliency of the IoT ecosystem: *Security* and *Usability*. The former term describes the existence of the appropriate security protocols on-board of the IoT device. The latter term refers to the ease of adoption of these security practices by the user. Clearly, the different categories of applications, described in Subsection 1.1.1, require variant degrees of security and usability. For instance, the consumer IoT category tends to favor a faster and more intuitive security protocol, which provides a higher usability aspect, at the expense of a more strict and well secured procedure. This is not obviously the case of the industrial and the military IoT applications due to the utmost importance of providing a secure service while deprioritizing the usability aspect. This is explained by the agent capability to perform correctly the security procedures which is not always the case for a regular user.

The existing IoT devices suffer from a number of significant security challenges:

- **Weak password protection:** The use of guessable or default passwords to communicate with the IoT object is considered as the most common malpractice in security. The Mirai malware [15] is a good example of the severity of this bad practice. The malware relies upon the use of 61 common default credentials from multiple remote access protocols to the vulnerable objects. The created botnet has approximately 400 000 infected devices that were used to launch, in 2016, a massive distributed denial-of-service attack on several Internet services. This attack has targeted the cloud service provider Amazon Web Services (AWS) and has consequently affected its clients, including Airbnb, Twitter, Github and Netflix [6].

- **Weak update mechanism:** The existence of an update protocol is essential to guarantee the continuous secure deployment of the IoT device. The object may be secure at the time of purchase but we need to make sure that we are able to securely patch the discovered vulnerabilities in the future. For instance, the Satori malware [205] exploits these issues by targeting the known and unpatched vulnerabilities in the D-Link routers to create a botnet. This attack has infected more than 500 000 routers around the globe. The impact of this malware has lead D-Link to issue a recommendation to replace the vulnerable devices which comes with a significant cost for the affected industries.

- **Insecure communication interfaces:** Numerous IoT device manufacturers still neglect the use of encryption or authentication techniques to guarantee the confidentiality and the integrity of the transferred data. Thus, it is fairly simple for an

adversary to control the communication between the IoT device and the user. Consequently, he can force specific business decisions that result from the received poisoned data. The NIST and the ENISA have published a series of detailed guidelines and recommendations for the IoT secure management in the US, the European Union and the UK. As a result, the manufacturers have started to use digital certificates to perform the encryption and the authentication of the devices.

The Cayla doll incident [36] in 2017 is a good illustration of the importance of having a secure encryption and authentication mechanism. This toy has the ability to interact with the children through the use of an Internet connection. Nonetheless, it has been discovered vulnerable to eavesdropping attacks due to the lack of encryption on the communication channel. Furthermore, the access to the doll through Bluetooth was completely insecure. Thus, it was banned in Germany because it endangers the safety and the privacy of the children.

- **Insufficient IoT device management:** The lack of IoT device management within the organization can result in the association of unauthorized devices to the network. A study has been conducted in 2020 on 5 million unmanaged connected devices that have been deployed in sectors such as healthcare, retail and manufacturing. Thus, they are expected to provide a relatively high level of security to protect the data of the clients. Unfortunately, 15% of the devices were unauthorized and 95% of healthcare networks integrated Personal Smart Assistants, such as Amazon Echo, alongside hospital surveillance equipment. Furthermore, up to 19% of these IoT objects were running not updated operating systems. This study has shed light on a number of defective and unregulated devices that were still active in the network.

The commonly adopted IoT standards such as the Constrained Application Protocol (CoAP) [180], the Message Queuing Telemetry Transport (MQTT) [87] and the Lightweight Machine to Machine (LwM2M) [117] tend to rely on the digital certificates to guarantee the authenticity and the confidentiality of the data. Other standards consider the use of a decentralized symmetric key exchange process that establishes a secure communication channel between the IoT device and the user without the need for pre-established security information. These standards avoid the management challenges of a large fleet of IoT object certificates throughout their life cycle, especially in massive deployment scenarios. Furthermore, the majority of these initiatives tends to focus on the key establishment procedure while neglecting the entity authentication of the associated IoT object. This strategy might expose the network to the risk of associating a malicious object that is under the control of the adversary.

This thesis aims at proposing a secure association procedure of resource-constrained IoT devices, referred to as secure bootstrapping, that provides the security of the communication channel and the authenticity of the object in question. The former objective is ensured through the use of a Secure Device Pairing (SDP) protocol. The latter objective is ensured through the use of a Secure Device Enrollment Secure Device Enrollment (SDE) protocol.

1.2 Contributions

Hereafter, we highlight the main contributions of the thesis.

The first part of the thesis tackles the secure device pairing of IoT devices without having pre-shared security knowledge. We have noticed that the usability assessment has become the main requirement due to the importance of providing the most user-friendly

IoT services. Thus, the complete security analysis has been replaced by a sketch of a proof to partially validate the robustness of the proposal. The few existing formal or computational security verifications on the SDP schemes have been conducted based on the assessment of a wide variety of uniquely defined security properties. Therefore, the security comparison between these protocols is not feasible and there is a lack of a unified security analysis framework to assess these pairing techniques. In this first part, we survey a selection of secure device pairing proposals that have been formally or computationally verified. We present a systematic description of the protocol assumptions, the adopted verification model and an assessment of the verification results. In addition, we normalize the used taxonomy in order to enhance the understanding of these security validations. We also discuss the consequences of a recent adversary model that provide the attacker with the ability to partially compromise one of the participating devices.

Afterwards, we introduce a hybrid pairing scheme, called COOB [103], that performs a key agreement procedure by efficiently combining two state-of-the-art techniques. Our method provides security against a sophisticated adversary model that is not supported by the existing state-of-the-art schemes. This security improvement has been formally validated in the symbolic model using the TAMARIN verification tool [133]. This contribution has been published in the 16th EAI International Conference on Security and Privacy in Communication Networks (SecureComm 2020) [104] and the pairing protocol has been the subject of a European patent EP 3913951A1.

The second part of the thesis focuses on the secure device enrollment by verifying the origin and the identity of the IoT device. To efficiently authenticate these objects, we study the use of a hardware security technique, called Physical Unclonable Function, that is considered as a promising solution regarding the resource-constraint nature of these devices. The use of Machine Learning techniques to model the PUF behavior has been recently proposed to authenticate the IoT objects while reducing the storage space requirement for each device. Nonetheless, the use of a mathematically clonable PUF requires a careful design of the enrollment process. Furthermore, the secrecy of the used machine learning model of the PUF and the leakage scenario of this sensitive information to an adversary due to an insider threat within the organization has not been discussed. Therefore, we review the state-of-the-art model-based PUF enrollment protocols. We identity two architectures of enrollment protocols based on the participating entities and the building blocks that are relevant to the security of the authentication procedure. In addition, we describe an insider threat scenario that has been identified where the PUF model is leaked to the adversary.

We propose an enrollment protocol, called Water-PUF [102], that exploits a machine learning model of the PUF in the authentication process. Our enrollment scheme is based on a specifically designed black-box watermarking technique for PUF models with a binary output response. This watermarking technique aims at identifying the use of the ML model by detecting precise wrong prediction patterns. This procedure prevents an adversary from relying on the watermarked model in question or another derivative model to bypass the authentication. Therefore, any leakage of the watermarked PUF model that is used for the enrollment does not affect the correctness of the protocol. The Water-PUF design is validated by a number of simulations against numerous watermark suppression attacks to assess the robustness of our proposal. This contribution has been published in the 20th IEEE International Symposium on Network Computing and Applications (NCA 2021) [102] and the enrollment protocol has been the subject of a French patent FR 2107112.

1.3 Dissertation Outline

This dissertation is organized into 5 chapters. In Chapter 2, we introduce the required technical concepts about the secure bootstrapping of IoT devices, which are important for the comprehension of this thesis. We also study the state-of-the-art pairing techniques and we discuss their limitations. In addition, we study the existing secure device enrollment solutions while shedding light on the hardware security techniques.

In Chapter 3, We conduct a study on the formal and computational security assessments of a selection of SDP protocols. This survey has revealed a lack of a unified security analysis framework that affects the correctness of the evaluation results. Moreover, we introduce our hybrid secure device pairing protocol that combines two state-of-the-art techniques to enhance the resiliency of the pairing process.

In Chapter 4, we focus on the secure device enrollment protocols. We discuss the roles of the entities that contribute to the secure enrollment of the IoT device based on the use of a physical unclonable function. Afterwards, we study a selection of enrollment protocols that take advantage of these hardware techniques and the machine learning algorithms to conduct the authentication process. This survey has pointed to an insider threat scenario that targets one of the identified architectures. Then, we introduce our enrollment protocol that applies a specially constructed watermarking technique to identify the use of the leaked PUF model.

The conclusion of this manuscript as well as the perspectives of this thesis are presented in chapter 5.

2 | Secure Bootstrapping

Contents

In order to address the IoT security challenges, we focus on studying the secure association solutions for resource-constrained IoT devices that are also referred to as *Secure Bootstrapping*. According to the Internet Engineering Task Force (IETF) [178], the bootstrapping process is defined as *any process that takes place before a device can become operational*. In our context, we need to make sure that the user is communicating securely with a legitimate device. Therefore, we can divide the bootstrapping procedure into two distinct phases based on their security objectives: The confidentiality of the communications and the authenticity of the IoT device.

After the brief introduction of the bootstrapping procedure, we further describe, in Section 2.1, the desired security objectives and properties, the applied cryptographic techniques and a study of the existing bootstrapping initiatives that are commonly adopted by the manufacturers. In Section 2.2 and Section 2.3, we respectively conduct a survey on the state-of-the-art techniques that are applied to perform the two phases of the bootstrapping. In Section 2.4, we conclude this chapter by presenting a summary of the discussed bootstrapping approaches and by stating the main objectives of this thesis.

2.1 Introduction to Secure Bootstrapping

2.1.1 Definition and Security Objectives

The bootstrapping is a process that aims at associating an IoT device to the network of the user. To correctly perform this operation, the user needs to configure the object with

the necessary information to be fully operational. The confidentially and the integrity of the sensitive configuration data must be guaranteed to discard any Man-in-the-Middle (MitM) attack. Therefore, the bootstrapping procedure aims at creating an encrypted communication channel between the IoT object and the user's trusted device to guarantee the security of the exchanged configuration data. The key establishment procedure is defined in the NIST SP 800-152 [24] as follows:

Definition 1 (Key Establishment)**.** The process that results in the sharing of a key between two or more entities, either by transporting a key from one entity to another (key transport) or generating a key from information shared by the entities (key agreement).

According to the Definition 1, we conclude the existence of two categories of key establishment techniques:

- *Key Transport:* One entity creates a secret value and transfers it securely to the other device. This is the example of using public key cryptography and digital certificates on the IoT device.
- *Key Agreement:* A number of entities compute a shared secret key through the use of public information that is associated with each of the other entities. This is the example of using the Diffie-Hellman key exchange protocol that is described in Subsection 2.1.3.

In order to properly protect the network of the user, the bootstrapping process needs to verify the origin of the IoT device. This verification is a necessity to avoid the association of a malicious object that might become an entry point for attackers. Thus, along side the establishment of a secure communication channel with the device, the bootstrapping procedure has to ensure the entity authentication that is defined in the ISO/IEC 9798-1 standard as follows:

Definition 2 (Entity Authentication)**.** Entity authentication mechanisms allow the verification of an entity's claimed identity by another entity. The authenticity of the entity can be ascertained only for the instance of the authentication exchange.

The continuous guarantee of entity authenticity after the initial verification can be obtained by securing the key establishment procedure and the secret key storage. Therefore, any entity that has issued the encrypted communication using the established secret key is the one who has been authenticated.

These security guarantees of the bootstrapping process are achieved after the execution of a cryptographic protocol that is defined as a sequence of steps and message exchanges between multiple entities in order to achieve a specific security objective. According to the specifications of Lentra [114], the security level of this protocol is computed based on the computational effort that is required by a successful general attack on the cryptosystem. Thus, it is defined as follows:

Definition 3 (Protocol Security Level)**.** A cryptographic protocol offers a security level of λ if the computational instructions that are essential to find a solution to the cryptographic problem in the problem's domain is approximately equal to 2^{λ}.

2.1.2 State-of-the-art Threat Models

In the secure key establishment context, we identify two categories of threat models based on two security properties: *the demonstrative identification* and *the integrity of the device*. The first property, *the demonstrative identification*, was first introduced in the work of Balfanz el al. [21] and it guarantees the correctness of the key agreement initiation

process by making sure that the devices performing the operation are the ones intended to. Therefore, the user plays a crucial part in accomplishing this objective. The second property, *the device integrity*, represents the access privileges acquired by the attacker on the victim IoT device. Thus, it outlines the fact that one of the participants is partially under the control of the adversary, as detailed in the work of Do et al. [58]. This property covers both the hardware and the firmware integrity of the object in question. The adopted intruder models, in the key establishment protocols, assume that the two previously described security properties are achieved. This is explained by the intention to assess the robustness of the scheme by mainly focusing on the protocol exchanges or the employed cryptography. However, the work of Sethi et al. [177] has demonstrated the severity of violating these security requirements by proving the feasibility of an attack that aims at luring the user to perform the key establishment with a malicious device instead of a legitimate one. Unfortunately, this attack cannot be countered by the first phase of the bootstrapping process that is detailed in Chapter 3. However, it can be actively mitigated by the second phase through the use of entity authentication protocols that are described in Chapter 4. We conclude the existence of two categories of threat models: Non-invasive and invasive.

2.1.2.1 Non-Invasive Threat Models

In this part, the models assume that the demonstrative identification and the device integrity are achieved. This means that the user correctly initiates the key agreement between the legitimate participants and that those devices are not under the control of the attacker. To better understand the security analysis, in the upcoming section, we briefly describe the associated intruder models:

- **Dolev-Yao model [59]:** This model assumes that the adversary has the following capabilities:
  - The adversary has a perfect knowledge of the protocol steps.
  - The adversary has access to the public parameters of the protocol session.
  - The adversary can block, replay, delay any transmission by the honest agents.
  - The adversary can modify any transmission by the honest agents. This capability is also referred to as forgery.
  - The adversary cannot perform any computational attacks against the cryptographic primitives without the knowledge of the secret parameters.

- **AKISS model [41]:** In this model, the capabilities of the adversary are similar to the Dolev-Yao intruder powers. However, the work of Delaune et al. [54] has extended the model to provide the attacker with the capability to guess a low entropy secret.

- **Bellare-Rogaway [26, 27]:** In this model, each participant is modelled as an oracle that can be queried by the adversary that allows him to control which party initiates a new pairing session and which participant executes a specific step of the protocol. In addition, the attacker controls the communication between all the participants on the insecure channel and his powers are limited based on the choice of the auxiliary channel that is used in the protocol, as detailed in Subsection 2.2.1.2.

2.1.2.2 Invasive Threat Model

In comparison with the initial assumptions of the formal threat model, the demonstrative identification and the device integrity properties in the invasive threat model are not guaranteed.

The former violated property provides the adversary with the ability to lure the user to initiate the key agreement with the wrong device which has been demonstrated feasible and easy to accomplish on the Bluetooth Secure Simple Pairing protocol [177]. Therefore, the correctness of the discovery process of the key agreement between the intended devices is affected by the Human Factor Error (HFE) and by the lack of authentication due to the absence of pre-shared security knowledge.

As for the latter violated property, the adversary is able to gain access to the input/output interfaces of one of the participants which makes him able to intercept any message received by that device without the need of eavesdropping on the communication channel. Furthermore, he is able to send any message through that compromised devices, which simply makes it a external Input/Output interface for the attacker. This ability can be achieved either by compromising the hardware or the software of the object.

2.1.3 Diffie-Hellman Key Exchange

A number of bootstrapping solutions are based on the Diffie-Hellman (DH) key exchange protocol [56] to secure the communications between the IoT device and the user by establishing a symmetric key exchange. This protocol performs the secret key agreement through the sharing of public parameters, referred to as DH public keys.

2.1.3.1 Modular Exponentiation Diffie-Hellman

In a pairwise scenario, the two participants, Alice and Bob, generate separately their private keys, a and b. Then, they exchange the DH public keys, $g^a \ mod \ p$ and $g^b \ mod \ p$, where g is an element from the cyclic group $\mathbb{G}$ and p is a big prime. A cyclic group is defined as follows:

Definition 4 (Cyclic Group)**.** The group $\mathbb{G}$ is cyclic if and only if every element of $\mathbb{G}$ can be expressed as the power of one element of $\mathbb{G}$.
$\exists g \in \mathbb{G}, \forall h \in \mathbb{G} : h = g^n$ for some $n \in \mathbb{Z}$ and we denote that $\mathbb{G} = \langle g \rangle$.

For instance, one of the main currently used groups is the non-zero integers modulo a prime p group $(\mathbb{Z}_p^*, .)$. The two values, g and p, are also assumed to be known by the adversary. Afterwards, each participant uses the received public key to compute the secret DH key $K = (g^a)^b = (g^b)^a$, as illustrated in Figure 2.1. The adversary cannot retrieve the private keys, a and b, from the publicly exchanged keys, g^a and g^b. This is explained by the computational unfeasibility of solving the Discrete Logarithm Problem (DLP) that is defined as follows:

Definition 5 (Discrete logarithm problem)**.** Let $\mathbb{G}$ be a cyclic group of order N, with a generator g. The DLP is:
Given $y \in \mathbb{G}$, find an integer x such that $y = g^x$.

The exchanges ❶ and ❷ in Figure 2.1 are assumed to be performed on an insecure channel, referred to as In-Band channel, when we adopt the Dolev-Yao threat model. The Dolev-Yao adversary can block the message ❶ and replace it with his own fraudulent public DH key $g^{a'}$ to force Bob to compute the key $K_B = (g^{a'})^b$. This action is repeated in the message ❷ to force Alice to compute the key $K_A = (g^{b'})^a$. At the end of the protocol execution, the adversary is in complete control of the communication channel

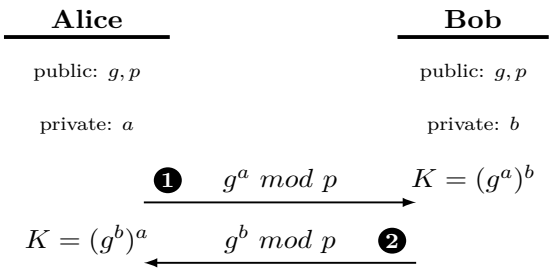


Figure 2.1: Diffie-Hellman key exchange protocol using modular exponentiation

since he has forced the honest pairing participants to establish the secret key with him, as illustrated in Figure 2.2. This Man-in-the-Middle attack requires the honest participants to perform a key verification step to ensure that the DH public key exchanges have been correctly received.

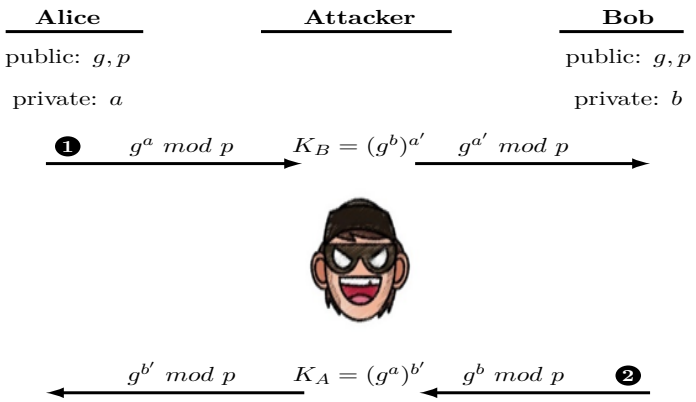


Figure 2.2: Man-in-the-Middle attack on the Diffie-Hellman protocol

2.1.3.2 Elliptic Curve Diffie-Hellman

An Elliptic Curve (EC) is a group of coordinates (x, y) on a graph that satisfy the equations $y^2 = x^3 + ax + b$ and the discriminant $\Delta = 4a^3 + 27b^2 \neq 0$. The values x, y, a, b are elements within the finite field $\mathbb{F}_p$ where p is a prime larger than 3. All the algebraic operations within this field, such as point multiplication and addition, result in a point within the field. For instance, the Bitcoin curve secp256k1 is represented in Figure 2.3 and takes the form:

$$y^2 = x^3 + 7$$

The Elliptic Curve Diffie-Hellman (ECDH) is similar to the modular exponentiation version, described in Subsection 2.1.3.1. Instead of using the modular exponentiation, the ECDH uses the point multiplication on the curve that is based on the following property:

$$(a * G) * b = (b * G) * a$$

The point G on the elliptic curve is the generator and the integer values, a and b, represent respectively the secret key of each participant in a pairwise scenario. In this case, the ECDH public keys are $(a * G)$ and $(b * G)$. At the end of the ECDH protocol, the two participating entities derive the shared secret key is $K = (a * G) * b = (b * G) * a$. The ECDH protocol shares the same weakness against the man-in-the-middle attack as the modular exponential version. Thus, a key confirmation step is essential to guarantee the correctness of the ECDH protocol execution.

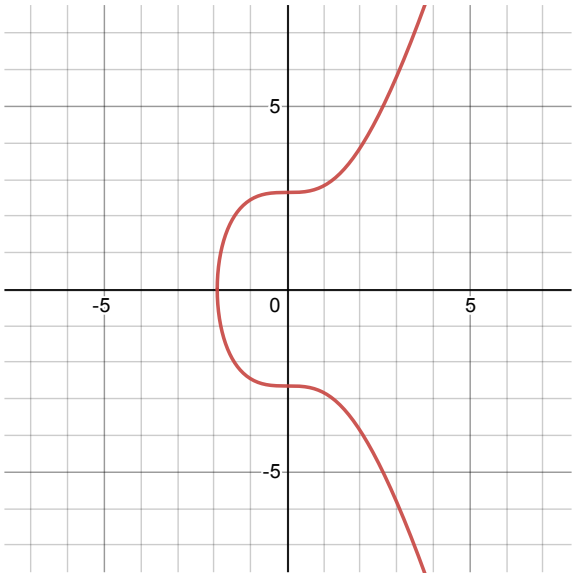


Figure 2.3: Elliptic curve secp256k1

2.1.4 Cryptographic Primitives

In this part, we introduce the properties of the cryptographic primitives that are commonly applied in the key agreement protocols according to the work of Laur and Nyberg [111]. These properties are used to perform the computational security proofs, as detailed in Section 3.1.

2.1.4.1 Keyed Hash Function

The keyed hash function $h : M \times K \rightarrow T$ has two arguments: the first one is the data to be hashed that comes from a word space M and the second one is the key from a key space K. This function provides an output in a tag space T and, depending on the construction of this cryptographic primitive, it can offer the following information theoretic properties:

- *universal:* For any two inputs $x_0, x_1 \in M$ such that $x_0 \neq x_1$, the probability $\Pr[k \leftarrow K : h(x_0, k) = h(x_1, k)] \leq \frac{1}{||h(x_0,k)||}$
- ϵ_u*-almost universal:* For any two inputs $x_0, x_1 \in M$ such that $x_0 \neq x_1$, the probability $\Pr[k \leftarrow K : h(x_0, k) = h(x_1, k)] \leq \epsilon_u$
- ϵ_u*-almost eXclusive OR (XOR) universal:* For any $x_0, x_1 \in M$ and $y \in T$ such that $x_0 \neq x_1$, the probability $\Pr[k \leftarrow K : h(x_0, k) \oplus h(x_1, k) = y] \leq \epsilon_u$

Also, the notion of almost regular functions has been identified in the case of sub-key manipulation $h : M \times K_a \times K_b \leftarrow T$ where K_a and K_b represent the sub-key spaces. The following definitions have been introduced:

- $(\epsilon_a, \epsilon_b)-$*almost regular* with respect to the sub-keys: For each input $x \in M$, $y \in T$ and sub-keys $\widehat{k_a} \in K_a$, $\widehat{k_b} \in K_b$, the probabilities $\Pr[k_a \leftarrow K_a : h(x, k_a, \widehat{k_b}) = y] \leq \epsilon_a$ and $\Pr[k_b \leftarrow K_b : h(x, \widehat{k_a}, k_b) = y] \leq \epsilon_b$
- ϵ_u-*almost universal* with respect to the sub-key k_a: For any two inputs $x_0, x_1 \in M$ such that $x_0 \neq x_1$ and $k_b, \widehat{k_b} \in K_b$, the probability $\Pr[k_a \leftarrow K_a : h(x_0, k_a, k_b) = h(x_1, k_a, \widehat{k_b})] \leq \epsilon_u$ where $\epsilon_u \geq \frac{1}{|T|}$

- *Strongly ϵ_u-almost universal* with respect to the sub-key k_a: For any two inputs $x_0, x_1 \in M$ and $k_b, \widehat{k_b} \in K_b$ such that $(x_0, k_b) \neq (x_1, \widehat{k_b})$, the probability $\Pr[k_a \leftarrow K_a : h(x_0, k_a, k_b) = h(x_1, k_a, \widehat{k_b})] \leq \epsilon_u$

- *Independence property*: Let x be a uniformly distributed variable over the word space M. Let $a \in 0, 1^l$ and b be an arbitrary value from the tag space T. The two hash function h_1, h_2 are assumed independent if they satisfy $\Pr[h_2(x) = a|h_1(x) = b] = \Pr[h_2(x) = a] = 2^{-l}$

2.1.4.2 Commitment Scheme

The commitment scheme is constructed using three algorithms :

- *The generation function Gen:* Generates the public parameters pk used by the commitment function.

- *The commitment function $Com_{pk} : M \times R \leftarrow C \times D$:* Transforms the input $m \in M$ and a random value $r \in R$ into a commitment string $c \in C$ and an open value $d \in D$.

- *The decommitment function $Open_{pk} : C \times D \leftarrow M$:* Reveals the value of the commitment string $m = Open_{pk}(c, d)$ for all $(c, d) = Com_{pk}(m, r)$. If the algorithm fails to open the commitment, it outputs a special error message $\perp$.

The security of these primitives is defined by a hiding and a binding game. These challenges are conducted against a t time adversary that tries to violate these properties. The attacker is represented by a function $A(x_1, .., x_n)$ that represents his knowledge $(x_1, .., x_n)$ as inputs to the algorithm. The commitment scheme is $(t, \epsilon_1)-$**hiding** if any t time adversary achieves the following attack success probability:

$$2\times\left|\Pr\left[pk \leftarrow Gen, s \leftarrow \{0,1\}, (x_1, x_0) \leftarrow A(pk), (c_s, d_s) \leftarrow Com_{pk}(x_s) : A(c_s) = s\right] - \frac{1}{2}\right| \leq \epsilon_1 \tag{2.1}$$

The commitment scheme is $(t, \epsilon_2)-$**binding** if any t time adversary achieves the following attack success probability:

$$\Pr\left[pk \leftarrow Gen, (c, d_0, d_1) \leftarrow A(pk) : Open_{pk}(c, d_0) \neq \perp \text{ and } Open_{pk}(c, d_1) \neq \perp\right] \leq \epsilon_2 \tag{2.2}$$

In addition, a commitment scheme is **non-malleable**, if given a commitment value c, the adversary is unable to generate a commitment vector $(c_1, .., c_n)$ that can be opened by a decommitment value d.

In the work of Pasini and Vaudenay [192, 149], there are two extra commitment properties introduced as follows:

- *Extractability*: There is a deterministic algorithm $extract(m, c)$, that reveals the value of the nonce r which is hidden along with a message m in the commitment value $c = Com_{pk}(m, r)$ when there exists a decommitment d such that $(r, m) = Open_{pk}(c, d)$.

- *Equivocability*: There are two deterministic algorithms $simcommit(m)$ and $equivocate(m, c, r, \phi)$. The former algorithm returns a fake commitment value c and an information ϕ. The latter one outputs a decommitment value d such that we obtain $(m, r) = Open_{pk}(c, d)$ from the information (c,ϕ) provided by *simcommit*.

Furthermore, they use, in [192, 149], the notion of a *random oracle commitment scheme* where the function $Com_{pk}(m, r)$ generates an l_e-bit value e, calls a hash function $H(e, r, m)$ and outputs the decommitment $d = (e, r)$. On the other hand, the decommitment function $Open_{pk}(m, c, d)$ simply verifies the hash $H(d, m) = c$ and uses d to retrieves r when the condition holds.

2.1.5 Security Properties

In the literature, a number of security properties have been evaluated to investigate the correctness of the key agreement schemes. However, there is a tendency to provide a different formulation under a different title of the authentication properties that drift away from the commonly known specifications. In order to present a clear overview of these security assessments, we match the outlined property with the adequate specification in the work of Lowe [120]. However, we keep the same property formulation as detailed in the original work to provide the reader with a better understanding of the originally conducted security assessment. Based on the definitions in [120], a brief description of the assessed security properties are presented as follows:

- *Weak agreement:* A protocol guarantees to a pairing participant, referred to as Alice, a weak agreement with another participant, referred to as Bob, if, whenever Alice completes a run of the protocol, apparently with Bob, then Bob has previously been executing the protocol, apparently with Alice.

- *Injective weak agreement:* A protocol guarantees to a pairing participant, referred to as Alice, an injective weak agreement with another participant, referred to as Bob, if it guarantees the weak agreement property and, additionally, each protocol run of Alice corresponds to a unique protocol run of Bob.

- *Non-injective agreement:* The initiator Alice completes a run of the protocol, apparently with Bob, then Bob has previously executed the protocol as a responder, apparently with Alice, and the two parties agreed, at the end of the protocol execution, on the same parameters.

- *Injective agreement:* A protocol guarantees to a pairing participant, referred to as Alice, an injective agreement with another participant, referred to as Bob, if it guarantees the non-injective agreement property and, additionally, each protocol run of Alice corresponds to a unique protocol run of Bob.

- *Key confidentiality:* Whenever a secret key x is computed by a pairing participant at a specific step i of the protocol, the adversary is unable to know x at any point of the execution.

2.1.6 Secure Bootstrapping Initiatives

The existing bootstrapping initiatives introduce numerous approaches to correctly perform the key establishment and the entity authentication phase. In the former phase, the presented techniques may or may not rely on the use of pre-shared security knowledge to either perform the key transport or the key agreement process. However, in the latter phase, the user is required to perform the entity authentication through the use of a proof of identity that is provided by the manufacturer of the IoT object. In this subsection, we study the applied methods of a selection of secure bootstrapping initiatives that are commonly used to perform the association process of the IoT devices to the network of the user. We aim at identifying the security properties that are provided by these protocols.

2.1.6.1 Open Mobile Alliance Lightweight Machine-to-Machine

The Lightweight Machine-to-Machine protocol [117] introduces a client-server approach to perform the bootstrapping. In the *provisioning* step, the server is responsible for providing the essential credentials to the client. Afterwards, the client device perform the *registering* step with one or more LwM2M server. The standard presents four bootstrapping approaches:

- *Factory Bootstrap:* The necessary bootstrap information are embedded onboard of the IoT device by the manufacturer prior to its deployment.
- *Smart card Bootstrap:* The necessary bootstrap information are retrieved from a smart card.
- *Client Initiated Bootstrap:* The client is responsible for retrieving the necessary information from the pre-configured bootstrapping server.
- *Server Initiated Bootstrap:* The server automatically configures the IoT devices once they connect to the network of the user.

2.1.6.2 Open Connectivity Foundation

The Open Connectivity Foundation (OCF) [147] defines the device provisioning phase, referred to as Owner Transfer Methods (OTM). In this step, the onboarding tool (user's device) and the IoT object share the necessary information to establish a secure communication channel. The adopted OTM can be specific to each manufacturer or it can be an implementation of an existing technique. The standard specifies the following OTMs:

- *Just Works:* The device perform an un-authenticated Diffie-Hellman key exchange that results in establishing a symmetric session key.
- *Random PIN:* The IoT object generates a 40-bit Personal Identification Number (PIN) that should be entered into the onboarding device.
- *Manufacturer Certificate:* The manufacturer embeds a digital certificate into the IoT object that is used, later on, to authenticate the device.
- *Vendor Specific:* The vendor is responsible for implementing their own transfer method according to their needs.

Afterwards, the two paired devices perform the ownership verification of the object before it gets authorized to join the network. The used identifier should always satisfy these three requirements: unique, immutable and verifiable. The *unique* requirement ensures that the identifier only authenticates one IoT object. The *immutable* requirement ensures that the identifier cannot be modified and it always authenticates the same device. The *verifiable* requirement ensures that the identifier can be easily verified by the user.

2.1.6.3 Wi-Fi Alliance Device Provisioning Protocol

The Wi-Fi Alliance Device Provisioning Protocol (DPP) [12] is used for the establishment of a secure and simple connectivity for the associated devices. The association process in DPP is referred to as *Provisioning* and it introduces two roles: *Configurator* and *Enrollee*. The former role represents the trusted device of the user that performs the provisioning with the IoT object, referred to as Enrollee. This procedure consists of three phases that should be executed sequentially:

I. *Bootstrapping:* The objective of this phase is to allow the enrollee to securely share his bootstrapping information with the configurator. The Enrollee should transfer these parameters, such as the ECDH public key, through an auxiliary channel. The elliptic curve Diffie-Hellman protocol is described in Subsection 2.1.3.2. The transfer can happen using a QR code scanner on the configurator or a Near-Field Communication (NFC) technology.

II. *Authentication:* The first objective of this phase is to allow the configurator to share his public key with the enrollee in case the bootstrapping exchange in the previous step was unidirectional (from the enrollee to the configurator). This phase can be initiated by the two roles and it permits the authentication of the public key of the responder by the initiator. The mutual authentication on the DH public keys could be possible if the bootstrapping exchange on the auxiliary channel was bidirectional.

III. *Configuration:* The objective of this phase is to permit the secure sharing of the configuration parameters between the configurator and the enrollee. These parameters may include the Service Set IDentifier (SSID) and the passphase of the Wi-Fi access point.

2.1.6.4 Fast IDentity Online Alliance

The Fast IDentity Online (FIDO) Alliance [49] describes a bootstrapping approach, referred to as *Device Onboarding*, that installs secret keys and configuration data into the IoT object. This protocol facilitates the secure deployment and interaction of the device with an IoT platform. The FIDO onboarding protocol introduces four roles:

- *The manufacturer:* This role performs the Device Initialize Protocol (DI) that inserts the FIDO onboard credentials into the IoT object during the manufacturing process. It creates the Ownership Voucher (OV) which is the identification information of the future owner. Thus, only the users who have the correct OV can perform the onboarding process.

- *The device:* This role represents the IoT object that must have a Restricted Operating Environment (ROE) in order to securely store the cryptographic credentials and to execute the FIDO operations. The ROE is a trusted execution environment that guarantees the confidentiality and the integrity of the computations that are conducted inside of it.

- *The owner:* This role represents the user that holds a valid OV to perform the onboarding process. The transfer of the voucher from the manufacturer to the owner has not been specified by the standard.

- *The rendez-vous server:* This role represents the authentication server that separately authenticates the owner and the device. Afterwards, this server provides the device with the IP address of the owner in order to perform the final authentication process.

The FIDO specifications describe four sub-protocols that are executed by each entity to facilitate the authentication between the device and the owner. These protocols are described as follows:

- *Device Initialize Protocol:* This scheme is executed by the manufacturer to onboard the IoT object with necessary credentials during the manufacturing process.

- *Transfer Ownership Protocol 0:* This scheme is executed by the owner to authenticate itself to the rendez-vous server in order to map the device identifier to its IP address.
- *Transfer Ownership Protocol 1:* This scheme is executed by the device to authenticate itself to the rendez-vous server in order to retrieve the owner IP address.
- *Transfer Ownership Protocol 2:* This scheme is executed by the device to perform the FIDO authentication directly with the owner using his IP address.

2.1.6.5 Nimble Out-of-Band Authentication for Extensible Authentication Protocol

The Nimble Out-of-Band Authentication (EAP-NOOB) [20] is a generic bootstrapping method that is intended for IoT devices without pre-configured authentication credentials. The objective of this protocol is to perform a key agreement between the participating entities based on the use of a user assisted auxiliary communication channel, referred to as an Out-of-Band channel (OoB) channel. This channel can be, for instance, a QR code scanning process or an NFC communication. The protocol start with the *Initial Exchange* phase by performing an ECDH key exchange between the IoT object and the user's device. Afterwards, the two entities carry out the user-assisted OoB phase that aims at conducting a key confirmation step by verifying the correctness of the computed DH shared key using the OoB channel. This protocol will be further described in Chapter 3.

2.1.6.6 Summary

According to the IETF report on the bootstrapping protocols [170], the existing solutions can be classified into three main categories:

- *Managed methods:* These mechanisms rely on pre-established security credentials such as pre-shared symmetric keys or digital certificates.
- *Ad-hoc methods:* These mechanisms assume that the IoT devices do not share any pre-established security knowledge with the end-user. Therefore, they perform the key agreement procedure, described in Subsection 2.1.1, in order to securely share credentials among the participating nodes.
- *Hybrid methods:* These mechanisms combine the two previously described categories in order to cover all the use-cases of the end-users.

The previously described bootstrapping initiatives have been classified according to the IETF categories in Table 2.1. These bootstrapping initiatives aim to establish a secure communication channel with an authenticated IoT device. However, only the managed methods that rely on a digital certificate embedded by the manufacturer can guarantee the entity authentication of the IoT device. The other methods only guarantee the correctness of the key establishment procedure by either performing a key transport or a key agreement protocol.

We assume that our resource-constrained IoT devices do not have any pre-established security knowledge during the first phase of the bootstrapping process. This assumption is motivated by the special nature of the IoT objects that do not support asymmetric encryption. Thus, it renders the use of digital certificates not suitable. Furthermore, the management of the certificates that are embedded on the deployed devices can be quite challenging, especially for a large scale deployment. Consequently, we focus on the ad-hoc solutions by studying both the key agreement and the entity authentication phases of the IoT bootstrapping. These phases are referred to respectively as Secure Device Pairing

(SDP) and Secure Device Enrollment (SDE). The security objectives of these two phases are complementary and crucial to guarantee the secure association of the IoT devices to the network of the user.

Table 2.1: Classification of the bootstrapping initiatives

Bootstrapping Initiatives	**Methods**	**Security Objectives**			**Method Category**	**Bootstrapping Category**
		Key Transport	Key Agreement	Entity Authentication		
Lightweight Machine-to-Machine	*Factory Bootstrap*	✓	✗	✓	Managed	Managed
	Smart Card Bootstrap	✓	✗	✗	Managed	
	Client Initiated Bootstrap	✓	✗	✗	Managed	
	Server Initiated Bootstrap	✓	✗	✗	Managed	
Open Connectivity Foundation	*Just Works*	✗	✗	✗	Ad-hoc (unsecure)	Hybrid
	Random PIN	✗	✓	✗	Ad-hoc	
	Manufacturer Certificate	✓	✗	✓	Managed	
	Vendor Specific	N/A	N/A	N/A	N/A	
Wi-Fi Alliance Device Provisioning Protocol	*Device Provisioning*	✗	✓	✗	Ad-hoc	Ad-hoc
Fast IDentity Online Alliance	*Device Onboarding*	✓	✗	✓	Managed	Managed
Nimble Out-of-Band Authentication	*EAP-NOOB*	✗	✓	✗	Ad-hoc	Ad-hoc

2.2 State-of-the-art of Secure Device Pairing

The secure device pairing is the first phase of the bootstrapping protocol. It consists of exchanging a secret key between the IoT object and the user without having any pre-shared secret information such as symmetric keys or digital certificates. Therefore, any two unidentified devices are able to perform this operation in a decentralized manner. This step is usually conducted using an additional communication channel, referred to as an Out-of-Band channel, that is not completely under the control of the adversary.

The second variant is based on the environmental events that are commonly sensed by the honest pairing participants in order to extract entropy. In the example presented in the Figure 2.4, the two devices are co-located within a specific area, referred to as the authentication zone. This area should be sufficient for two devices with off-the-shelf equipment to sense the same environment. The honest participants can exploit for example the random acoustic events around them to generate a key with sufficient entropy. These events cannot be perceived by the attacker since he is out of the safe zone. This area represents the zone where even an adversary with advanced equipment cannot sense the shared environment between the honest pairing participants. Hence, the safe zone is usually wider than the authentication zone. This concept is referred to as context-based pairing or Zero-Interaction Pairing (ZIP).

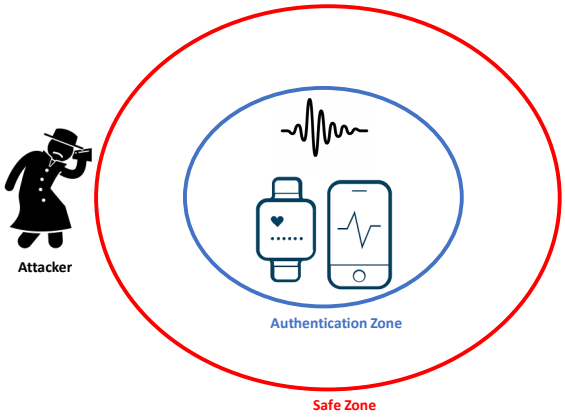


Figure 2.4: Context-based SDP scheme

2.2.1 Out-of-Band Pairing

The majority of the deployed secure device pairing solutions rely on an auxiliary channel with specific security properties to send information that validates what has been exchanged on the In-Band channel. The reason behind this diversity in the communication mediums is due to the proof, using BAN Logic analysis [37], that the authentication based on a single communication link controlled by a Dolev-Yao intruder, described in Subsection 2.1.2.1, is not feasible [48]. This powerful adversary is assumed to have a perfect knowledge of the protocol and he is able to overhear, block, delay, replay and forge any transmission over that channel. However, he is not able to perform any computational attacks against cryptographic functions without the knowledge of the secret parameters. As a consequence of adopting this intruder model, the usage of the main insecure channel without having pre-shared secrets is not sufficient to provide the desired security guarantees for the key exchange process. Therefore, there is a need for an auxiliary communication link on which the authentication of the exchanged keys can happen. These channels can be constructed based on audio, visual or haptic transmissions, as detailed in Subsection 2.2.1.2.

Due to their special nature and their communication properties, they provide an initial level of security that is sufficient to primarily guarantee the integrity of the data and the demonstrative identification [21], which is ensuring that the communicating devices on these channels are the intended ones for pairing. Other security objectives might be provided in some cases such as the confidentiality and the data origin authenticity. These assumptions on the OoB channel reduce the attacker capabilities in comparison with his abilities on the main insecure channel. In this context, we adopt the Out-of-Band security classification from the work of Mirzadeh et al. [134] that defines the three following categories: the *confidential* channel which eliminates all attacker capabilities, the *protected* channel that limits the adversary powers to intercepting, blocking and delaying the messages which breaks the confidentiality assumption and affects the guarantee of the message reception. Finally, the *authentic* channel grants the attacker the additional capability to replay messages that were exchanged in previous sessions which violates the data freshness guarantee [176]. Some proposals such as Secure Simple Pairing (SSP) [32] and Push Button Configuration (PBC) [11] exploit short-range radio communications like NFC [140] as an auxiliary channel. Unfortunately, this technology is not secured against an attacker that is sufficiently close to the pairing objects as demonstrated in the work of Akter et al.[7]. Thus, we do not consider it as a secure option for an OoB channel. In the work of Fomichev et al. [66], a selection of pairing proposals that rely on Out-of-band channels have been thoroughly described based on the nature of the auxiliary channel (radio [11, 32, 40], visual [171, 71, 161, 209], acoustic [73, 182] or haptic [183, 113, 172]), the degree of the user involvement and the application context of the pairing. The latter criteria classes the pairing use-cases into categories that have related security threats and objectives.

2.2.1.1 Out-of-Band Threat Model

In this study, we adopt the Dolev-Yao intruder model on the In-Band channel where he has complete control over the network. We assume that the attacker is able to perform the following actions: **overhear**, **block**, **delay**, **replay** and **forge** any message on the channel. This latter action includes a modification attempt on a previously captured legitimate message. Due to the absence of any pre-established security information, the attacker has the same level of knowledge as the legitimate devices which eliminates any possibility of performing a secure key establishment using only the In-Band channel, as proved in [48] using BAN logic analysis [37].

This is obviously not the case for the Out-of-Band channel since it is assumed by

design to be partially out of reach of the adversary. Therefore, it should guarantee at least the integrity and the data origin authenticity of the messages. Also, the confidentiality property on the OoB channel, referred to as *Private OoB* [134], is demanded by some SDP schemes ([71], [80]). This assumption is hard to obtain and might ultimately lead to vulnerabilities in the protocol design [21]. The OoB channels reduce the attacker capabilities to overhearing, blocking and delaying the authentication strings. Thus, the adversary cannot replay or forge a message without being exposed. These restrictions result in an authenticated Out-of-Band channel that is referred to as *Public OoB* [134]. In some cases, the attacker might be given the capability to replay previously sent messages on the Out-of-Band channel and it is referred to as *Weak OoB* [134].

Unfortunately, under the assumption that we have no prior security knowledge between the legitimate devices and the assumption that the attacker has perfect knowledge of the protocol execution, it is not realistic to assume that an adversary is only able to replay a message without having the power to forge a suitable one and send it on the peer-to-peer Out-of-Band channel, as adopted in a great body of research work. We state that, based on this logic, any SDP scheme that allows an adversary to replay but not to inject their own messages under the assumption that we have no pre-shared secret are ultimately vulnerable. Therefore, while considering the presence of a vigilant user, we model our attacker capabilities by only three actions: **overhear**, **block** and **Inject** any exchange on the OoB channel. The latter action includes the transmission of either a previously captured or a freshly constructed message. Also, the **delay** capability can be hard to achieve directly over the peer-to-peer Out-of-Band channel without considering the combination of the block and the replay actions. However, it can be considered possible using the attacker capability to perform this action on a previous exchange over the In-Band channel that was intended to trigger the OoB transmission. In this case, the act of delaying the previous insecure exchange results in stopping the protocol execution for the same amount of time which, consequently, leads to a delay over the reception of the OoB transmission. Therefore, this action targets the protocol execution in order to affect the Out-of-Band channel which affects any protocol that has an In-Band exchange prior to the OoB transmission. As an example of a protocol structure that is immune against this malicious act, the well-known device pairing scheme, *Talking to Strangers* [21], starts by a bidirectional OoB exchange of the public key hashes which, according to our model, it does not grant the adversary the power to perform a delay attack. In order to target all the cases, we consider the delay as an action that is dependent on the protocol structure instead of the OoB channel specifications.

These previously described actions are assessed to evaluate the following security objectives on the Out-of-Band channel that we deem necessary to guarantee the required security of the OoB exchange under our adversary model:

- *Confidentiality (C) [176]:* The information, sent over the channel, can only be accessed by the authorized pairing parties. Therefore, the attacker cannot **overhear** the communication.
- *Data Freshness (DF) [176]:* The information, sent over the channel, cannot be replayed by a malicious actor. Therefore, the attacker cannot **inject** any old messages on the channel.
- *Data Origin Authentication (DOA):* Any receiver of the information, transmitted on the channel, is able to authenticate its sender. Therefore, the attacker cannot **inject** his own messages on the channel as if they were coming from a legitimate sender.
- *Liveness (L) [13]:* Any information, transmitted over the channel, is eventually received by the intended party. Therefore, the attacker cannot **block** any transmission

over the channel.

- *Channel Availability (CA):* Any information, transmitted over the channel, is received at the intended protocol execution order. Therefore, the attacker cannot **delay** any transmission over the channel.

Based on these five security goals, we can conduct a more refined and realistic Out-of-band channel classification. We have six main channel categories:

- **Confidential OoB:** All the security goals are guaranteed. Therefore, the adversary has no capabilities.
- **Delayable-Confidential OoB:** Only the channel availability assumption is not guaranteed. Therefore, the adversary can only delay the transmission.
- **Protected OoB:** Only the confidentiality goal does not hold. This means that the attacker is only capable of overhearing the communication.
- **Delayable-Protected OoB:** Only the confidentiality and the channel availability goals do not hold. This means that the attacker is only capable of overhearing and delaying the communication.
- **Authentic OoB:** Only the integrity, the data freshness, the data origin authentication and the channel availability goals are achieved. Therefore, the adversary is capable of blocking and overhearing the OoB channel.
- **Delayable-Authentic OoB:** Only the integrity, the data freshness and the data origin authentication security goals are achieved. Therefore, the adversary is capable of blocking, delaying and overhearing the OoB channel.

The confidential channel represents the most secure channel since it achieves all the security goals desired. On the other hand, the delayable-authentic represents the minimum required OoB channel to ensure the security of the device pairing process, as shown in Table 2.2.

Table 2.2: Attacker capabilities on the In-Band and Out-of-Band channels

Channel type	Adversary powers				Achieved security goals					
	Overhear	*Block*	*Inject*	*Delay*	*Confidentiality*	*Integrity*	*Data freshness*	*Data origin authentication*	*Liveness*	*Channel availability*
In-Band channel	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
Confidential OoB	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓
Delayable-Confidential OoB	✗	✗	✗	✓	✓	✓	✓	✓	✓	✗
Protected OoB	✓	✗	✗	✗	✗	✓	✓	✓	✓	✓
Delayable-Protected OoB	✓	✗	✗	✓	✗	✓	✓	✓	✓	✗
Authentic OoB	✓	✓	✗	✗	✗	✓	✓	✓	✗	✓
Delayable-Authentic OoB	✓	✓	✗	✓	✗	✓	✓	✓	✗	✗

2.2.1.2 Out-of-Band Security Classification & Usability Analysis

The majority of the existing pairing solutions rely on an auxiliary channel with specific security properties to send information that validates what has been exchanged on the In-Band channel. The reason behind this diversity in the communication channel usage is that the authentication based on a single communication link is not feasible using BAN Logic analysis [37]. The authors of [48] have proven that the *"Key-based device authentication between two previously unknown mobile devices in an ad-hoc computing environment is not possible using only a single wireless communication channel"*. Therefore, the use of only the main insecure channel is not sufficient and there is a need for an auxiliary

channel on which the authentication of the exchanged keys can happen. The Out-of-Band communications can be constructed based on audio, visual or haptic transmissions and their goal is to guarantee the integrity of the transmitted information.

The major limitation of these channels is their low data rate which means that transferring long hashes or keys is not possible. In the work of Fomichev et al. [66] the described communication properties of the chosen Out-of-Band channels contradict the previous declaration. This fact is, simply, explained by the absence of the dedicated hardware on the commercial IoT devices due to cost optimization factors. therefore, this constraint explains the long completion time of a 15 bit OoB exchange conducted in the work of Kumar et al. [107].

Some of the proposed schemes rely, more extensively, on the human user to interact with the devices and either *relay*, *compare* or *generate* an information. These interactions make him the communication medium, known as human-computer-interaction channel [66]. The security objectives are assessed based upon the user behavior which makes them prone to Human-factor error that, if not well designed, might compromise the effective security of the protocol and its performance [95].

In this section, we present both the security and the usability properties for a selection of the most common Out-of-Band channels based on our refined adversary model. Furthermore, we briefly introduce some of the existing schemes that take advantage of each of the selected OoB channels. Finally, the five security goals, defined in the adversary model in Section 2.2.1.1, are used to classify these chosen channels based on the security they offer while taking into account the presence of a vigilant user, as summarized later in Table 2.3.

2.2.1.2.1 Near Field Communication

The NFC is a wireless communication technology used for point-to-point exchanges between two devices under the condition of *close physical proximity* as shown in Figure 2.5. These devices can be active or passive [140]. NFC chips are widely deployed and they are used in a wide variety of IoT devices.

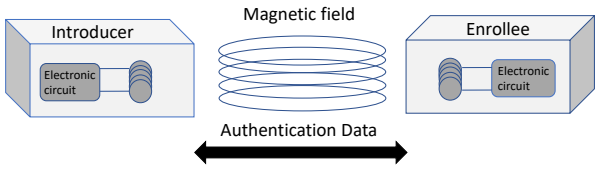


Figure 2.5: Communication model of NFC technology

I. **Usability Properties**

As stated previously, NFC requires the two devices to be in a close proximity which means that the user is required to have a minimal intervention of putting the objects close to each other. The Line of Sight (LoS) transmission is not required which eliminates the need for a major user involvement in the case of aligning the two pairing parties. Due to its non-perceptibility property, this technology relies on the user vigilance to make sure that there is no suspicious behavior around them which is quite hard, especially for non-expert users. This requirement represents a burden on the user and a drawback when it comes to the user-friendliness aspect.

II. **Security Properties**

The devices using NFC chips can be active in order to act as a contactless card reader or communicate with another object. They can also be passive in the case of a static

message carrier such as a hash of a key or a password. This means that the risk of unauthorized readings can lead to a practical relay attack [68].

From a security perspective, the close proximity assumption plays a major role in protecting the devices from a sufficiently distant attacker since he is considered unable to overhear or interfere on the communication. Unfortunately, it has been proven possible in [212] where an eavesdropping attack on a commodity NFC-enabled mobile device has been successful from a distance up to 240 *cm*. Furthermore, a Man-in-the-Middle attack has been demonstrated in [7] between two NFC-enabled devices separated by a 10 cm distance. Hence, an attacker can always violate such requirement which does not make this Out-of-Band channel any better than the In-Band channel because of its similar communication properties.

III. **Proposed Schemes**

(a) **Push Button Configuration (PBC)** is part of the standardized WI-FI Protected Setup [11] that introduces a pairing scheme using two options:

- **Password Token:** The Enrollee device transmits a 32 byte random password to the NFC-enabled Registrar. The same password is used with the In-Band registration protocol to provision the Enrollee with WLAN configuration data.
- **Connection Handover:** The two NFC-enabled devices exchange the hashes of their Diffie-Hellman public keys (exchanged previously on the In-Band channel) using NFC to verify that they are communicating with the same device that was involved in the near field communication

(b) **Secure Simple Pairing (SSP):** is part of the standardized Bluetooth Secure Simple Pairing [32] that introduces a pairing scheme using an Out-of-Band option:

- **Out of Band:** After the discovery phase via Bluetooth, the cryptographic authentication parameters as well as the identification information (Bluetooth Device Address) are sent over the OoB channel in order to attempt to mitigate the MitM attacks.

2.2.1.2.2 Radio-Frequency IDentification Channel

The Radio-Frequency IDentification (RFID) is a wireless communication technology used for both indoor and outdoor identification. These systems consist of small *tags* that emit stored identification information when interrogated by an RFID *reader* which makes them a sort of automatic identification system [198]. The majority of the used RFID tags are *passive* since they rely on the energy emitted by the RFID readers, as shown in Figure 2.6. We can find *active* tags having their own power supply on-board, which makes them able to establish a bidirectional communication channel.

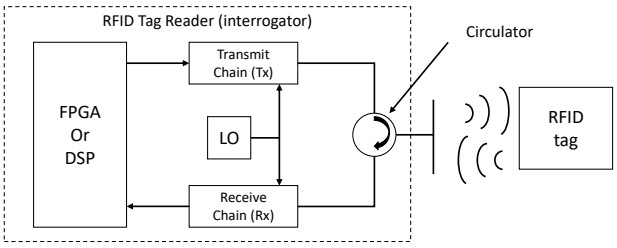


Figure 2.6: Block diagram of a RFID communication system [89]

I. **Usability Properties**

This technology does not require any human intervention when using high frequencies which makes it more user-friendly and more appealing to non-expert users. On the other hand, for the low frequencies, it has the same requirements as the NFC technology.

II. **Security Properties**

For the **low frequencies**, RFID has similar security properties to the NFC technology. As for the **high frequencies**, the range of the passive reads increases to reach 10 meters which makes an attacker able to retrieve the identification information and relay it since that kind of tags is very constrained and it responds to any reader [198]. Including the active tags and their long range ($> 100\ m$), this technology offers similar communication properties to what is used for the In-Band channel. This makes the adversary in total control of the communication as stated in our adversary model in Subsection 2.2.1.1.

III. **Proposed Schemes**

Noisy Tag [40]: Injection of intentional noise, using an extra RFID tag (nosy tag), into an authentic channel making the eavesdropping process meaningless for the adversary. Only the legitimate reader (owner of the noisy tag) is able to retrive the original message from the noisy emitted signal. One downside to this scheme is that it does not protect the tag against an active attacker. It assumes that the active attacks require the adversary to be closer to the tag than in the case of eavesdropping and such active distance requirement can be circumvented by natural barriers, e.g, in private areas (user surveillance, house, office, building).

2.2.1.2.3 Millimeter Waves

The Millimeter Waves (MM-Waves) is a wireless communication technology that operates on the Extremely High Frequency (EHF) range. The high frequencies and their propagation properties make them useful for applications such as the transmission of large amount of data, cellular communications and radar [86]. A standard IEEE 802.11ad [17] enables multi-gigabit wireless communications in the unlicensed 60 GHz band [10], as shown in Figure 2.7. This band is considered ideal for a variety of indoor applications since it supports data rates up to 7 *Gbps* [10].

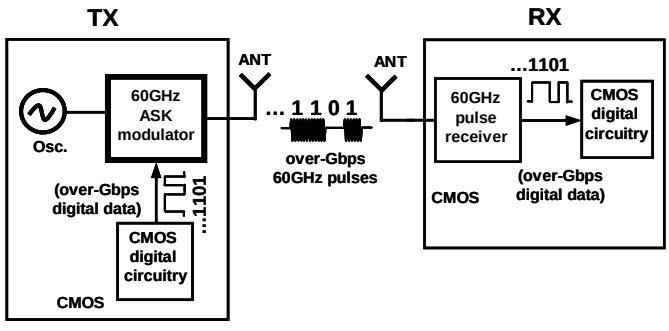


Figure 2.7: Block diagram of a Millimeter wave communication system with a 60GHz ASK modulator [146]

I. **Usability Properties**

The short range requirement, similar to the NFC technology, forces the user to be in close proximity of the two devices and to be vigilant of their surroundings in the

covered area. Alongside the penetration characteristic, the act of pairing devices from a distance is not feasible which is not convenient in the case of a smart-home containing multiple deployed IoT devices. As for the LoS condition, a user intervention during the pairing is crucial in order to setup the devices to face each other for a proper communication.

II. **Security Properties**

The short-range, penetration and LoS characteristics of the MM-Waves provide a highly secure operation. This has been explained by the unfeasibility of a simple eavesdropping attack since the adversary has to be in the same room which would expose him to our vigilant user. However, as presented in [184], eavesdroppers can successfully intercept even highly directional transmissions using small-scale objects (from coffee cups to cell phones) as reflectors. These properties make the MitM attack hard for the attacker especially in a closed area where the walls create a natural barrier to the MM-Wave emissions.

III. **Proposed Schemes**

There are not many devices that support MM-Waves, e.g [2], but their popularity is on the rise. The previously described pairing schemes PBC from the standardized WPS [11] uses MM-Waves as an Out-of-Band channel to perform the authentication process and it has been implemented on the HP Advanced Wireless Dock (HP Elite x2 1011 G2 [3]). Even though the original version of the PBC scheme is vulnerable to MitM attacks, the close physical proximity, LoS and no-penetration characteristics of the MM-Waves forces the attacker to be co-present which exposes him even by a benign user.

2.2.1.2.4 Visible Communication

The Visible Communication (VC) is a wireless communication technology that relies on modulating the visible spectrum using an illumination source such as a display or an LEDs to transmit data. The short-range property of this technology is explained by the propagation distance of the emitting interface [139]. This technology includes multiple practices such as the use of a display-camera setup that shows a specific message (a QR code or a short authentication string) in order to create a short-range, interference-free Out-of-Band channel. The characteristics of the channel are directly dependent on the size of the screen to provide a good viewing quality from multiple angles and the quality of the camera to guarantee a better detection, e.g., PixNet [152]. However, this option assumes the existence of a display and a camera on the transmitter and the receiver side which is not always the case for the low budget IoT devices. On the other hand, we can find the most common and most easily constructed variant that is referred to as Visible Light Communication (VLC). A one-way VLC channel is described in Figure 2.8 as three main components: a transmitter, a channel and a receiver.

I. **Usability Properties**

Similarly to the NFC and the Millimeter waves, the short range requirement forces the user to be in close proximity of the two devices and to be vigilant of their surroundings in the covered area. This monitoring act is more feasible from a user perspective since he is able to perceive any light emissions coming from an unauthorized source (potentially malicious).

Alongside the penetration characteristic, the act of pairing devices from a distance is not feasible which is not convenient in the case of a smart-home containing a wide

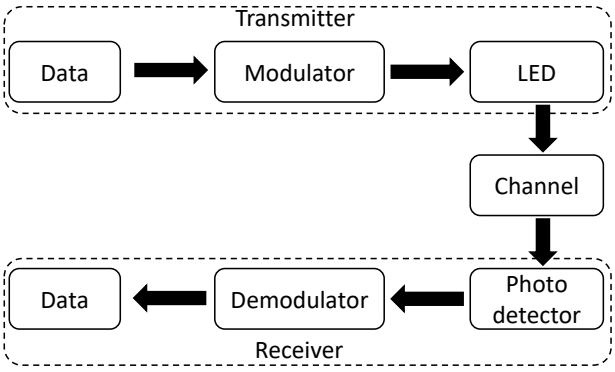


Figure 2.8: Block diagram of a VLC communication system [139]

variety of devices. As for the LoS condition, a user intervention during the pairing is crucial in order to setup the devices to face each other for proper communication.

The devices to be paired have to be equipped with at least a LED and a photo-sensor in the case of a bidirectional communication which is not the case for the constrained IoT products. On the other hand, the majority of devices are equipped with a display capable of performing the transmission but not a camera which means that the communication channel can only be unidirectional.

II. **Security Properties**

Even though VLC might seem secure by design against eavesdropping especially when taking into account the LoS requirement and the no-penetration of solid objects such as the walls of the smart-home. It has been proven in [47] that this attack is feasible and easy to perform through the door gaps, the keyholes and the windows. These attack scenarios make use of the reflections of the light emissions and they provide low to no Bit Error Rate (BER) depending on the modulation scheme used by the transmitter.

Also an adversary can use a directional light to alter the transmitted message by sending pulses to the photo-sensor. This attack won't be of a great impact on the pairing process and it would only lead to a Denial of Service (DoS) without compromising the key agreement. However, this technique might be useful to block the reception of the light pulse by saturating the photo-detector on the receiving side.

One major threat when using a Display-Camera communication is the risk of replay attacks. This malicious act targets the liveness of the video captured by the camera. The attacker can easily record a previous conversation between a camera-enabled phone and an IoT object with a display using shoulder surfing or CCTVs [60]. Then, he replays the video to the camera in a way to pair with it. One solution to this issue, which has been proposed in [157], is the comparison of the inertial measurements taken by the phone during the transmission and the motion analysis captured on the recorded video as better described in Figure 2.9.

The data freshness property can be assured by the unfeasibility of any injection attacks on this Out-of-Band channel when the user vigilance assumption is assumed. In addition, the perceptibility of the light emissions and the Los requirement facilitate the monitoring of the area surrounding the legitimate devices.

III. **Proposed Schemes**

(a) **Blinking Light [171]:** After exchanging the key between the devices on the In-Band channel, a checksum value is sent from a LED-equipped device to a

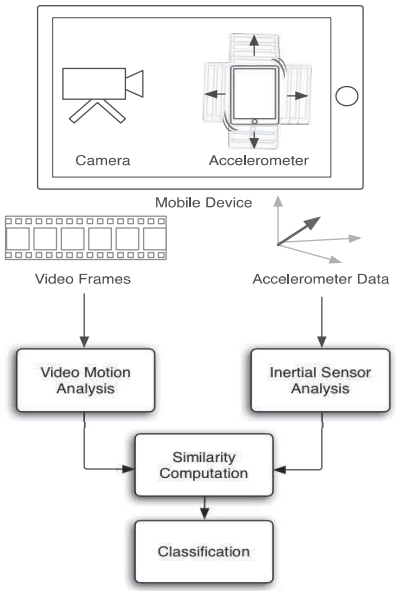


Figure 2.9: Classification of replay attack using Video Motion Analysis and Inertial Sensor motion Analysis [157]

camera or a photo-sensor equipped device using light pulses. The size of the checksum vary between 24 bits with an execution time of 5 to 8 seconds and 32 bits with an execution time of 15 seconds. These values are not consistent with the results in [107] where the authors re-implemented the pairing scheme with a 15 bit OoB message and measured an average completion time equal to 28.8 *s*.

(b) **KeyLED [161]:** Two devices use LED/photo-sensor pair to set up a short distance visible light communication channel with a raw bit rate of 500 *bps* and transmit their ECC public keys (352 bits) using on-off keying.

(c) **Flashing Displays [105]:** It utilizes two channels, wireless radio as an In-Band channel and a unidirectional VLC, where the former is considered as insecure and the latter is used as Out-of-Band. A VLC is established between the display of a smartphone and a light sensor of a constrained device once it is on top of the screen.

(d) **Secure Barcode-based Visible Light Communication (SBVLC) [209]:** a full duplex VLC channel between two camera/display-enabled devices using 2D barcodes. This technique is suitable for device pairing since the main focus of the desired Out-of-Band channel is the data integrity and not the confidentiality. The barcode can represent the authentication information such as the hashes of the exchanged DH public keys.

2.2.1.2.5 Acoustic

An audio channel is an acoustic networking system that exploits audible sounds to construct a low-bandwidth communication link using a speaker that generates audio snippets and a microphone that records them, as illustrated in Figure 2.10. Numerous modulation techniques have been used such a the Dual-Tone Multi-Frequency (DMTF) and the On-Off Keying (OOK) to enhance the reliability of the channel.

I. **Usability Properties**

The reliability of these channels depends on multiple factors such as the acoustic environment surrounding the devices since the ambient noise drastically increases

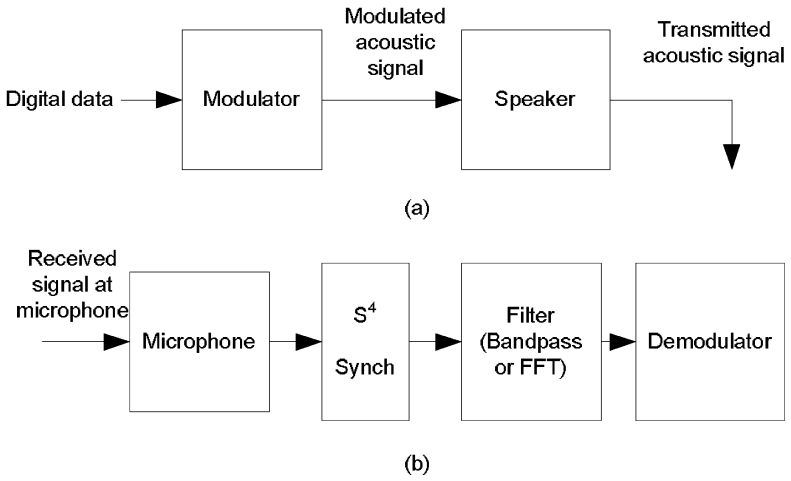


Figure 2.10: Block diagram of an acoustic communication system: (a) Modulator/Transmitter (b) Demodulator/Receiver [94]

transmission errors. Also the sensitivity of the receiver (microphone) and the distance between the communicating nodes affect the correctness of the signal reception. Based on these factors, the channel requires a human assistance in order to place the devices in a close proximity, to make sure the ambient acoustic environment is suitable for this type of channels and, most of all, to monitor the acoustic transfer against any malicious attempt to interfere with the transmission.

II. **Security Properties**

The feasibility of an eavesdropping makes the confidentially assumption on these channels out of reach, as demonstrated in the work of Halevi et al. [78] using off-the-shelf equipment. Furthermore, the high applicability of a relay attack, as demonstrated in [181], makes the user vigilance during the transmission a necessity.

One of the main advantages of this channel is that an attack is easily detected by a user close to the legitimate devices which prevents any active malicious attempts to interfere with the authentication message transmission.

III. **Proposed Schemes**

(a) **Loud&Clear [73]:** The scheme starts by a Diffie-Hellman key exchange over the main insecure channel and, then, the devices exchange the hashes of the public keys encoded in MadLib sentences that are verifiable by the user. Finally, the user confirms whether or not the sentences match on both devices. This protocol can also work on a speaker-display enabled pair of objects where the sentence sent by the speaker of the first one is displayed on the second one.

(b) **HAPADEP [182]:** The scheme starts by sending the encoded Diffie-Hellman public keys on the audio channel using fast codec which provides faster transmission rate but is meaningless to the user. The key verification phase happens also on the audio channel where an audio sequence that is recognizable by the user and that is related to the exchanged public keys is transmitted from each node using slow codec and, then, the devices wait for the user to confirm the match.

2.2.1.2.6 Haptic

A haptic channel is constructed using low frequency mechanical waves that result in a tactile sensation. This type of channel can be either built using only the communicating devices, for example the use of vibrations to transmit a message [163], as illustrated in Figure 2.11a, or it can be a consequence of a user interaction with the objects, for example by applying a pattern of button presses on the devices [183]. Recently, another variant of SDP protocols has emerged. These schemes rely on the haptic channel that is based on the physical contact between the pairing participants through the body of the user [115, 160], as shown in Figure 2.11b. This Out-of-Band channel is referred to as Body-Coupled Channel (BCC) [164] and this pairing context is also known as Wireless Body Area Network (WBAN) or Body Sensor Network (BSN) as detailed in the work of Ali and Khan [8].

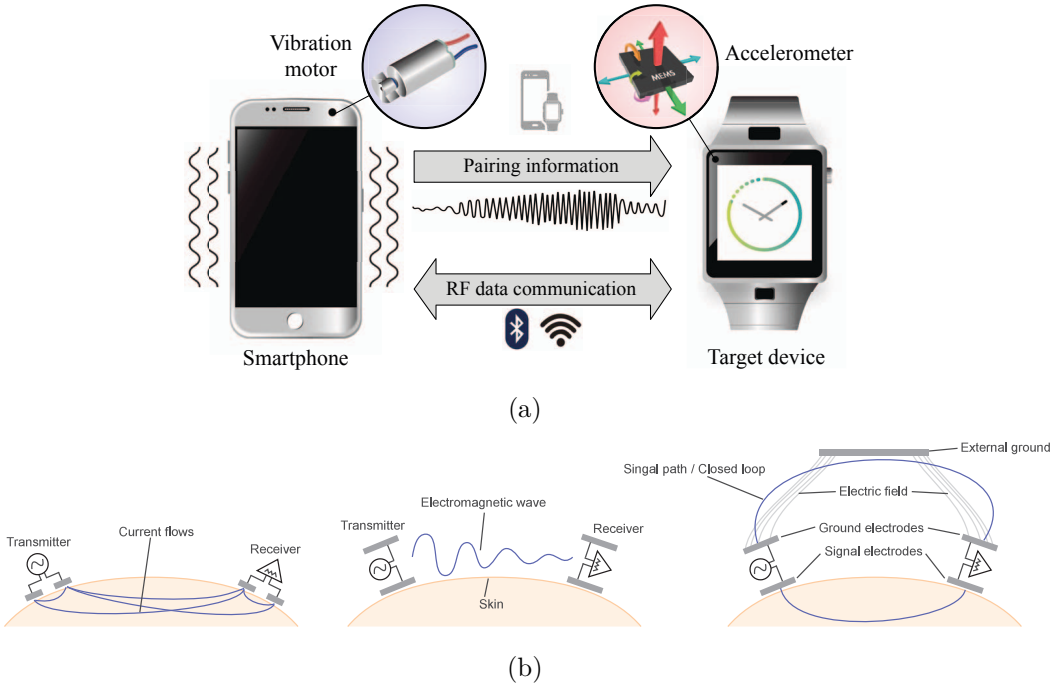


Figure 2.11: Examples of haptic out-of-band channels: (a) Haptic Out-of-Band channel based on the physical vibrations [113], (b) Types of body-channel communication: galvanic coupling, surface wave and capacitive coupling [160]

I. **Usability Properties**

The haptic channels tend to demand an extensive user involvement since in most cases he needs to intervene and apply a physical action on one or both devices or to monitor any suspicious vibrations coming from an external source. Also, the use of a vibration motor can be costly when it comes to energy-constrained devices.

However, the fact that the mechanical waves can hardly pass through thick solid objects, such as walls, makes the transmission limited to the physical barriers around the devices, for example a room. The fact that the communicating objects have to be in direct contact eases the surveillance of the vibrational transfer since the user is only required to focus on the same restricted area.

II. **Security Properties**

Similar to the audio channels, the confidentiality assumption on these channels no longer holds since they have been proven vulnerable to eavesdropping through acoustic side channel attacks [78]. Due to the necessity of establishing a physical contact

between the devices, either by a user intervention or using mechanical waves, an injection attack can be easily detected which guarantees the integrity and the origin authenticity of the exchanged messages. Also, this channel is the only one that is resistant to blocking which makes it the only one that is assuring the liveness property.

III. **Proposed Schemes**

(a) **Vibrate-to-Unlock [172]:** the scheme establishes a secret between a smartphone and an RFID tag using a 14 bit PIN sent through vibration. That secret information, generated by the smartphone, is required by the tag to identify the legitimate reader.

(b) **BEDA [183]:** this scheme takes advantage of the user intervention to apply a physical action (button press) on the devices.

- The first variant of this protocol requires the user to establish the same pattern of button presses on both devices (at least seven presses) where these objects take advantage of the random inter-event timing, that is almost equal on each of them, to extract 21 secret bits.
- The second variant only requires the user to follow a pattern of signals emitted by the first device (pulses of light, vibrations or beeps) and apply it on the second device using a button. This scheme represents a variant of the protocol MANA III [71] which requires the confidentiality of the PIN entry process. This means that if an adversary is able to witness the pattern of button presses then he can recompute the 21 secret bits and eventually corrupt the protocol.

(c) **Body-channel based secure device pairing [160]:** this protocol is based on the capacitive coupling to establish the body communication channel. It has two main phases:

- *Key agreement*: the two pairing participants establish a secret key K through the Diffie-Hellman key agreement protocol [56].
- *Key confirmation*: each one of the devices emits a keyed hash of the authentication parameters used through an electrode that is in touch with the human body in order to confirm the correctness of the previous step, as illustrated in Figure 2.12.

Table 2.3: Channels classification based on the achieved security goals

Out-of-Band channel	Confidentiality	Integrity	Data Freshness	Data origin authentication	Liveness	Channel classification
NFC	✗	✗	✗	✗	✗	In-Band
RFID	✗	✗	✗	✗	✗	In-Band
MM-Waves	✗	✓	✓	✓	✗	Authentic
VC	✗	✓	✓	✓	✗	Authentic
Audio	✗	✓	✓	✓	✓	Protected
Haptic	✗	✓	✓	✓	✓	Protected

2.2.1.3 Limitations

The significant limitations of these channels are their low data-rates and their need for an extensive user intervention. The former drawback is due to the quality of the interfaces on the commercial IoT products, which makes the transfer of long hashes or keys not

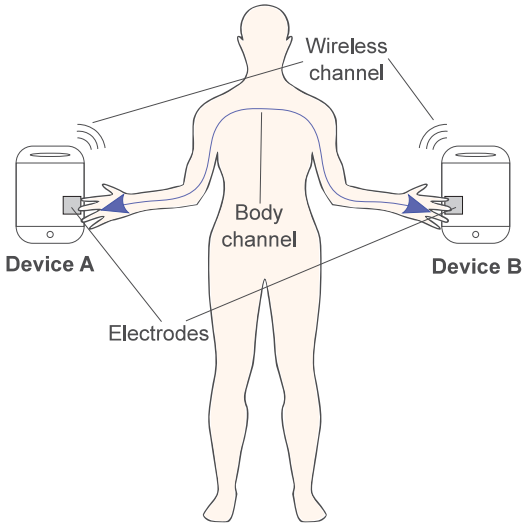


Figure 2.12: Body-channel based secure device pairing [160]

possible. Some of the proposed schemes rely on the human user to *setup* the devices for the exchange, to *relay* an information from one device to another, to *compare* a short authentication string on both objects or to simply *generate* a secret PIN and to enter it in both devices [66]. As an example, the security of the pairing scheme MANA III [71] is based on the confidentiality of the PIN entered by the user. Even though the confidential OoB channels are not considered as a reliable option due to the feasibility of eavesdropping attacks on the acoustic, the visual and the haptic transmissions using side-channel analysis techniques [78]. Another prominent threat in the protocol design is the predictable human input. This vulnerability is considered as a *Human-factor error* that, if not well designed, might compromise the effective security of the protocol [95]. As illustrated in Table 2.3, the RFID and the NFC technologies are considered as In-Band channel where the adversary has the ability to violate the desired security properties with the appropriate hardware equipment. Furthermore, in most cases, the correctness of the pairing process requires the use of an auxiliary channel that guarantees at least the integrity, the freshness and the origin authenticity of the exchanged information. Thus, the use of MM-Waves, visible, acoustic or haptic communications can be a promising solution to construct an OoB channel with the desired security properties. However, off-the-shelf IoT devices do not commonly integrate the needed interfaces in order to exploit these communication technologies which limits the adoption of such SDP schemes. A selection of the highlighted Out-of-Band pairing protocols will be studied in depth in Section 3.1.

2.2.2 Context-based Pairing

The use of the Out-of-Band channels introduces a number of usability challenges such as the time-consuming pairing process and the extensive human involvement as shown in [95, 107]. Therefore, the research focus has shifted toward a more autonomous authentication technique based on a proof of co-presence. These protocols use the ambient environment to extract a contextual information on both devices within a specific area called the *authentication zone*. It represents the area where the legitimate devices are required to be placed in order to enhance the usability of the protocol by minimizing the errors when sensing the environment. The contextual information can be used to extract a key for encryption later on [130], a fingerprint of the device location [92] or as a way to encode a secret between the pairing parties [204]. Based on the close proximity assumption,

the two objects are expected to have similar measurements of the chosen environmental metrics, which results in a similar contextual security parameters. The choice of the metrics should be based on aspects such as:

i. **Location dependency:** The measurements computed from the contextual information are specific to the location of the device.

ii. **Static randomness:** The random changes in the metrics should allow a static device to extract a contextual information with a sufficient entropy.

iii. **Dynamic randomness:** The random changes in the metrics should allow a moving device to extract a contextual information with a sufficient entropy.

iv. **Unpredictability:** An attacker should not be able to predict the values produced by the metrics at a specific location.

v. **Time invariance:** The changes in the contextual feature do not have a periodic nature

vi. **Availability:** The locations where the feature is available (e.g. urbain-indoor, urbain-outdoor, everywhere)

There are multiple context-based schemes that use the audio as a source of randomness such as [175, 97, 75]. In the work of Schürmann et al. [175], the authors used an audio fingerprint of the energy fluctuation between the frequency bands coupled with a fuzzy commitment [93] in order to exchange a key between two co-located devices. Also, the work of Karapanos et al. [97] exploits the acoustic environment by computing a similarity score using the average of the maximum cross-correlation of audio samples applied on a set of one-third octave bands. This result is then compared to a fixed threshold to decide the co-presence of the devices. This metric is based on the unpredictability of the acoustic signals received in the dynamic scenarios where these schemes were tested. Unfortunately, this choice does not satisfy most of the previously mentioned criteria such as the location dependency and the static randomness in quiet environments. In the work of Fomichev et al. [67], it has been proven that the microphones heterogeneity increases drastically the error rates of the contextual pairing, which makes the scheme less robust against contextual attacks. Also, we can never discard the risk of *audio amplification*, as discussed in [175], where the adversary uses a directional microphone to amplify the audio signals, which makes him able to reconstruct the fingerprint and get hold of the shared secret.

Another variant of contextual protocols relies on a number of metrics from the ambient radio environment as a proof of physical proximity such as the Receiver Signal Strength Indicator (RSSI) [173, 130, 191, 150, 90] and the Channel State Information (CSI) [131, 119, 203, 204]. These protocols are based on the assumption that devices within a close range and using a high frequency radio technology perceive the same unpredictable changes in the signal strength in short periods of time. Therefore, they are able to extract high entropy contextual information that can be ultimately used to exchange a secret or to derive an encryption key. This hypothesis satisfies our three main criteria mentioned above but it has been recently proven in [181] that the RSSI can be manipulated by the adversary. This attack has been demonstrated using a fake Wi-Fi access point on which the transmission power is adapted to the location of the target device so that it computes the wanted signal strength indicator. On the other hand, the CSI measurements represent the propagation of the signal in terms of scattering, fading and power decay with respect to their physical location. This metric becomes rapidly de-correlated between two devices as the distance between them increases. It is also highly unpredictable due to its

dependency on the ambient environment as shown in [204]. Such properties of the CSI are used to provide a high random bit generation rate that can reach hundreds of bits per second. The authenticity and the confidentiality of the CSI-based secret are guaranteed against a passive attacker outside the *safe zone*. However, its resilience in the face of an active adversary is still considered under investigation since it has been theoretically proven feasible in the work of Zafer et al. [208].

2.2.2.1 Limitations

The context-based techniques enhances the usability of the pairing procedure by limiting the required user intervention. However, these protocols tend to assume that the pairing is performed in a secure environment. Thus, the adversary is assumed unable to collect the same contextual information as the legitimate devices. This assumption can be hard to guarantee in a public environment depending on the chosen contextual feature. Furthermore, a number of context-based proposals rely on contextual features that require a high degree of activity such as the acoustic environment. Therefore, the pairing procedure would take a considerable amount of time to correctly perform the entropy extraction when the devices are in a low-activity environment. This limitation has been demonstrated in the work of Han et al. [79] by evaluating their context-based pairing proposal, called Perceptio, in a quiet environment. This protocol can take up to 21 hours to extract a 128-bit key on two co-located devices. In addition, these protocols require a precise synchronization between the contextual measurement of the devices which can be a challenging task.

2.3 State-of-the-art of Secure Device Enrollment

After the successful completion of the SDP phase, the enrollment procedure is responsible for verifying the identity and the origin of the device. In order to correctly conduct the entity authentication process, we need to implicate the manufacturer to provide us with a proof of identity that uniquely identifies the device. A number of enrollment solutions have been proposed based on a variety of initial assumptions that may not cover all the possible use-cases. Thus, in this section, we describe the categories of the existing techniques and we evaluate their suitability to our IoT deployment context.

2.3.1 Overview of the Secure Device Enrollment Techniques

2.3.1.1 Identity-based Solutions

The Identity-based solutions rely mainly on the identity-based cryptography that is defined, according to RFC 5091 [174], as follows:

Definition 6 (Identity-based cryptography)**.** The identity-based cryptography is a public-key encryption technology that allows a public key to be calculated from an identity, and the corresponding private key to be calculated from the public key.

The public keys that are produced using this cryptosystem are not randomly generated which is the case for the traditional public-key systems. The work of Salman et al. [168] introduces an identity-based scheme for IoT devices. The protocol uses a public identifier and a private key, that are generated by the object, to compute a suitable public key by the authentication server during the registration phase. The registration parameter transfer from the IoT device to the server is performed using the server's public key. The authentication is conducted using the identity-based asymmetric encryption parameters that have been established between the authentication server and the object in question.

Unfortunately, this solutions requires the use of an asymmetric encryption which is not suitable for our resource-constrained devices.

2.3.1.2 Certificate-based Solutions

The certificate-based technique uses a digital certificate that is embedded by the manufacturer on the IoT device in order to perform the authentication. This method would represent a promising solution when combined with a lightweight asymmetric cryptographic algorithm that is supported by an IoT device. However, in our case, we are dealing with resource-constrained devices that do not support the asymmetric encryption. Furthermore, we prefer to avoid exploiting any pre-established security knowledge between the user and the IoT object to facilitate the integration of our solution with an ad-hoc secure device pairing scheme [130, 103]. The no prior secret condition is motivated by the challenges related to the management of the digital certificates among a growing number of deployed IoT devices.

2.3.1.3 One-Time-Password Solutions

A third possible alternative is to exploit a One-Time Password (OTP) system [91] to authenticate the object. This technique is based on the use of a trusted third party entity that establishes a secret on the IoT object and the user's trusted device prior to the authentication phase. Afterwards, the IoT device provides a proof of identity to the user that is computed based on that shared secret information. Nevertheless, this technique requires the IoT device to communicate independently with a remote OTP server. As a consequence, we would prefer to perform the authentication process prior to the association of the IoT object to the network of the user. Therefore, the OTP solution would not be compliant with our requirements.

2.3.1.4 Hardware-based Solutions

A final alternative is to use a hardware-based enrollment protocol that relies on a secure element, such as a Physical Unclonable Function [16], on-board of the object. This method provides a lightweight and cost-effective authentication system that is adequate with the IoT context. Several integrated circuit vendors have opted for a hardware-level technology approach for securing the use of the IoT object through a PUF. These hardware secure elements serve multiple objectives such as the device identification, the secure key management and the secure boot functionality. This technology has been applied to the IoT products but it can, also, play a major role in the security systems applied to other industrial areas such as the vehicular context, as discussed in [16]. This role can cover, for example, the vehicle component identification [197] or the cryptographic key management for securing the vehicular ad-hoc network [16].

Unfortunately, a growing number of the recently proposed PUFs, such as the Interpose-PUF [143] and the Double Arbiter Physical Unclonable Function (DAPUF) [122], have been proven vulnerable against a variety of machine learning attacks that aim at modeling their behavior by collecting a sufficient number of Challenge-Response Pairs (CRP) [165, 200]. Therefore, several enrollment protocols have intentionally exploited some vulnerable PUF architectures to create a ML model that simulates its behavior [128, 116]. The work of Pour et al. [153] has briefly discussed the benefits of exploiting these modeling methods in an industrial scenario. These advantages include the reduction of the time that is required to enroll a big number of devices and the storage space that should be used to store the challenge-response pairs. As a consequence, the server can efficiently handle an increasing number of deployed IoT devices. The existing review studies of PUF-based enrollment procedures tend to focus on the traditional use of these hardware circuits through the

storage of the CRPs [46, 61]. Other surveys concentrate on reviewing the vulnerabilities of these PUF architectures against ML modeling attacks [166, 65, 101]. However, we have noticed that they overlook the exploitation of these ML modeling techniques in order to reduce the required storage space while maintaining the same level of security.

2.3.2 Physical Unclonable Function

A Physical Unclonable Function is a secure element that identifies, in a unique manner, a specific device through a challenge-response process. This pair of information represents the pattern of responses when we have a set of specific challenges as inputs. This function has to be unclonable and unique for each device since it relies on the physical randomness that can be either explicitly introduced or intrinsically presents in the physical system [23]. The micro variations in the hardware system allow the same construction of a PUF to provide unique responses when deployed on different circuits. Thus, these variations play the role of the seed to a random response generator.

There are two major categories of PUFs that are based on the source of the randomness. The first category, referred to as *electronic PUFs*, relies on a number of micro physical parameters that are hidden from the physical observation inside the electronic circuit. However, these parameters can be detected only when needed to produce the unique responses. These variables include the time, the frequency, the current or the voltage, the bistable states and the capacitance [9]. The second category, referred to as *non-electronic PUFs*, represents the PUF elements that rely on unique characteristics of the physical system in a non-electronic manner such as the use of light in the optical PUFs [51] or the radio variations in the RF-PUF [44]. Readers that are eager to learn more about the different PUF architectures can consult the survey [132]

The electronic PUF elements can be further classified into two categories: *Strong PUFs* and *Weak PUFs*. The former category provides a large space of challenge-response pairs which makes them suitable for the authentication operations. This is explained by the possibility to conduct numerous authentication attempts using different CRPs with each session without the need to reuse the same credentials. Thus, it represents an interesting candidate solution in the context of multi-user IoT objects. The latter category, *weak PUFs*, provides a smaller number of CRPs. However, these PUFs have been increasingly popular as internal key generators [88, 110]. In this work, we focus on the authentication protocol that are based on Strong PUFs.

2.3.2.1 Arbiter PUF

The Arbiter PUF [112] is one of the most popular electronic PUF that are exploited for the authentication operations. This PUF construction is based on the comparison of the travel time between two electrical signals propagating down two symmetrical paths. The uniqueness of the responses is based on the manufacturing variations in the creation of these two paths. This PUF is constructed using a pre-determined number of 2-2 cells that connects these paths. The choice of the connected routes depend entirely on the l challenge bits $C[x], x \in [1, l]$. Finally, the arbiter component decides which signals has arrived first and, accordingly, outputs the associated binary response, as illustrated in Figure 2.13.

2.3.2.2 XOR Arbiter PUF

This PUF architecture is a variant of the previously described Arbiter PUF. It has been developed as a way to enhance the complexity of the mapping function between the input challenges and the output responses. As highlighted in Figure 2.14, this construction uses

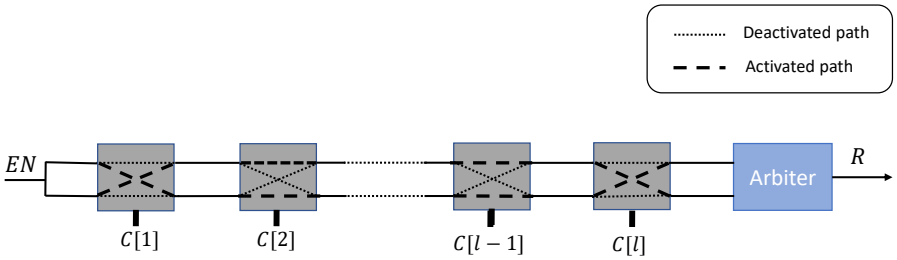


Figure 2.13: Arbiter PUF architecture

n independent Arbiter PUFs and it applies an XOR operation on their individual responses to obtain the output response R. However, the stability of the responses is highly affected by the increased number n of applied Arbiter PUFs.

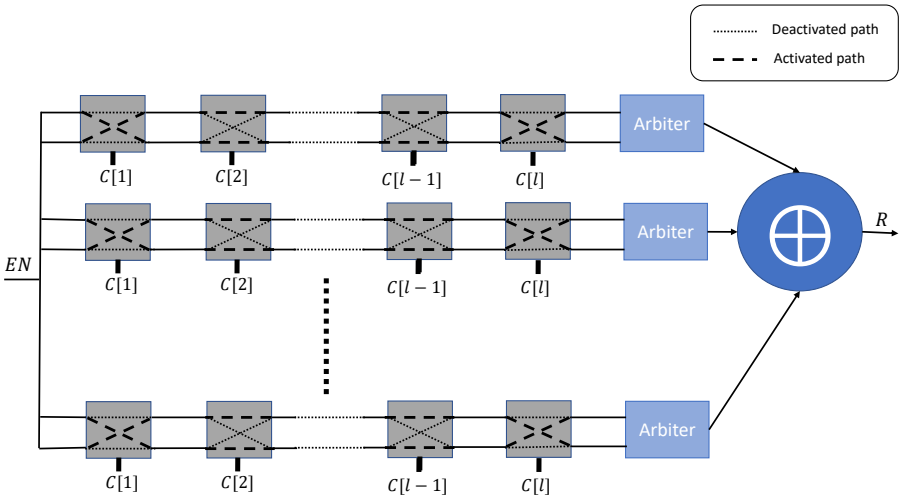


Figure 2.14: n-XOR Arbiter PUF architecture

The work of Yu et al. [207] has presented another variant of the XOR Arbiter PUF by applying a different challenge on each stage. The used n challenges can be constructed by applying a Linear-Feedback Shift Register (LFSR) to the received root challenge C, as shown in Figure 2.15.

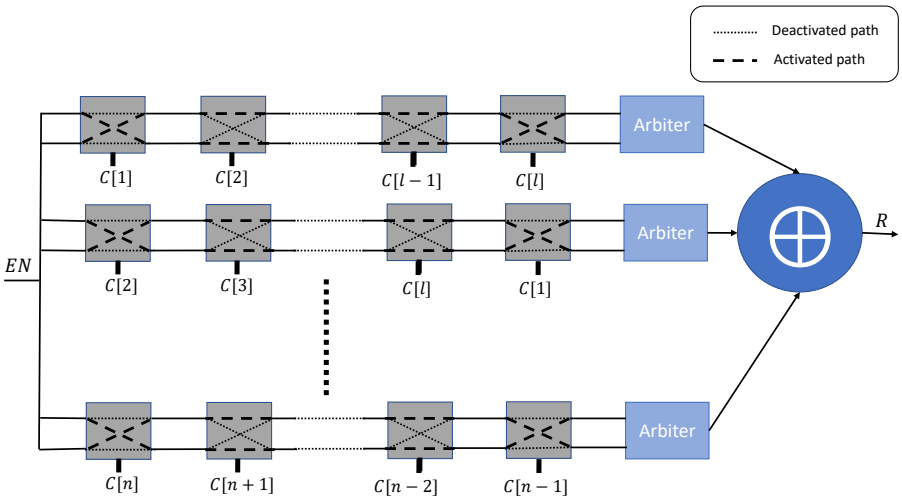


Figure 2.15: n-XOR Arbiter PUF variant with a derivative challenge per stage

2.3.2.3 Logically Reconfigurable PUF

The Logically Reconfigurable PUF (LR-PUF) [98] represents a hardware secure element that has the ability to change its challenge-response behavior. The reconfigurability aspects can be achieved in the context of integrated circuits through the use of a Field-Programmable Gate Array (FPGA). These PUF circuits should guarantee two properties: *Forward* and *Backward-unpredictability*. The former property assures that the challenge response pairs collected before the reconfiguration are invalid. Thus, the adversary cannot model the current PUF behavior through the use of previously collected CRPs. The latter property guarantees that an adversary with access to the current reconfigured PUF cannot estimates the responses before the reconfiguration. The work of Liu et al. [118] has identified two type of LR-PUFs: *Circuit-based Reconfigurable PUF (C-RPUF)* and *Algorithm-based Reconfigurable PUF (A-RPUF)*. The former category uses reconfigurable components onboard of the circuit to change the original construction. Thus, this hardware level modification changes the behavior of the PUF. The latter category keeps the original hardware components and, instead, it applies a configurable algorithm to change the mapping between the challenges and the responses.

2.3.3 Modeling of PUF Designs

The PUF circuits are considered vulnerable to modeling attacks using machine learning techniques. In this subsection, we present the most effective modeling approaches that can learn the behavior of these hardware security elements.

2.3.3.1 Logistic Regression

The Logistic Regression (LR) is a well-known supervised learning technique. This method models the probability of a discrete outcome that is associated to specific input variable. The LR learning algorithm is based on the sigmoid function and a set of weights that are learned by using the training dataset. The logistic regression technique is commonly used for the binary classification problems. Therefore, this methodology has been applied, in [65, 101, 200], to model the behavior of a binary output PUF such as the Arbiter PUF variants, described in Subsections 2.3.2.1 and 2.3.2.2.

The Resilient Propagation (RProp) [158] has been an increasingly popular algorithm to optimize the weight coefficients of the Logistic Regression technique. This is due to its ability to dynamically adapt the step size, independently, for each weight. This technique has been applied in the work of Rührmair et al. [165, 167] to model the x-XOR Arbiter PUF with $x \leq 5$ and with an accuracy that reaches 98%. Furthermore, the work of Khalafalla and Gebotys [101] has exploited a LR learning technique with a linear decision boundary against a more complex Arbiter PUF variant (DAPUF [122]). This method has yielded a enhanced modeling accuracy up to 99% with less challenge-response pairs and with cheaper computing resources.

2.3.3.2 Support Vector Machine

The Support Vector Machine (SVM) algorithm [33] has been widely used in classification tasks. The objective of this technique is to find an optimal hyperplane in a N-dimentional space that separates the data points. This hyperplane should classify the data points in a way that maximizes the distance between the identified classes. The Figure 2.16 illustrates a binary classification problem where the optimal hyperplane is represented as a continuous line. However, the dashed lines represent other candidate hyperplanes that do not provide the maximum margin between the two classes. Due to the popularity of the SVM algorithm in the binary classification tasks, it has been used

in numerous research work [154, 165, 167, 108, 101] to model some variants of the Arbiter PUF with limited complexity.

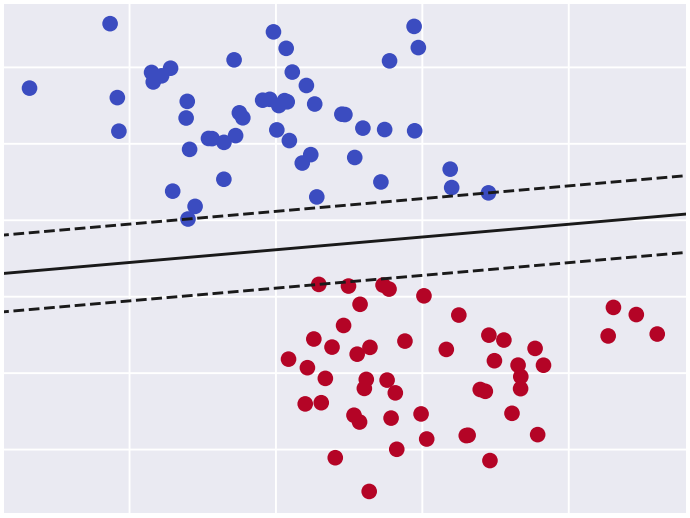


Figure 2.16: Binary classification problem using Support Vector Machine algorithm

2.3.3.3 Artificial Neural Networks

The Artificial Neural Networks (ANN) [193] are a system that imitates the function of the human brain through the use of multiple artificial neurons. This system consists of a number of neuron layers that are referred to as an input layer, one or multiple hidden layers and the classifier layer, as illustrated in Figure 2.17. Each neuron in the network is connected to another and has an associated weight and a threshold. These parameters are updated over time based on the training data to improve the prediction accuracy of the neural network model.

The ANN models, that consist of a single hidden layer, are referred to as Single Layer Perceptron (SLP) and they are only applicable in the case of linearly separable data. Therefore, the Multiple Layer Perceptron (MLP) are used in the case of non-linear problems. In the context of PUF modeling, a great body of work exploit the power of these models to either attack the state-of-the-art PUF constructions or to demonstrate their resiliency against ML modeling attempts. Unfortunately, a growing number of the proposed ML-resistant PUFs, such as the Interpose-PUF [143] and the 9-Xor Arbiter PUF [185], have been proven vulnerable against a variety of ANN attacks that aim at modeling their behavior by accessing a sufficient number of challenge-response pairs [74, 169, 100, 200].

2.3.3.4 Evolutionary Strategies

The Evolutionary Strategies (ES) are a stochastic technique for the numerical optimization of non-linear and non-convex learning problems. This class of ML methodologies is inspired from the biological evolution of individuals due to specific environmental conditions, also referred the survival of the fittest. In the context of the PUF technology, this individual is represented by a vector of runtime delays in the circuit components. The algorithm generates random PUF instances that are referred to as parents. They are tested to check the resemblance to the target PUF responses using a fitness function that should be specified by the user. Afterwards, the children instances inherent the parents characteristics (delay vectors in the case of Arbiter PUFs) with minor random modifications and the resemblance process is conducted for many generations.

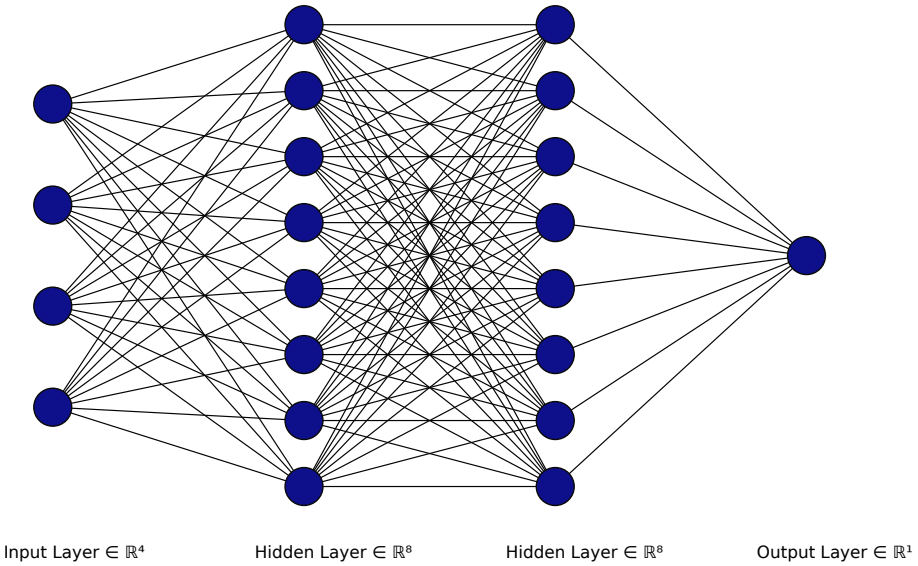
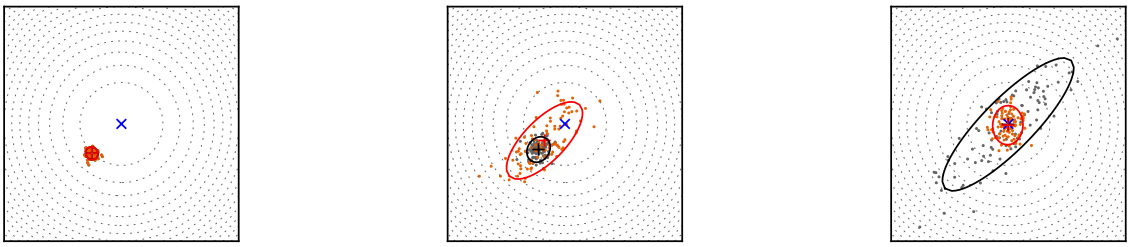


Figure 2.17: Artificial Neural Network architecture

The Covariance Matrix Adaptation-Evolutionary Strategies (CMA-ES) [83] is one of the most known ES that performs well on complex optimization problems. This variant uses the covariance matrix to adjust the dependencies between the variables in the normal distribution. Figure 2.18 illustrates the steps of the CMA-ES technique. The algorithm starts by generating random parent individuals according to the normal distribution. Afterwards, the fittest candidates are selected based on a specific fitness function and the algorithm updates its internal parameters. Finally, a new population is generated based on the previous updates and the process is repeated until convergence.



(a) Iteration 1 (b) Iteration 3 (c) Iteration 5

Figure 2.18: Optimization problem of a 2 dimensional linear function using the CMA-ES algorithm. The orange and gray dots represent, respectively, the distribution for the child and the parent population. [195]

The CMA-ES technique has been widely used to model complex PUF architectures such as the Interpose-PUF [143]. This algorithm has also been applied in the work of Becker [25] to attack the two versions of the Slender-PUF [128, 162]. The adversary has targeted the response obfuscation mechanism in order to use the obfuscated CRPs to efficiently model PUF circuits.

2.3.3.5 Other Machine Learning Techniques

Other ML techniques such as Decision Tree, Random Forest and Naïve Bayes classification have been applied to model the behavior of the PUF circuits [65, 106]. These techniques have been previously adopted to address the signal classification and network-

ing problems [155, 96]. However, in the work of Kroger et al. [106] they have been demonstrated less effective in comparison with the previously described algorithms when using a relatively large dataset of challenge-response pairs (more than 400 CRPs). On the other hand, these techniques have achieved a better accuracy when the training is performed on a small dataset (less than 400 CRPs).

2.4 Conclusion

In this chapter, we have described the two phases of the secure bootstrapping process and their associated security objectives. We have introduced the necessary technical concepts that are applied. In addition, we have conducted a literature review of the state-of-the-art pairing. We have introduced the two main categories of the secure device pairing procedure: The Out-of-Band and the contextual pairing. Primarily, we have focused on the Out-Of-Band channels by providing a refined adversary model that is suitable for the ad-hoc pairing context. Based on these refinements, we have proposed a new Out-of-Band channel classification by evaluating a number of security guarantees such as the confidentiality, the data freshness, the integrity, the data authenticity, the liveness and the channel availability. The objective of this thesis is to study the limitations of these state-of-the-art bootstrapping solutions and to investigate for possible improvements that can enhance their performances. Afterwards, we have conducted a literature review of the state-of-the-art enrollment approaches and we have identified the use of hardware security as a promising candidate in the case of resource-constrained IoT devices.

In the next chapters, we study in-depth these two bootstrapping phases and we introduce our proposed SDP and SDE protocols that tackle the identified state-of-the-art limitations. Therefore, we focus, primarily, on guaranteeing the secrecy and the integrity of the data that is exchanged between the user and the IoT device. For this purpose, we rely on the two pairing techniques in order to enhance the security of key agreement process. Secondly, we aim at providing the entity authentication of that particular object based on the use of machine learning techniques and physical unclonable functions. However, we do not consider that the legitimate devices have been compromised due to a firmware vulnerability. Thus, the software security issues are considered out of the scope of this work.

3 | Secure Device Pairing

Contents

With the growing numbers of personal devices, sensors and actuators, the use of a decentralized device-to-device communication system has become a necessity for numerous applications in the context of the Internet of Things. Therefore, the protection of this communication channel requires a secure key establishment protocol between the devices. This device pairing process ensures that the communicating nodes agree on the same symmetric encryption key, which represents an initial trust establishment between devices that have no pre-shared knowledge (a shared password or a symmetric key). The no prior secret condition is motivated by the challenges of exploiting a public key infrastructure due to the growing numbers of deployed IoT devices.

Two main techniques are used to achieve these goals, as presented in Section 2.2. The first one uses a pre-authenticated auxiliary channel, location limited or human assisted, usually referred to as an Out-of-Band channel [21]. However, these channels most importantly suffer from low data-rates, which is not optimal when it comes to pairing time. This drawback can severely affect the user-experience and the optimization of this usability criteria is considered a necessity for such protocols. The second technique ensures authentication through a proof of co-presence based on the ambient environment and is better known as context-based pairing or zero-interaction protocols [67]. Even though this type of pairing schemes is optimal in terms of usability and user-friendliness, it demands a safe zone where no attacker is assumed present to avoid any risks related to facing a well-equipped adversary. This can be quite hard to guarantee by a regular user and quite easy to take advantage of by an adversary that can hide a sensor in that allegedly safe environment.

In addition, we need to analyze the security proofs that were conducted on OoB-based secure device pairing protocols. This study aims at identifying the adopted protocol hypotheses and the security properties that are required to guarantee the correctness of

the ad-hoc key agreement procedure. The formal and computational verification methods have been used lately to assess the security of the SDP schemes since they provide a rigid and thorough evaluation of the soundness of a protocol. Thus, even subtle defects can be systematically uncovered based on a set of specifically defined protocol hypotheses and security properties. As a consequence, the inaccurate formalization of these properties and the protocol modeling would lead to incorrect verification results.

In Section 3.1, we conduct a survey on the existing formal and computational analysis that were conducted on state-of-the-art SDP protocols. In Section 3.2, we describe our hybrid secure device pairing protocol that combines the contextual pairing approach with the use of an Out-of-Band channel to provide an enhanced security. Based on the findings of the survey that has been conducted in the previous section, the robustness of our secure pairing proposal has been formally validated. In Section 3.3, we provide the future SDP protocol designer with a set of security recommendations and open research directions that have been discussed throughout the chapter. In Section 3.4, we conclude the first part of the thesis.

3.1 Analysis of the State-of-the-art SDP Security Assessments

In this section, we study a selection of existing formal and computational security proofs that are conducted on secure device pairing schemes. This review lays out the definitions of the chosen security properties, the adopted verification model, the associated protocol assumptions and an assessment of the verification results. Although every analysis tends to use different terminologies and definitions, we normalize the used taxonomy in order to enhance the understanding of these security validations. In addition, we describe an advanced threat model that consists of violating two security guarantees: *the demonstrative identification* and *the device integrity*[1]. This adversary model has yield a recently published attack, called *misbinding* [177], that affects the security of the existing device pairing schemes. This section aims at introducing and motivating the use of the formal and the computational security analysis in the process of validating the robustness of the secure device pairing schemes. Also, it serves as a road map for properly designing an SDP protocol that achieves the desired security goals and that can be applicable to realistic scenarios by providing the adequate criteria for choosing the appropriate Out-of-band channel. In addition, it sheds light on the recently discovered attacks and vulnerabilities that affect the robustness of the SDP protocols.

3.1.1 Security Analysis Under the Non-Invasive Threat Model

3.1.1.1 Description Framework

The Out-of-band based device pairing protocols have two main building blocks. The first one is the Out-of-Band channel which constitutes the most important security aspect. The second one is the protocol design that is represented by the cryptographic computations and the exchanges on the In-Band channel. In the literature, there are two different aspects when it comes to describing these types of pairing schemes. The first one focuses on the nature of the Out-of-Band channel by highlighting its communications, security and usability properties. The second aspect focuses on the protocol design by taking advantage of different cryptographic techniques while abstracting the OoB part to a channel that provides precise security goals as described in Subsection 2.2.1.1.

[1]The device integrity outlines that one of the pairing participants is under the control of the adversary

In this part, we will presents a selection of OoB-based device pairing protocols that provide a *formal* or *computational* security analysis based on the adopted threat model that is described in Subsection 2.1.2.1. Based on the existing specifications of the chosen research works, we will describe the OoB component using four main criteria: its nature as stated in Subsection 2.2.1.2, its security classification as detailed in our adversary model 2.2.1.1 and the type of the required user intervention (*relay*, *compare*, *generate* or *setup*) that was first introduced in [66]. Furthermore, we will state the purpose behind the OoB data transmission (*Exchange* a parameter, *Verify* a value or *Validate* a specific event) since the security requirements on the Out-of-Band channel are entirely dependant on this information. For example, the use of a confidential channel is only required when the purpose is to exchange a security parameter such as a nonce which is the case for MANA III [71] and MVSec protocols [80].

Finally, we will provide a description framework that represents a summary of the existing security analysis conducted on SDP schemes. This framework will highlight the model used in the analysis: *symbolic* where we assume that the cryptographic primitives used are perfect and we focus entirely on the exchanges or *computational* where we evaluate the cryptographic aspects of the protocol. Also, we will describe the properties evaluated and the outcomes of the verification based on the tested scenarios in the original work. Furthermore, we will assess the results of the analysed security properties in order to highlight the discovered protocol vulnerabilities that will be, ultimately, used to propose the adequate mitigation. This description framework represents a complete and a systematic approach to describe the two components of the pairing protocol and a clear way of mapping the advantages and limitations of such schemes. The symbols, used in this description, are highlighted in Table 3.2.

3.1.1.2 MANnual Authentication II (MANA II)

3.1.1.2.1 Protocol steps:

This protocol, proposed by Gehrmann et al. [71], is described in Fig. 3.1 and it works as follows:

- ❶ ❷ The two devices, named Alice and Bob, exchange their Diffie-Hellman public keys g^a and g^b on the In-Band channel.
- ❸ ❹ The user initiates the authentication process on the device Alice after receiving a confirmation of the public key exchange. This action can be represented as a push button after receiving LED signals from the two objects.
- ❺ Alice computes a short secret K $(16-20$ bits) that is used to generate a short authentication string $sh_K(g^a||\widehat{g^b})$. $sh_K(.)$ represents a one-way function that takes as an argument a short key K and the concatenation of the DH public keys. Afterwards, she sends it to Bob on the In-Band channel.
- ❻ Alice and Bob display to the user their authentication values, K, $sh_K(g^a||\widehat{g^b})$ and K, $sh_K(\widehat{g^a}||g^a)$, using an output interface (e.g., screen).
- ❼ The user compares the strings displayed and confirms or rejects the pairing on both devices (e.g., by pressing a button in the case of a successful pairing attempt)

3.1.1.2.2 Out-of-Band specifications:

The MANA II protocol uses essentially a haptic OoB channel that rely on the physical intervention of the user to compare the displayed messages ❸ and ❻. The purpose of

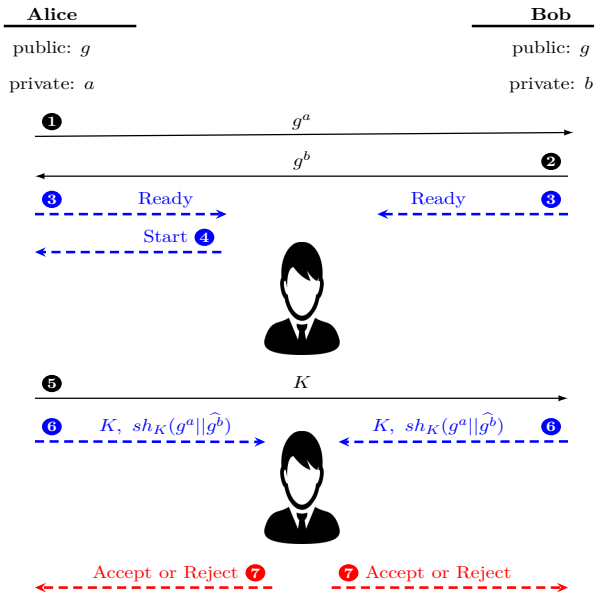


Figure 3.1: Alice and Bob diagram: MANA II protocol

these interactions is to *verify* the short authentication string that is constructed using both the key K and the short hash function $sh_K(g^a||g^b)$. In addition, the same channel is used to *validate* the pairing in message 7. The authors assume the use of an authentic channel that guarantees the data freshness, the integrity and the data origin authentication. However, the protocol structure only allows the use of a *delayable-authentic channel* since the adversary is able to perform a delay attack on the previous In-Band exchanges, as explained in Section 2.2.1.1, which violates the channel availability property.

3.1.1.2.3 Security analysis:

The protocol has been formally verified in [54, 43]. The results of the validation are shown in Table 3.1 and the evaluated security properties are described as follows:

1. **Paper:** Delaune et al. [54]
   - **Property description:**
     - *Non-injective agreement:* Whenever one of the devices finishes the protocol with the data d then the other device must have started the protocol with the same data.
   - **Assessment:** In the original work, the short hash is assumed to be breakable using collision attacks. However, the chosen properties hold over a single session and over two sessions. This is due to the fact that the short authentication key, K, and the hash of the public DH keys, $sh_K(g^a||g^b)$, are both shown to the user for comparison. This prevents any modification attack that targets any parameters used in the authentication. Therefore, the correctness of the user verification is the only weak link in the authentication process.
2. **Paper:** Chang and Shmatikov [43]
   - **Properties description:**
     (a) *Weak agreement:* If a device, Alice, successfully completes a protocol execution, apparently with another device Bob, then Bob has executed the protocol at least once and the two participants agreed on their identities.

(b) *Injective weak agreement:* If a device, Alice, successfully completes a protocol execution, apparently with another device Bob, then Alice has executed the protocol at least once and the two participants agreed on their identities. Additionally, each protocol run of Alice corresponds to a unique protocol run of Bob.

(c) *Non-injective agreement:* If a device, Alice, successfully completes a protocol execution, apparently with another device Bob, then Alice has executed the protocol at least once and the two participants agreed on all the parameters used to compute the challenge-response values.

(d) *Injective agreement:* If a device, Alice, successfully completes a protocol execution, apparently with another device Bob, then Alice has executed the protocol at least once and the two participants agreed on all the parameters used to compute the challenge-response values. Additionally, each protocol run of Alice corresponds to a unique protocol run of Bob.

- **Assessment:** Only the weak agreement property holds. This is due to the feasibility of launching multiple protocol executions without binding the session number to the authentication values showed to the user for comparison. This vulnerability leads the human verifier to approve on a pairing process that happened in a second session (tampered with by an attacker) based on the short authentication strings computed over the first session (without any attacker involvement). The protocol should associate a session identifier with the hash displayed to the user in order to mitigate the violations of the authentication properties. The contradiction between the results of the non-injective agreement property is explained by the feasibility of conducting a security verification over an unbounded number of session by ProVerif [31] which is not the case for the AKISS tool [41].

3.1.1.3 MANnual Authentication III (MANA III)

3.1.1.3.1 Protocol steps:

This protocol, proposed by Gehrmann et al. [71], is described in Fig. 3.2 and it works as follows:

- ❶ ❷ The two devices, named Alice and Bob, exchange their Diffie-Hellman public keys g^a and g^b on the In-Band channel.
- ❸ The user enters a four to six-digit random number R on both devices through their input interfaces (e.g., a keypad).
- ❹ Alice computes a long secret K_A that is used to generate an authentication string $h_{K_A}(g^a||\widehat{g^b}, R)$. $h_K(.)$ represents a keyed one-way hash function that takes as an argument a long key K, the concatenation of the DH public keys and a short nonce R. Afterwards, she sends it to Bob on the In-Band channel.
- ❺ Bob computes a long secret K_B that is used to generate an authentication string $h_{K_B}(\widehat{g^a}||g^b, R)$. $h_K(.)$ represents a keyed one-way hash function that takes as an argument a long key K, the concatenation of the DH public keys and a four to six-digit random number R. Afterwards, he sends it to Alice on the In-Band channel.
- ❻ ❼ Alice and Bob exchange the long keys, K_A and K_B, on the In-Band channel.
- ❽ Each device notifies the user of the verification outcome (e.g., using an LED signal)

- 9 The user confirms or rejects the pairing on both devices (e.g., by pressing a button in the case of a successful pairing attempt)

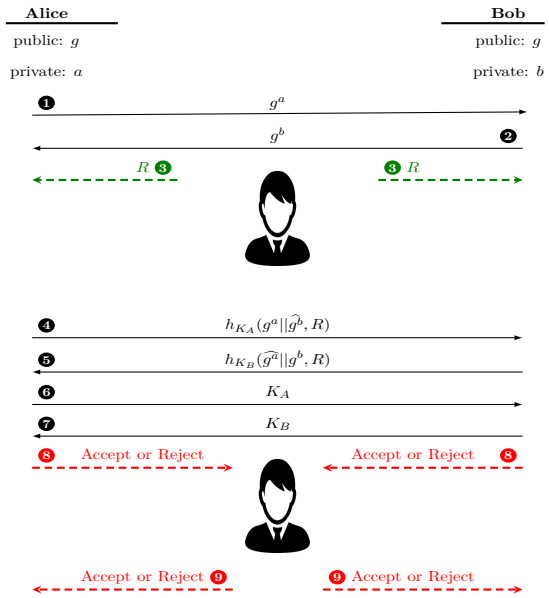


Figure 3.2: Alice and Bob diagram: MANA III protocol

3.1.1.3.2 Out-of-Band specifications:

The MANA III protocol uses two Out-of-Band channels that rely on the physical intervention of the user. The first one requires him to generate a random PIN R and to enter it on the two pairing devices. This channel is supposed to be out of the reach of the adversary which means that it should be classified as *confidential*. However, the second one only requires the data freshness, the integrity and the data origin authentication. Therefore, this channel is assumed to be classified as *authentic*. On the other hand, the protocol structure only allows the use of *delayable* channels since the adversary is able perform a delay attack on the previous In-Band exchanges, as explained in Section 2.2.1.1, which violates the channel availability property for both OoB communication links.

3.1.1.3.3 Security analysis:

The protocol has been formally verified as follows:

- **Paper:** Chang and Shmatikov [43]
- **Properties description:**
  1. *Key confidentiality:* At the end of a successful protocol execution between the two devices, the key is only known to Alice and Bob.
  2. *Non-injective agreement:* If a device, Alice, successfully completes a protocol execution, apparently with another device Bob, then Alice has executed the protocol at least once and the two participants agreed on all the parameters used to compute the challenge-response values.
- **Assessment:** The PIN's confidentiality is a key aspect to accomplish the authentication goal. However, the fact that we rely on the user to provide a random PIN represents a potential vulnerability in the protocol design. This is due to the human tendency to generate a memorable PIN which is easy to guess by the attacker.

Therefore, the formal verification of the key secrecy and the non-injective agreement properties does not hold when the PIN has a low entropy. The only solution to guarantee the correctness of the protocol is to use a random PIN that is hard to guess by the attacker. This solution is validated by the formal verification when using a high entropy PIN where both the confidentiality and the authentication goals are achieved.

3.1.1.4 MANnual Authentication IV (MANA IV) & Manual Authentication Diffie-Hellman (MA-DH)

3.1.1.4.1 Protocol steps:

This protocol MANA IV, proposed by Laur and Nyberg [111], is described in Fig. 3.3 and it works as follows:

- The two devices, Alice and Bob, generate respectively an l-bit key, k_A and k_B and their DH private key, a and b.
- ❶ Alice uses a commitment scheme to commit on the key k_A and sends the commitment and her DH public key g^a to Bob on the In-Band channel.
- ❷ Bob sends both his DH public key g^b and the authentication key k_B to Alice.
- ❸ Alice sends her open value d_A to Bob on the In-Band channel.
- ❹ Alice computes her Short Authentication String (SAS) $SAS_A = h_{k_A||\widehat{k_B}}(g^a||\widehat{g^b})$ and sends it to Bob on the Out-of-Band channel.
- ❺ Bob verifies the correctness of the SAS sent by Alice and notifies the user to confirm the pairing.

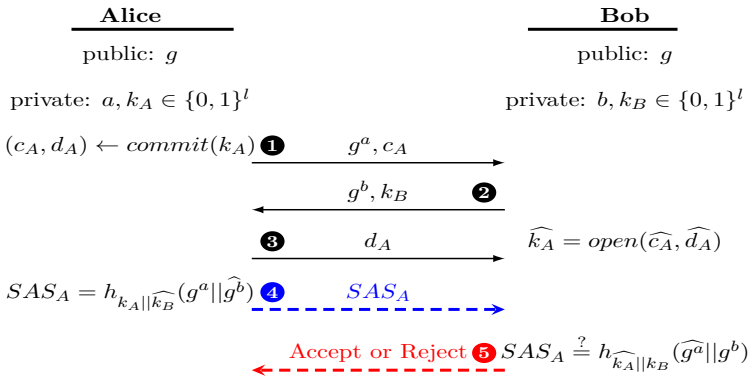


Figure 3.3: Alice and Bob diagram: MANA IV protocol

In the case of the MA-DH protocol, the authors are using the exchanged Diffie-Hellman public keys for the construction of the authentication string instead of generating the keys, k_A and k_B, to avoid the additional computations. The MA-DH protocol structure is described in Fig. 3.3 and it works as follows:

- The two devices, Alice and Bob, generate respectively a unique session identifier, ID_A and ID_B and their DH private key, a and b on the In-Band channel.
- ❶ Alice uses a commitment scheme to commit on her DH public key g^a and sends the commitment and her identifier to Bob on the In-Band channel.

- ❷ Bob sends both his DH public key g^b and his identifier to Alice on the In-Band channel.
- ❸ Alice sends her open value d_A to Bob on the In-Band channel.
- ❹ Alice computes her Short Authentication String (SAS) $SAS_A = h_{g^a||\widehat{g^b}}(ID_A||ID_B)$ and sends it to Bob on the Out-of-Band channel.
- ❺ Bob verifies the correctness of the SAS sent by Alice and notifies the user to confirm the pairing.

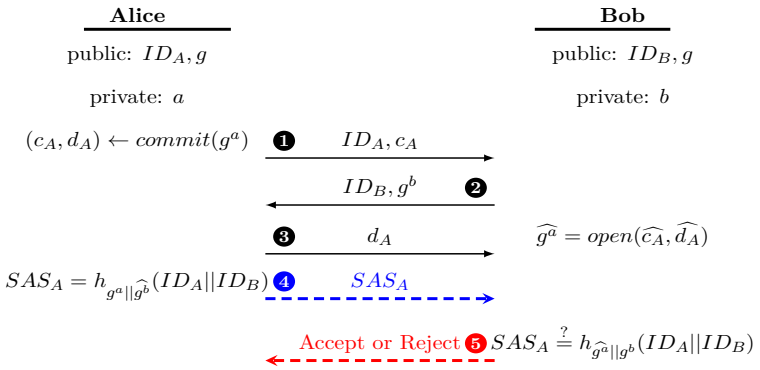


Figure 3.4: Alice and Bob diagram: MA-DH protocol

3.1.1.4.2 Out-of-Band specifications:

The MANA IV and the MA-DH protocols are based on the use of two Out-of-Band channels that have two main purposes: the verification of the authentication string and the validation of the pairing process. The former channel is required to guarantee the integrity and the data origin authentication without the need for the data freshness property. The security provided is questioned by our adversary model due to the tolerance policy toward replay attacks as detailed in Section 2.2.1.1. However, the latter channel is required to be classified as *authentic* which makes it hard for the adversary to transmit any messages on the Out-of-Band. Therefore, we can guarantee the correctness of the validation process. Finally, the structure of the protocols allows the attacker to perform a delay attack based on the previous In-Band exchanges which violates the channel availability property.

3.1.1.4.3 Security analysis:

The two protocols have been computationally verified as follows:

- **Paper:** Laur and Nyberg [111]
- **Verification terminology:** Subsection 2.1.4
- **Evaluated properties:**
  1. **Property:** *Upper-bound of the successful attack probability*
     - **Property description:** An adversary succeeds in deception if at the end of the protocol Alice and Bob reach the accepting state but $(g^a, \widehat{g^b}) \neq (\widehat{g^a}, g^a)$. As stated in [111], let A be the attacker algorithm. A protocol is considered $(t, \epsilon)-$secure if for any t-time attacker A, the attack success probability is formulated as follows:

$$Adv^{attack}(A) = \max_{g^a, g^b} \Pr[\text{Successful pairing } (g^a, \widehat{g^b}) \neq (\widehat{g^a}, g^a)] \leq \epsilon \quad (3.1)$$

  - **Tested scenarios:**
    * *Statistically binding commitment scheme:* For any t, there exists $\tau = t + \mathcal{O}(1)$ such that if the commit function $Commit(.)$ is $(\tau, \epsilon_1)-$hiding, ϵ_2-binding and $(\tau, \epsilon_3)-$non-malleable and the hash function $h(.)$ is (ϵ_a, ϵ_b)-almost regular and ϵ_u-almost universal then **the protocol is** $(2\epsilon_1 + 2\epsilon_2 + \epsilon_3 + \max\{\epsilon_a, \epsilon_b, \epsilon_u\})-$**secure**
    * *Computationally binding commitment scheme:* For any t, there exists $\tau = 2t + \mathcal{O}(1)$ such that if the commit function $Commit(.)$ is $(\tau, \epsilon_1)-$hiding, ϵ_2-binding and $(\tau, \epsilon_3)-$non-malleable and the hash function $h(.)$ is (ϵ_a, ϵ_b)-almost regular and ϵ_u-almost universal then **the protocol is** $(2\epsilon_1 + \epsilon_2 + \sqrt{\epsilon_2} + \epsilon_3 + \max\{\epsilon_a, \epsilon_b, \epsilon_u\})-$**secure**

- **Assessment:** The use of a statistically binding commitment scheme provides better security guarantees than the computational one as demonstrated by the upper bounds of the attack probabilities. Also, it is possible to choose a hash function that provides $\max\{\epsilon_a, \epsilon_b, \epsilon_u\} = 2^{-l}$ where l represents the number of bits sent over the Out-of-Band channel. Furthermore, it is possible to have a negligible ϵ_1, ϵ_2 and ϵ_3 with respect to the security parameter for a suitable choice of commitment scheme. Thus, MANA IV is considered, based on the definition provided by the original work, **asymptotically optimal** in term of security.

3.1.1.5 SAS-based Cross-Authentication Protocol

3.1.1.5.1 Protocol steps:

This protocol, proposed by Vaudenay [192], is described in Fig. 3.5 and it works as follows:

- The two devices, Alice and Bob, generate respectively a nonce, R_A and R_B, and their DH private key, a and b.
- ❶ Alice uses a commitment scheme to commit on her DH public key g^a and her nonce R_A. Then, she sends the commit value c_A and her public key to Bob on the In-Band channel.
- ❷ Bob uses a commitment scheme to commit on his DH public key g^b and his nonce R_B. Then, he sends the commit value c_B and his public key to Alice on the In-Band channel.
- ❸ Alice sends her open value d_A to Bob on the In-Band channel.
- ❹ Bob sends his open value d_B to Bob on the In-Band channel.
- ❺ Alice retrieves the values hidden in the commitment $\widehat{c_B}$ sent by Bob using the open value $\widehat{d_B}$. She verifies both the public key committed and the fact that the first bit is equal to one to avoid the attack where her message gets sent back to her. Then, she computes her Short Authentication String (SAS) $SAS_A = R_A \oplus \widehat{R_B}$ and sends it to Bob on the Out-of-Band channel.
- ❻ Bob verifies the correctness of the SAS sent by Alice and replies with his SAS as a confirmation of the pairing.

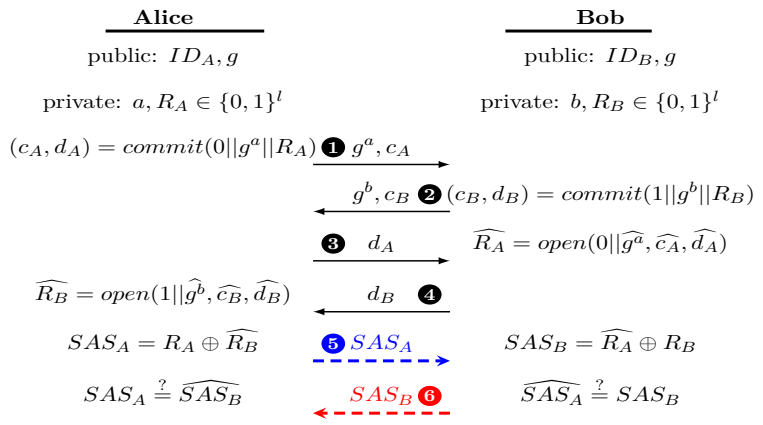


Figure 3.5: Alice and Bob diagram: SAS-based Cross-Authentication protocol

3.1.1.5.2 Out-of-Band specifications:

Similar to the MANA IV and MA-DH protocols, refer to Subsection 3.1.1.4, the SAS-based cross authentication scheme is based on the use of two Out-of-Band channels that have two main purposes: the verification of the authentication string and the validation of the pairing process. The two channels are required to guarantee the integrity and the data origin authentication without the need for the data freshness property. Therefore, the security provided is questioned by our refined adversary model due to the tolerance policy toward replay attacks as detailed in Section 2.2.1.1 which can compromise the security of the scheme in a practical scenario. Finally, the structure of the protocol allows the attacker to perform a delay attack based on the previous In-Band exchanges which violates the channel availability property.

3.1.1.5.3 Security analysis:

The protocol has been computationally verified as follows:

- **Paper:** Vaudenay [192]
- **Verification terminology:** Subsection 2.1.4
- **Evaluated properties:**

  1. **Property:** *Upper-bound of the successful attack probability*
     - **Property description:** An attack is considered successful if there exist an instance of the protocol, between Alice and Bob, which terminates by reaching an accepting state $(ID_A, \widehat{ID_B}, g^a, \widehat{g^b}) \neq (\widehat{ID_A}, ID_B, \widehat{g^a}, g^b)$.
     - **Tested scenarios:**
       * *One-shot attack:* Assuming that the commitment scheme is either $(t_c, \epsilon)-$extractable or $(t_c, \epsilon)-$equivocable, there exists a small constant μ (overall time complexity of the protocol) such that for any t-time adversary, $p_{one-shot} \leq 2^{-l} + \epsilon$ or $t \geq t_c - \mu$ where ϵ is negligible.
       * *Multi-session attack:* Assuming that Q_A (respectively Q_B) and μ_A (respectively μ_B) are the maximum number of sessions launched by Alice (respectively Bob) and the time complexity of the overall authentication protocol for each participant. For any t_0-time adversary, any Q_A and Q_B, the multi-session attack success probability $p_{multi-session}$ can be formulated using the t-time one-shot adversary scenario to have $p_{multi-session} \leq p_{one-shot} \times Q_A \times Q_B$ with a complexity $t \leq t_0 + \mu_A \times Q_A + \mu_B \times Q_B$.

- **Assessment:** The first tested scenario provides the upper bound of the one-shot attack success probability. This bound is dependant on the number of bits l transmitted on the authentication channel and the security parameter ϵ of the commitment scheme. Based on the second tested scenario, we can see that the upper bound of the success probability of a multi-session attack can be deduced based on the first result as follows $p_{multi-session} \leq p_{one-shot} \times Q_A \times Q_B$. For a negligible ϵ, the probability can be $Q_A \times Q_B \times 2^{-l}$.

3.1.1.6 Improved SAS-based Cross-Authentication Protocol

3.1.1.6.1 Protocol steps:

This protocol, proposed by Pasini and Vaudenay [149], is described in Fig. 3.6 and it works as follows:

- The two devices, Alice and Bob, generate respectively a hashing key, K_A and a nonce R_B. Then they generate their DH private key, a and b.
- ❶ Alice uses a commitment scheme to commit on her DH public key g^a and her hashing key K_A. Then, she sends the commit value c_A and her public key to Bob on the In-Band channel.
- ❷ Bob sends his public key g^b and his nonce R_B to Alice on the In-Band channel.
- ❸ Alice sends her open value d_A to Bob on the In-Band channel.
- ❹ Alice computes her Short Authentication String (SAS) $SAS_A = R_A \oplus h_{K_A}(\widehat{g^b})$ and sends it to Bob on the Out-of-Band channel.
- ❺ Bob retrieves the hashing key value from Alice's commitment. Then, he verifies the correctness of the received message on the Out-of-Band channel and replies with his SAS as a confirmation of the pairing.

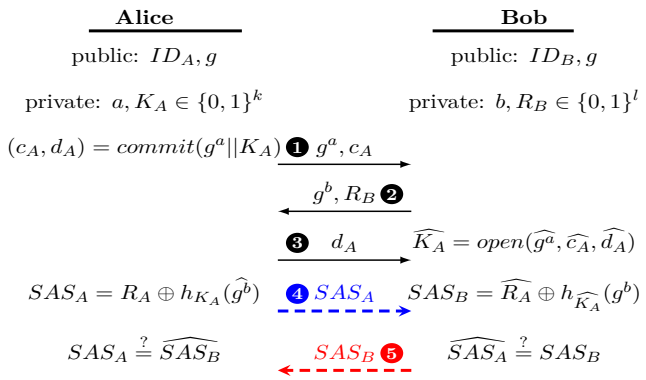


Figure 3.6: Alice and Bob diagram: Improved SAS-based Cross-Authentication protocol

3.1.1.6.2 Out-of-Band specifications:

Similar to the previous version of this protocol, this improvement is based on the use of two Out-of-Band channels that have two main purposes: the verification of the authentication string and the validation of the pairing process. The two channels are required to guarantee the integrity and the data origin authentication without the need for the data freshness property. Therefore, the security provided does not stand based on our refined adversary model due to the tolerance policy toward replay attacks as detailed in Section 2.2.1.1 which can compromise the security of the scheme in a practical deployment scenario. This tolerance can be further explained by giving the adversary the power to replay previous exchanges but not the ability to inject their own messages under the assumption that we have no pre-shared secret to construct a signature-based mechanism.

Finally, the structure of the protocol allows the attacker to perform a delay attack based on the previous In-Band exchanges which violates the channel availability property.

3.1.1.6.3 Security analysis:

The protocol has been computationally verified as follows:

- **Paper:** Pasini and Vaudenay [149]
- **Verification terminology:** Subsection 2.1.4
- **Evaluated properties:**
  1. **Property:** *Upper-bound of the successful attack probability*
     - **Property description:** An attack is considered successful if there exist an instance of the protocol, between Alice and Bob, which terminates by reaching an accepting state $(ID_A, \widehat{ID_B}, g^a, \widehat{g^b}) \neq (\widehat{ID_A}, ID_B, \widehat{g^a}, g^b)$.
     - **Tested scenario:**
       - *Multi-session attack:* Let $\epsilon = q^2 2^{-l_e} + q^2 2^{-l_c}$ where q is the maximum number of H function queries, l_e is the bit-length of the nonce e used in the random oracle commitment scheme and l_c is the bit-length of the commit value c. Let h be a strongly ϵ_u-almost universal hash function with a l-bit output. The success probability, against an adversary that can launch at maximum Q instances of Alice or Bob, is bounded by $\frac{Q(Q-1)}{2}(2^{-l} + \epsilon + \epsilon_u)$
- **Assessment:** The case of a one-shot sucess probability attack can be found when assuming $Q = 2$. Also, in the work of Laur and Nyberg [111], the extractability and the equivocability notions have been put into question. Furthermore, the use of the Bellare-Rogaway adversary model have been deemed complex and unsuitable for evaluating the security of authentication schemes that run statistically independent consecutive protocol executions (ad-hoc device pairing protocols).

3.1.1.7 Ephemeral Key Exchange Protocol

3.1.1.7.1 Protocol steps:

This protocol, proposed by Hoepman [84], is described in Fig. 3.7 and it works as follows:

- The two devices, Alice and Bob, generate respectively their DH private key, a and b.
- ❶ Alice commits on her DH public key g^a using a long hash function $h(.)$. Then, she sends the commit value $h(g^a)$ to Bob on the In-Band channel.

- ❷ Bob applies the same computation on his DH public key g^b. Then, he sends the commit value $h(g^b)$ to Alice on the In-Band channel.
- ❸ Alice sends a short hash of her public key $sh(g^a)$ to Bob on the Out-of-Band channel.
- ❹ Bob sends a short hash of his public key $sh(g^b)$ to Alice on the Out-of-Band channel.
- ❺ Alice sends the real value of her DH public key to Bob on the In-Band channel.
- ❻ Bob verifies the two hashes sent in ❶ and ❸ using the received public key of Alice. Then, he sends the real value of his DH public key on the In-Band channel.
- ❼ Alice verifies the two hashes sent in ❷ and ❹ using the received public key of Bob. Then, she sends a confirmation of the shared DH secret key $\widehat{g^b}^a$ using the long hash function on the In-Band channel.
- ❽ Bob verifies the key confirmation of Alice and confirms the pairing by sending the hash of his DH secret key $\widehat{g^a}^b$ on the In-Band channel.

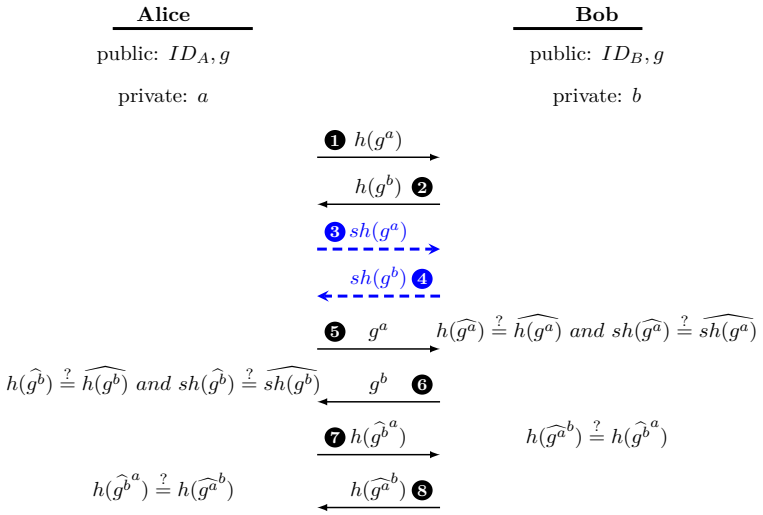


Figure 3.7: Alice and Bob diagram: Ephemeral key exchange protocol based on a bidirectional Out-of-Band channel

3.1.1.7.2 Out-of-Band specifications:

The protocol uses a bidirectional Out-of-Band channel to verify the short hash of the exchanged DH public keys. The channel is supposed to only guarantee the integrity and the origin authentication of the data. Thus, the protocol tolerates any replay attempts by the adversary which might violates the security provided by the scheme when applied to a realistic use-case as detailed in 2.2.1.1. Also, the channel availability property is not guaranteed based on the structure of the protocol.

3.1.1.7.3 Security analysis:

The protocol has been computationally verified as follows:

- **Paper:** Hoepman [84]

- **Verification terminology:** Subsection 2.1.4
- **Evaluated properties:**
  1. **Property:** *Upper-bound of the successful attack probability*
     - **Property description:** An attack is considered successful if there exist an instance of the protocol, between Alice and Bob, which terminates by reaching an accepting state $(g^a, \widehat{g^b}) \neq (\widehat{g^a}, g^b)$.
     - **Tested scenario:**
       * *Multi-session attack:* Let l be the bit-length of the short hash. Let Q be the maximum number of sessions that can be initiated by the adversary. The successful attack probability is bounded by $1 - e^{-\frac{Q}{2^l}} + 2^{-|g^a|}$.
- **Assessment:** The success probability bound has two parts. The first one describes the advantage of an active adversary searching for a collision between the two hashes to bypass the verification. The second part describes the advantage of a passive attacker that tries to guess an $|g^a|$-bit DH secret key based on the exchanged public keys. The $2 \times l$-bit bidirectional exchanges on the Out-of-Band channel affect the optimality of the scheme in terms of communication cost since it only provides an attack success probability bound close to $q \times 2^{-t}$. This aspect has been improved in the work of Laur and Nyberg [111] where they reduced the number of OoB exchanges by using a single unidirectional channel that only carries a t-bit authentication string. This improved scheme provides the same level of security by using a single OoB transmission.

3.1.1.8 Wong-Stajano Asymmetric Pairing Protocol

3.1.1.8.1 Protocol steps:

This protocol, proposed by Wong and Stajano [201], is described in Fig. 3.8 and it works as follows:

- The two devices generate respectively their DH private key, a and b.Then, Bob generates a short nonce R_B and long hashing key K_B.
- ❶ Alice sends her identifier ID_A and her DH public key g^a to Bob on the In-Band channel.
- ❷ Bob computes the keyed hash $h_{K_B}(ID_B, R_B, g^b, \widehat{g^a})$. Then, he sends it along with his identifier and his DH public key g^b to Alice on the In-Band channel.
- ❸ Alice replies by an acknowledgement *Ack* on the Out-of-Band channel to confirm the reception of the message ❷.
- ❹ Bob sends the short nonce R_B to Alice on the Out-of-Band channel.
- ❺ Bob sends the value of the hashing key K_B to Alice on the In-Band channel.
- ❻ Alice verifies the hash sent in ❷ using the hashing key and the public key of Bob. Then, she confirms or rejects the pairing on the Out-of-Band channel.

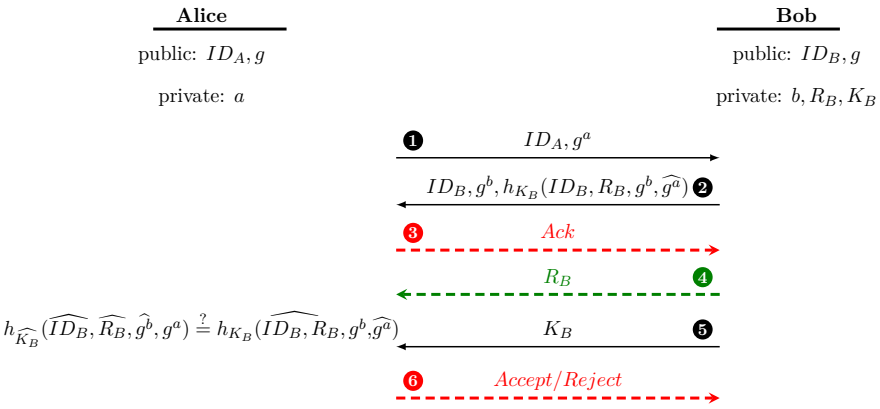


Figure 3.8: Alice and Bob diagram: Asymmetric pairing protocol based on a unidirectional Out-of-Band channel

3.1.1.8.2 Out-of-Band specifications:

This protocol is based on three Out-of-band transmissions that have two main purposes: the validation of a specific event and the exchange of a parameter related to the authentication process. The two OoB transmissions, 3 and 6, require the physical intervention of the user to validate the reception of the message 2 by relaying a one-bit interaction to the other device. Thus, these Out-of-Band channels can be considered haptic, as described in Section 2.2.1.2.6, which classifies them as *protected* by guaranteeing the integrity, the data origin authenticity, the data freshness and the liveness properties. As for the Out-of-band transmission in message 4, the protocol uses a visible light communication that is classified as *authentic* by providing the integrity, data origin authenticity and data freshness. Based on the usability analysis conducted in section 2.2.1.2.4, the vigilant user is required to setup the devices in a way to create a direct Light of Sight (LoS). Finally, the protocol structure allows the attacker to delay messages on the Out-of-Band channel by apply this action on the previous In-Band exchanges which violates the channel availability property. Therefore, the channels used in this scheme are considered *delayable*.

3.1.1.8.3 Security analysis:

The protocol has been formally verified as follows:

- **Paper:** Nguyen and Leneutre [144]
- **Evaluated properties:**
  1. **Property:** Non-injective agreement [120]
     - **Property description:** The initiator Alice completes a run of the protocol, apparently with Bob, then Bob has previously executed the protocol as a responder, apparently with Alice, and the two parties agreed at the end of the protocol execution on the same DH secret key.
- **Assessment:** The formal analysis has yielded two multi-session attacks that violate the agreement property. These vulnerabilities are based on the delay capability of an attacker over the Out-of-Band channel and the feasibility of a replay attack that is allowed by the security model of the protocol. This scheme has been improved in the work of Nguyen and Roscoe [142] by eliminating the acknowledgment message which reduces the user intervention. Furthermore, they improved the protocol design by

removing the use of two successive unidirectional messages that eliminates the vulnerability noticed by Nguyen and Leneutre [144] later on. From the computational aspect, the new version uses two short nonces and discards the use of a long hashing key which makes it more convenient for the resource-constrained devices.

3.1.1.9 2-round Authenticated Key Agreement Protocol

3.1.1.9.1 Protocol steps:

This protocol, proposed by Nguyen and Leneutre [145], is described in Fig. 3.9 and it works as follows:

- The two devices, Alice and Bob, generate respectively their DH private keys, a and b, and their nonces, r_a and r_b.
- ❶ Alice sends her DH public key g^a and the hash value $h(g^a, r_a)$ to Bob on the In-Band channel.
- ❷ Bob sends his DH public key g^b and his nonce r_b to Alice on the In-Band channel.
- ❸ Alice computes the value $r_a \oplus h_{r_b}(g^a, g^b)$ and transfers it to Bob on the Out-of-Band channel.
- ❹ Bob retrieves the value of r_a from the message ❸, verifies the hash sent in message ❶ and confirms or rejects the pairing on the Out-of-Band channel.

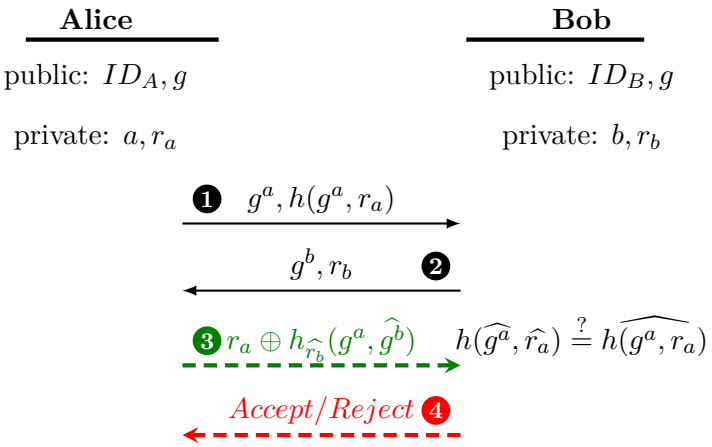


Figure 3.9: Alice and Bob diagram: 2-round authenticated key agreement protocol with unidirectional Out-of-Band channel

3.1.1.9.2 Out-of-Band specifications:

This protocol is based on two Out-of-Band channels that, respectively, serve the purpose of exchanging a security parameter related to the authentication process and the purpose of validating the pairing. The first channel is supposed to guarantee the integrity and the data origin authenticity without the need for the data freshness property. Thus, the attacker is able to perform a replay on the OoB channel which, according to our security model 2.2.1.1, might lead to compromising the security of the scheme when deployed in a realistic use-case. The second OoB channel requires the physical intervention of the human operator to relay a one-bit interaction to validate the pairing on the other device. Thus, this haptic channel is classified as *protected* since it guarantees, in addition to the first one, the data freshness and the liveness security properties. Finally, the protocol

structure allows the attacker to delay messages on the Out-of-Band channel by apply this action on the previous In-Band exchanges which violates the channel availability property. Therefore, the channels used in this scheme are considered *delayable*.

3.1.1.9.3 Security analysis:

The protocol has been formally verified as follows:

- **Paper:** Nguyen and Leneutre [145]
- **Evaluated properties:**
  1. **Property:** Non-injective agreement [120]
     - **Property description:** The initiator Alice completes a run of the protocol, apparently with Bob, then Bob has previously executed the protocol as a responder, apparently with Alice, and the two parties agreed at the end of the protocol execution on the same DH secret key.
- **Assessment:** Based on the manual formal analysis conducted by the authors, the scheme achieves the non-injective agreement property while minimising the communication costs in terms of number of messages on the In-Band and the Out-of-Band channel. Furthermore, the authors reduced the number of cryptographic primitives to two hash functions without the need to generate another key for hashing in order to comply with the limitations of the resource-constraint devices.

Table 3.1: Summary of the security proofs

Protocol	Security analysis	Security analysis model	Security analysis tool	Properties	Tested scenarios	Results
MANA II [71]	Delaune et al. [54]	Symbolic	AKISS [41]	Non-injective agreement	Alice to Bob (single session)	Verified
					Bob to Alice (single session)	Verified
					Alice to Bob (two sessions)	Verified
					Bob to Alice (two sessions)	Verified
	Chang and Shmatikov [43]	Symbolic (Dolev-Yao [59])	ProVerif [31]	Weak agreement	Alice to Bob	Verified
					Bob to Alice	Verified
				Injective weak agreement	Alice to Bob	Failed
					Bob to Alice	Failed
				Non-injective agreement	Alice to Bob	Failed
					Bob to Alice	Failed
				Injective agreement	Alice to Bob	Failed
					Bob to Alice	Failed
MANA III [71]	Chang and Shmatikov [43]	Symbolic (Dolev-Yao [59])	ProVerif [31]	Key confidentiality	Low entropy PIN	Failed
					Random PIN	Verified
				Non-injective agreement	Low entropy PIN	Failed
					Random PIN	Verified
MANA IV [111] & **MA-DH**[111]	Laur and Nyberg [111]	Computational	Manual	Upper-bound of the successful attack probability	Statistically binding commitment scheme	$2^{-l} + 2\epsilon_1 + 2\epsilon_2 + \epsilon_3$
					Computationally binding commitment scheme	$2^{-l} + 2\epsilon_1 + \epsilon_2 + \sqrt{\epsilon_2} + \epsilon_3$
SAS-based cross-authentication [192]	Vaudenay [192]	Computational (Bellare-Rogaway [26, 27])	Manual	Upper-bound of the successful attack probability	One-shot attack	$2^{-l} + \epsilon$
					Multi-session attack	$Q_A \times Q_B \times (2^{-l} + \epsilon)$
Improved SAS-based cross-authentication [149]	Pasini and Vaudenay [149]	Computational (Bellare-Rogaway [26, 27])	Manual	Upper-bound of the successful attack probability	Multi-session attack	$\frac{Q(Q-1)}{2}(2^{-l} + \epsilon + \epsilon_a)$
Ephemeral pairing [84]	Hoepman [84]	Computational (Bellare-Rogaway [26, 27])	Manual	Upper-bound of the successful attack probability	Multi-session attack	$1 - e^{-\frac{Q}{2^p}} + 2^{-\lvert g^a \rvert}$
Wong-Stajano asymmetric pairing protocol [201]	Nguyen and Leneutre [144]	Symbolic (Strand Spaces model [63])	Manual	Non-injective agreement	Alice to Bob	Failed
					Bob to Alice	Failed
2-round authenticated key agreement protocol [145]	Nguyen and Leneutre [145]	Symbolic (Strand Spaces model [63])	Manual	Non-injective agreement	Alice to Bob	Verified
					Bob to Alice	Verified

3.1.1.10 Summary

In this subsection, we summarize the highlighted results shown in Table 3.1. The MANA II protocol [71] has been formally verified in [43, 54] using two automated verification tools: ProVerif [31] and AKISS [41]. The work of Delaune et al. [54] focused on evaluating the non-injective agreement property, described in Subsection 2.1.5, under

the assumption of having at maximum two protocol sessions. This property holds since the key confirmation step is based on the correctness of a comparison conducted by the user on a short hash displayed by both devices. Thus, any human factor error related to a rush behaviour or a one digit mismatch might compromise the security of the pairing process as detailed in the work of Fomichev et al. [66]. However, a similar formulation of this property has been verified in the work of Chang and Shmatikov [43] based on an unbounded number of sessions. This property does not hold because of the feasibility of launching multiple protocol runs without binding the session number to the short authentication string. Therefore, it is feasible that the user approves a suitable but erroneous authentication value that belongs to a previous session. In addition, three other similar formulations of the properties, described in Subsection 2.1.5, have been evaluated: weak agreement, injective weak agreement and injective agreement. Only the first one holds since it provides the weakest definition of authentication by guaranteeing the agreement on the identities of the two intended devices that is assured by their participation in the pairing process. The same work has addressed the confidentiality aspect and the non-injective agreement of the MANA III protocol [71] based on the assessment of the entropy residing in the PIN that is entered by the user. These results of the verification reflect the importance of having such randomness in the PIN input which is not always the case due to the human tendency to provide memorable four to six digit values. On the other hand, the Wong-Stajano asymmetric pairing protocol [201] does not guarantee the non-injective agreement that have been formally evaluated, in the work of Nguyen and Leneutre [144], based on the Strand Spaces model [63]. This is due to a vulnerability in the protocol structure against a multi-session attack that exploits the use of two successive unidirectional exchanges which has been corrected in the design proposed in the work of Nguyen and Roscoe [142]. A lightweight pairing scheme has been introduced in another work of Nguyen and Leneutre [145] that achieves formally the previously discussed authentication property using only 4 exchanges. However, this construction is not robust computationally due to the feasibility of a brute-fore attack that aims at extracting the nonce value from the exchanged hash.

From the computational point of view, the upper bound of the attack success probability of four device pairing schemes has been evaluated. Two variants of the MANA suite protocols, MANA IV and MA-DH [111], have been verified under the assumption of using two different cryptographic primitives: a statistically and a computationally binding commitment schemes. Obviously, the use of the former primitive enhances the security since it reduces the probability bound but using both constructions, these protocol are asymptotically optimal in term of security with respect to the number of authentication bits exchanged over the Out-of-Band channel. The success probability of a multi-session attack on the two Short-Authentication-String (SAS) pairing protocols, proposed in [192, 149], has been evaluated under the Bellare-Rogaway model [26, 27]. Nonetheless, in the work of Laur and Nyberg [111], the extractability and the equivocability notions, described in Subsection 2.1.4, have been questioned along with the use of the Bellare-Rogaway adversary model since it is infeasible to run statistically independent consecutive protocol executions. Finally, the security analysis of the ephemeral pairing scheme, proposed in the work of Hoepman [84], has two outcomes. It describes the advantage of an active adversary searching for a collision between the two hashes to bypass the verification. The second part describes the advantage of a passive attacker that tries to guess an $|g^a|$-bit DH secret key based on the exchanged public keys that is usually neglected by the other computational evaluations. On the other hand, the $2 \times l$-bit bidirectional exchanges on the Out-of-Band channel affect the optimality of the scheme in terms of communication cost since it only provides an attack success probability bound close to $q \times 2^{-t}$ which has been improved in the work of Laur and Nyberg [111] where they reduced the number of

OoB exchanges by using a single unidirectional channel.

3.1.2 Security Analysis Under the Invasive Threat Model

3.1.2.1 Identity Misbinding Attack

The identity misbinding attack, also known as *unknown-key-share* attack, was first identified on the Station-to-Station protocol (STS) [57] in the work of Blake-Wilson et al.[29] in 1999. To simplify the attack's applicability on secure device pairing schemes, brought to light in the work of Sethi et al.[177], we will refer to three objects: the legitimate participants Alice and Bob, and the malicious actor Eve. For this attack to work, first, we need to assume that one of the legitimate devices is compromised in a way that lets the attacker control its input and output interfaces. This assumption might be quite strong but it is feasible to introduce a malicious object without being detected especially under the SDP hypothesis of not having any pre-shared information between the devices. Secondly, for the attack to work, we need to assume that the identity of the device is determined by the user's physical access to the object such as setting the discovery name on a Bluetooth-enabled device. This assumption is almost always validated since it is the case on the Bluetooth technology that is widely used by the IoT devices.

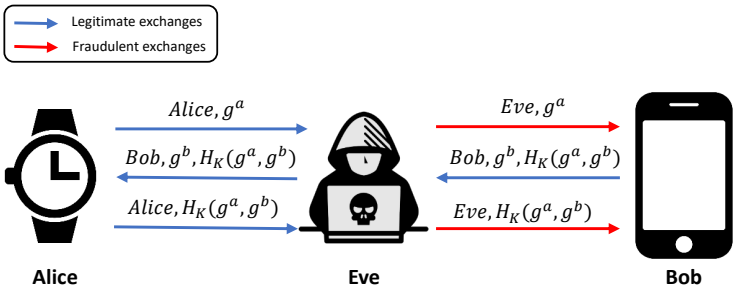


Figure 3.10: Misbinding attack against a Diffie-Hellman key exchange

In Fig. 3.10, we show a misbinding attack during a simple Diffie-Hellman key exchange protocol. Alice initiates the exchange by sending her identifier, represented by her name, and the DH public key g^a. Eve, our Dolev-Yao intruder, will block the transmission and induce her identifier instead of Alice's. Bob receives the message, identifies the existence of the other device which is Eve, binds her public key to her identifier, computes the secret session key $K = (g^a)^b = g^{ab}$, computes the keyed hash $H_K(g^a, g^b)$ and finally, sends the message $Bob, g^b, H_K(g^a, g^b)$ to Eve. The attacker replays the same message to Alice that will reply with her own keyed hash to confirm to Bob that she has the same key which wasn't tampered with. This attack results in a mismatching in the key authentication belief: Alice thinks that she has established a key exchange with Bob, which is technically true, and Bob thinks that he has established a key with a legitimate device that is Eve while hiding completely the existence of Alice. On the other hand, the key confidentiality is not compromised but the key authentication property has been violated.

The presence of an Out-of-Band channel can solve the issue when the devices, that are performing the pairing, are not compromised. This is due to the demonstrative identification and data origin authentication properties ensured by the pre-authenticated channel. However, the device's physical integrity is not always granted. Therefore, the risk still needs to be considered for high security level scenarios. At this moment, the SDP assumption of having two unidentified devices without any pre-shared knowledge, completely discards the possibility of having any secure binding between the ephemeral session key and the physical objects. Thus, the protocol is vulnerable to any misbinding attempts.

This attack can be more severe when applied against the device pairing schemes. It will not only compromise the key authentication between Alice, the legitimate sound initiator, and Bob, the legitimate compromised device, but also, it can lead to pairing Eve with Alice and to neglecting the existence of Bob. This attack is a combination between the unknown-key-share, the human error exploitation and the relay attack. In this case, we lure the user to pair Alice with Eve while thinking it is Bob. The attack steps can be detailed as follows:

1. Eve uses the same identifier as Bob to maximise the chances of luring the user to initiate the pairing with Eve instead of Bob.
2. Alice performs a DH key exchange with Eve.
3. Eve computes the Short Authentication String (SAS) and sends it to Alice through the Out-of-Band channel output interface of Bob.
4. Alice receives the SAS and confirms the pairing to the user.

At this stage the user thinks that Alice and Bob are securely paired while, in fact, he performed the pairing with a malicious object. Therefore, the attacker has succeeded in breaking both the key confidentiality and the key authentication assumptions without the possibility of detecting it.

3.1.2.2 Case Study: Bluetooth Secure Simple Pairing Protocol

This attack has been demonstrated on the *Numerical Comparison* variant of the Bluetooth Secure Simple Pairing protocol [32], as shown in Fig. 3.11. The devices, in the figure, were called differently: the user, Alice, Bob and Eve are respectively referred to as Alice, A, C and Mallory.

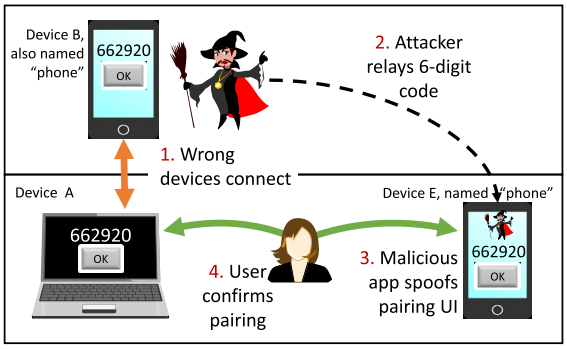


Figure 3.11: Misbinding attack against Bluetooth SSP numeric comparison[151]

The attack on the SSP protocol can occur as follows:

1. The user makes Bob discoverable and starts discovering the neighbouring objects enabling Bluetooth.
2. Eve copies the Bluetooth identifier of Bob then makes it non-discoverable.
3. The user chooses Eve on the list of discoverable devices thinking it was Bob.
4. Alice and Eve perform the exchanges of the necessary parameters (DH public keys, nonces, commitments...).
5. Eve computes the authentication PIN (six digit verification code) and commands Bob to display it to the user.

6. Alice computes the authentication PIN and displays it to the user.
7. The user verifies the match between the two PINs displayed on Alice and Bob.
8. The user confirms the pairing between Alice and Bob when, in fact, Alice and Eve are paired.

The hardest part of the attack, on the SSP protocol, is the feasibility to control the device discovery name by the user. This is due to the necessity of luring the user to initiate the pairing with Eve instead of Bob. This attack can also be conducted on the other two variants of SSP, *PIN Entry* and *Out-of-Band* (using the NFC technology), while excluding the variant *Just works* since it is not intended for security purposes.

3.1.2.3 Case Study: Extensible Authentication Protocol-Nimble Out-of-Band (EAP-NOOB)

This attack can be also applicable to a security bootstrapping protocol under the same assumptions that one participating node is compromised and that the devices identities are defined by the user physical access to them. As an example, the authors of [177] demonstrated this attack on the bootstrapping scheme Extensible Authentication Protocol-Nimble Out-of-Band (EAP-NOOB) [19] that pairs and registers the IoT devices to an online server. This scheme is an authentication method for the Extensible Authentication Protocol [4] that includes an Out-of-Band channel verification, which requires a degree of user involvement. EAP-NOOB targets the problem of pairing devices without any pre-shared knowledge and it offers a variety of OoB channels that transfer the authentication string using a QR code, an NFC transmission or an acoustic exchange. The protocol consists of four main phases:

- **In-Band key exchange:** The IoT object performs an ECDH key exchange with the server.
- **Object selection:** The user selects the IoT object from a list, provided by the server, on his personal device.
- **Out-of-Band key authentication:** The server sends, on the chosen Out-of-Band channel, the authentication/identification string that authenticates the key exchange and specifically informs the device of its user.
- **In-Band registration:** Completes the registration of the device to the user's account on the In-Band channel.

The misbinding attack, in this case, aims at registering a malicious device, called Eve, to the user's account instead of the legitimate but compromised one, referred to as Bob. Following the same example as the one introduced in the original article, Bob will be an object that only has an input interface such as a surveillance camera. The suited Out-of-Band channel, in this case, is the use of a QR code displayed on the user's personal device (e.g., smartphone). The attack steps occur as follows:

1. The user initiates the pairing by switching on the object Bob.
2. Bob performs an ECDH key exchange with the server
3. The attacker copies Bob's metadata to Eve and initiates the pairing with the server
4. The user looks for Bob in a list of the potential devices to be paired that is provided by the server

5. The user selects Eve instead of Bob
6. The user receives a QR code from the server and shows it to Bob
7. Bob sends the QR code to the attacker
8. The attacker shows the QR code to Eve
9. Eve continues the authentication and the registration process instead of Bob

The hardest part of the attack is luring the user to wrongfully select Eve instead of Bob in the second phase of the protocol. Due to this inattentive user behavior, the registration of a malicious device can occur without being noticed using a compromised relay device.

3.1.2.4 Mitigation

The misbinding attack can be mitigated by cryptographically binding the devices identifiers to the protocol session. Unfortunately, this solutions is not possible for the secure device pairing schemes since the objects don't share any prior information, including preshared symmetric keys or certificates. Another potential solution is the use of co-presence verification techniques that are based on variables from the ambient environment. However, numerous of these methods have been proven vulnerable against active attacks in the work of Shrestha et al.[181] which does not provide us with a complete solution but it only makes the attack's execution harder on the adversary. Therefore, the mitigation against this attack in the device pairing context is still an open discussion. However, it can be detected when we apply an secure enrollment protocol which reveals to the user that the paired device isn't the one intended to. Thus, the bootstrapping initiatives, presented in Subsection 2.1.6, that only cover the ad-hoc SDP phase are vulnerable against this particular attack.

3.2 Contribution N°1: Hybrid Secure Device Pairing Protocol

In this section, we study the combination of the two previously described types of secure device pairing protocols that are detailed in Section 2.2. Our new technique benefits from the fast contextual secret agreement in the context-based schemes to reduce the pairing completion time in comparison with the protocols relying solely on the low data-rate Out-of-Band channels. Also, we exploit the advantages of the Out-of-Band channels in terms of security under a threat model which deals with an ambient environment controlled by the attacker. Such strong intruder represents the Achilles' heel of any existing contextual scheme, especially without the requirement of human interactions such as performing some pattern of movement or taping, as suggested in [92]. Thus, we aim to enhance the security against a sophisticated contextual attacker without an extensive user involvement, which is not supported by the state-of-the-art contextual schemes. Our proposal has been designed based on the findings of the previous section. In addition, the security of the hybrid protocol will be formally validated based on the strongest notion of the agreement properties according to the work of Lowe [120].

3.2.1 System Overview

Our hybrid protocol, called Contextual Out-Of-Band Protocol (COOB) [104], has two distinct components. The first one is a contextual module where we take advantage of any fast and reliable contextual key agreement technique. The second component

is a Delayable-Authentic OoB channel that guarantees at least the authenticity and the integrity of the exchanged information. This design provides a security improvement in comparison with the existing context-based schemes since it is robust against a powerful contextual attacker. This adversary can sense and even control the ambient environment surrounding the two legitimate devices. Furthermore, it provides a usability improvement by reducing the protocol completion time in comparison with the existing pairing schemes that rely solely on a low data-rate OoB channel. In addition, COOB maintains a reduced cryptographic cost of only two hash computations for each device. In order to reach this level of optimality, a nonce exponentiation is exploited while constructing the Diffie-Hellman public keys [56] to temporarily hide their real values.

3.2.2 Assumptions and Threat Models

We take into account the scenario where two devices, Alice and Bob, try to pair by authenticating their public Diffie-Hellman keys exchanged over the In-Band channel under the non-invasive threat model that is described in Subsection 2.1.2.2. We assume that the *discovery* phase, where the two devices gain knowledge of each other, has been correctly established by the user. The target devices of our protocol need, based on the choice of the contextual part, a Bluetooth module to communicate on the In-Band channel and a Wi-Fi chipset able to extract the CSI measurements. Also, we need, based on the choice of the Out-of-Band channel, a LED and a button as interfaces on the initiator device, named Alice, a LED and a light-sensor as interfaces on the enrollee, named Bob. Additionally, we need enough computational power to handle the Diffie-Hellman key computations [56].

We assume the existence of a powerful Dolev-Yao adversary that is able to control both the In-Band channel and the ambient environment surrounding the pairing participants such as the audio, the radio (Wi-Fi, Bluetooth and GPS) and even the physical environment (temperature, humidity, altitude and their combinations). This capability is not limited to a single target device since we assume that the attacker can be in the same context as all of the legitimate objects for an unlimited period of time. Furthermore, in our analysis, we consider the feasibility of computational attacks that are targeting the cryptographic functions that rely solely on a short secret as the source of randomness. This assumption makes the security evaluation of our scheme more realistic with respect to the use of short secrets to perform the ad-hoc pairing. Therefore, we assume the existence of two types of attackers against the contextual module: the first one is an **Ordinary Contextual Intruder** that is not able to suppress any existing contextual information and is not allowed inside a pre-defined *authentication zone* fixed by the pairing scheme assumptions. This adversary is considered in order to investigate the security of our protocol COOB against the commonly adopted attacker model by the state-of-the-art context-based schemes. The second one is a **Sophisticated Contextual Intruder** that is able to sense and ultimately control the ambient environment, which makes him aware of the secret extraction outcome in both devices. The latter threat model might seem unrealistic but it has been proven in [181] that such attacks, against co-presence authentication systems, are possible using a form of "ghost-and-leech" technique [99]. Due to the close proximity of the pairing parties, the adversary might use a leech and a ghost at the same place. The leech plays the role of an eavesdropping device that senses the environment and sends it back to the attacker using a fast digital communication, i.e a microphone or a photo-sensor. On the other hand, the ghost plays the role of a device that controls the environment, e.g a speaker or a laser.

3.2.3 COOB Pairing Protocol

Our proposal is split into two main steps. First, we briefly introduce, in the background section 3.2.3.1, the contextual module where we highlight the key aspects of the context-based protocol TDS [204] that is applied in our proposal. Then, we explain our choice of the visible light communication as our Out-of-Band channel. Secondly, we present the exchanges of our protocol, COOB, that combines the two previously mentioned blocks in an optimal manner in terms of time, communication and computational efficiency by exploiting the advantages of a nonce exponentiation technique. The applied terminology to describe the protocol is detailed in Table 3.2.

Table 3.2: Notations

Notation	Definition
mod	Modulus operation
ID_X	Identifier of the device X (e.g., MAC Address)
$\oplus$	Exclusive OR operation
$sh(.)$	Short hash function
$sh_K(.)$	Keyed short hash function using the key K
$h(.)$	Long hash function
$h_K(.)$	Keyed long hash function using the key K
$\|X\|$	Number of bits of X
$\hat{x}$	Received value that can be modified by the adversary
$x\|\|y$	Concatenation of the two values x and y
x'	A value induced by the adversary
$[X]_i^j$	A truncation of the binary value X starting from the bit in position i to position j
$\times$	Multiplication operator
$(x \times y)$-Matrix	Matrix with x rows and y columns
$\longrightarrow$	In-Band channel
$\dashrightarrow$	Exchange Out-of-Band channel
$\dashrightarrow$	Verification Out-of-Band channel
$\dashrightarrow$	Validation Out-of-Band channel
Q_X	The maximum number of sessions launched by the participant X
Q	The maximum number of sessions launched by any participant

3.2.3.1 Background

3.2.3.1.1 Contextual Module

In our design, we apply the fuzzy extractor used in the work of Xi et al. [204] that exploits the channel state readings from a Wi-Fi access point that is publicly agreed upon. Due to the close proximity of the two legitimate devices (within an authentication zone $0.4\lambda \approx 5$ cm using the 2.4GHz frequency), they receive highly correlated CSI amplitude measurements as highlighted in Figure 3.12. The sensing of the ambient environment is initiated by each device at the beginning of the discovery phase.

After gathering a sufficient number of samples, Alice tries to synchronize the sampled data with the other device by sending a sequence of values to Bob marking the beginning of the valid samples that are used in the encoding process. The S-box in our case represents a $(2 \times l)$-matrix where l is the bit-length of the secret. Each element of the matrix includes a number $m \times n$ of CSI samples that uniquely represent a bit value 0 or 1, where m is the number of sub-carriers used and n is the number of measurements per sub-carrier. Thus, two consecutive $m \times n$ samples need to be distinct in order to reflect a 0 or a 1

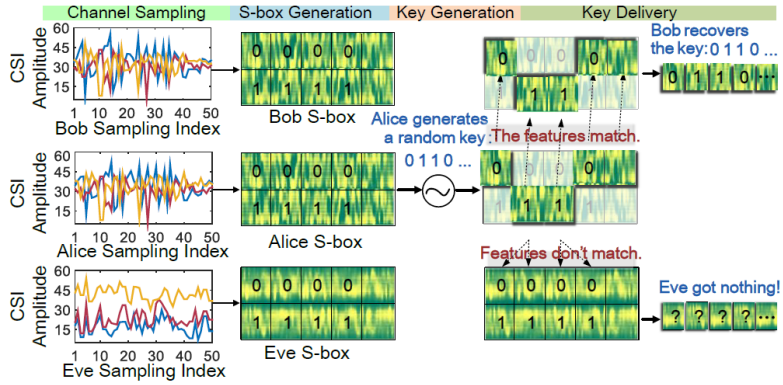


Figure 3.12: The main steps of TDS [204]

bit. After uniquely identifying each block of the matrix, an l-bit secret is independently generated by Bob and then, for each bit, he sends its corresponding block in the S-box. As an example, if the secret starts with the sequence 0110 then Bob sends the first 0-block, the second 1-block, the third 1-block and the fourth 0-block as illustrated in Figure 3.12. Since Alice has computed a similar S-box due to the reception of similar CSI samples, she decides whether the received i^{th} block represents a 0 or a 1 bit value based on a comparison with her i^{th} column in her matrix. However, in the original work, the adversary is not able to reconstruct the original message due to his different measurements, which result in a different matching S-box. In this design, we use Reed-Solomon (RS) codes to ensure that Alice can correct a number of bits fewer than a fixed limit. This guarantees the reconstruction of the secret by only a legitimate device inside the *authentication zone*. Readers willing to learn more about the TDS scheme can consult the original paper [204].

To simplify the protocol description in the upcoming sections, we model this technique as a fuzzy-commitment scheme [93] that uses two similar contextual bit-values r_{c_a} and r_{c_b} generated respectively by Alice and Bob. These two variables represent the S-box process of encoding and decoding based on the CSI features. The transfer of the blocks V_b by Bob is modeled as $V_b = r_{c_b} \oplus Encode(r_b)$ where $Encode(.)$ is the Reed-Solomon encoding function. This message is decoded on the other side using r_{c_a} as follows: $r_b = Decode(r_{c_a} \oplus V_b)$ where $Decode(.)$ is the Reed-Solomon decoding function. The feasibility of this modeling is due to the similarity between the concept of representing a bit by multiple random information and the idea of hiding its value using a random contextual bit and an XOR operation.

3.2.3.1.2 Out-of-Band Module

In our proposal, we need two Out-of-Band channels that limit the attacker capabilities to blocking, delaying and eavesdropping on the transmissions. These channels are differentiated based on their nature and their security properties, as described in Subsection 2.2.1.2. The first one can be a Delayable-Authentic Out-of-Band channel since it has the purpose of exchanging an authentication parameter. Due to the constrained nature of our target devices, we have decided to choose a simple unidirectional visible light OoB channel based on a LED on the initiator (Alice) and a light sensor on the enrollee (Bob). This choice is based on the nature of the channel since it is hard for an attacker to replay or forge a message without being detected by the user. Also, it is less susceptible to the ambient noise than the acoustic or the haptic channels and easier to setup due to the close proximity assumption. For the second one, we have decided to apply a Delayable-Protected OoB channel since it serves as the final validation step of the pairing. This channel required a very limited user action that is represented by pushing a button on Alice after receiving a signal from Bob. This signal can vary between a vibration, a sound or

a simple LED blink. This choice of human-aided channel provides the user with an explicit feedback about the state of the pairing process. The used OoB channels are illustrated based on their purpose (🔵 verification of a parameter, 🟢 exchange of an information or 🔴 validation of a process) in Table 3.2.

3.2.3.2 Protocol Structure

After the discovery phase, the devices become aware of each other and agree on the Diffie-Hellman public parameters (the cyclic group $\mathbb{G}$, the generator g and a big prime p). At the same time, they start sensing the environment in order to collect a sufficient number of samples to perform the contextual encoding and decoding operations. They generate their ephemeral DH private keys (a and b), two secret l-bit nonces (r_a and r_b) and they derive their poisoned DH public keys ($g^{a-r_a}\ mod\ p$ and $g^{b-r_b}\ mod\ p$). In addition, Alice generates a hashing key K_h to avoid any exhaustive search attempts on the nonce r_a using a simple hash output $h(ID_A, ID_B, g^a, r_a)$. To simplify the expressions, we refer to the DH keys as g^{a-r_a} and g^{a-r_a}, without the modulus operation. Alice starts the pairing protocol execution as depicted in Figure 3.13. The symbols used to describe the scheme operations are presented in Table 3.2. The steps of the protocol are highlighted as follows:

- ❶ Alice sends g^{a-r_a} to Bob along with its identifier ID_A and the keyed hash $h_{K_h}(ID_A, ID_B, g^a, r_a)$ on the In-Band channel. She begins the transmission and the construction of her S-box using the CSI values that come after the sequence S_A shared with Bob for synchronization purposes.
- ❷ Bob starts the sensing at the reception of the first bit of Alice's message and he constructs his S-box using the CSI values that come after S_A. This operation is modelled by the construction of a contextual nonce r_{c_b} that serves as a parameter in a fuzzy commitment scheme. Afterwards, Bob transmits the parameters ID_B, g^{b-r_b} along with the fuzzy commitment value $V_b = r_{c_b} \oplus Encode(r_b || [g^{a-r_a}]_i^{i+l-1})$ to Alice on the In-Band channel. The parameter i is computed as follows $i = r_b\ mod\ (|g^{a-r_a}| - l)$.
- ❸ Alice extracts Bob's secret parameter $\hat{r_b}$ using her contextual parameter r_{c_a} as follows $\hat{r_b} || \widehat{[g^{a-r_a}]}_{\hat{i}}^{\hat{i}+l-1} = Decode(r_{c_a} \oplus \widehat{V_b})$. Then she verifies the correctness of the reconciliation of $\hat{r_b}$ based on the verification of $\widehat{[g^{a-r_a}]}_{\hat{i}}^{\hat{i}+l-1}$. The l-bit verification of g^{a-r_a} is used to improve the contextual mismatch detection time, which provides a way to enhance the usability in the case of an inattentive user placing the devices far apart. At this point, Alice sends the XOR of the three values $\hat{r_b}$, r_a and $\widehat{[g^b]}_{\hat{j}}^{\hat{j}+l-1}$ over the authenticated OoB channel. The parameter $\hat{j}$ is computed as follows $\hat{j} = \hat{r_b}\ mod\ (|g^b| - l)$ In the case of a *hash verification failure*, Bob emits a signal depending on the existing output interfaces to notify the user of such outcome.
- ❹ Bob recomputes $\hat{r_a} = r_a \oplus \hat{r_b} \oplus \widehat{[g^b]}_{\hat{j}}^{\hat{j}+l-1} \oplus r_b \oplus [g^b]_j^{j+l-1}$, verifies the received hash $h(ID_A, ID_B, g^a, r_a)$ and sends to Alice an HMAC value $h_K(ID_A, ID_B, \widehat{g^a}, g^b)$, using the shared key $K = (g^{a-r_a}.g^{\hat{r_a}})^b$, on the In-Band channel.
- ❺ Alice verifies the keyed hash received in the previous message. Then, she sends the hashing key K_h to Bob.
- 🔴❻ Bob verifies the keyed hash received in message ❶. Then, he provides a signal to the user to notify Alice of his validation by asking him to push a button on the other device. At the end of the protocol run, Alice and Bob share a secret DH key K that is used to encrypt the communications between them.

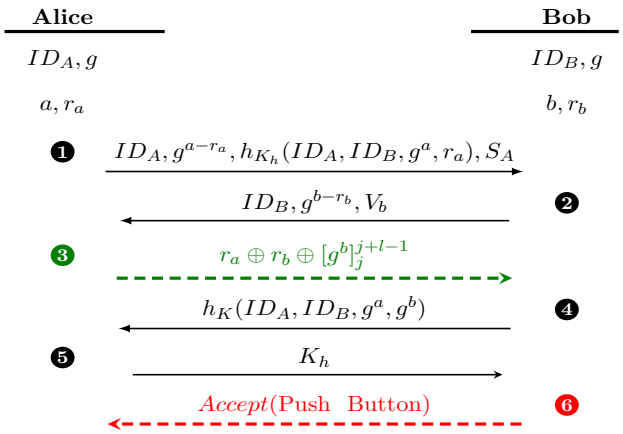


Figure 3.13: COOB: Contextual key agreement scheme with an authenticated OoB channel

The reason behind the use of the nonce exponentiation of the DH public keys, g^{a-r_a} and g^{b-r_b}, is to temporarily hide the real values of the legitimate devices DH public keys from the attacker. This secrecy is needed to guarantee the correctness of the second hash verification. To better explain this requirement, we describe an attack scenario. First, we start by assuming that we use the real DH keys instead of the hidden ones. The adversary injects his own DH public key g^x in the message ❷. At this point, the adversary has a perfect knowledge of the secret DH key computed by Alice, $K_A = g^{xa}$. Therefore, he has all the parameters needed to recompute the keyed hash sent in message ❹ which leads to bypassing the verification on Alice's side even when the value of Bob's nonce in the contextual commitment, sent in message ❷, has not been revealed by the attacker. As a consequence, the use of the real values of the DH public keys bounds the protocol security to a single hash verification instead of two. Thus, we have only l bits of security when we used $2l$ bits of authenticated exchanges against an ordinary contextual intruder which is not optimal. One possible solution to this issue is to use a commitment scheme, which needs two separate messages to provide the temporary secrecy property for a single public key. This requirement adds in a computation and communication cost of 4 exchanges for the two keys. This complexity can be easily avoided using the DH exponentiation to hide the public values while relying on a fuzzy commitment scheme that is based on an ambient information source. Also, this contextual technique makes the ordinary contextual attacker unable to reveal the values of the nonces for the entire protocol run with the exception of a successful random guess. Accordingly, this provides a permanent confidentiality of these security parameters instead of a temporary property. This approach makes the protocol optimal in term of security with less computational cost than the first proposal and, most of all, without adding a communication cost.

This novel approach combines two pairing techniques using two short nonces as a way of hiding the legitimate DH public keys from the attacker in order to prove their authenticity later on based on two hash verifications. The values r_a and r_b are protected by the discrete logarithm problem, which makes it hard for an adversary to retrieve them from the keys g^{a-r_a} and g^{b-r_b}, especially without the knowledge of the private keys a and b. To the best of the authors knowledge, COOB is the first scheme that combines the contextual and the OoB based pairing. This has been made possible using the exponential challenge-response technique that hides Alice's DH public key g^a. This security measure makes the adversary unable to recompute the keyed hash and fail to bypass the verification. Our hybrid protocol relies on a very constrained set of human interactions that consists of placing the devices in close proximity and pushing a button on Alice to confirm the pairing.

3.2.4 Security Analysis

3.2.4.1 Computational Security Evaluation

In this part, we aim at evaluating the attack success probability of a MITM attack, referred to as P_{adv}, that forces the participants, Alice and Bob, to compute different pairing keys, respectively K_A and K_B, at the end of the protocol execution $COOB(l, [Alice, Bob])$. The notation $COOB(l, [Alice, Bob])$ represents the execution of the protocol COOB between Alice and Bob using l-bit nonces. We refer to the security definition presented in [38] which is described as follows:

Definition 7. A protocol enabling an authentication of DH public parameter between Alice and Bob is secure if an adversary cannot succeed in deceiving Alice and Bob into accepting different DH public keys than the legitimate ones except with a satisfactory small probability $\mathcal{O}(2^{-l})$.

Proposition 1. *At the end of the protocol execution, $COOB(l, [Alice, Bob])$ satisfies the definition and both participants, Alice and Bob, accept the DH public keys with a satisfactorily small attack success probability $\mathcal{O}(2^{-l})$.*

Proof. In the normal execution, Bob verifies the keyed hash commitment sent in message ❶ to accept the exchanged DH public keys. If Bob opens it successfully, the *Accept* notification is then issued to Alice in message ❻. Thus, to win the game, the adversary has to commit on a keyed hash $h_X(Y)$ in message ❶ such that $h_X(Y) = h_X(ID_X, ID_B, Pub_{key_A}, nonce_A)$ (1) for some value Y. In this context, ID_X is the identifier of the initiator, Pub_{key_A} is the DH public key of Alice, $nonce_A$ is the nonce of Alice and X represents the hashing key that is received by Bob on the insecure channel in message ❺. The value Pub_{key_A} is derived by Bob based on the Alice's modified DH public key, Key_{m_A}, that is sent on the In-Band channel and the received nonce, $nonce_A$, through the OoB transmission:

$$Pub_{key_A} = (Key_{m_A})^{nonce_A}$$

Assuming that the used keyed hash function is universal, the probability $P_{collision}$ of finding a message $Y \neq (ID_X, ID_B, Pub_{key_A}, nonce_A)$ that satisfies the equation (1) is smaller or equal to $\frac{1}{2^{||h_X(Y)||}}$, as described in Subsection 2.1.4. On the other hand, the probability P_{guess} of correctly guessing a value Y where $Y = (ID_X, ID_B, Pub_{key_A}, nonce_A)$ is directly related to the correct guessing of the l-bit nonce $nonce_A$. This is due to power of the adversary to control the sent parameters to Bob: Key_{m_A}, ID_X and ID_B. Therefore, considering the following assumptions:

- The nonce $nonce_A$ is independently and uniformly distributed random variable.
- The guess of the adversary must be generated and submitted before $nonce_A$ is revealed.
- The protocol parameters are freshly generated with each pairing session.

The collision probability is negligible in comparison with the guessing probability $P_{guess} \leq \frac{1}{2^l}$ because $l \ll ||h_X(Y)||$. Thus, the adversarial success probability $P_{adv_B} = P_{guess} \leq 2^{-l}$. Afterwards, Bob notifies Alice of the outcome of the keyed hash verification in message ❻.

The sophisticated attacker is able to reveal Bob's nonce through the contextual commitment but he is forced to correctly guess Alice's nonce at the beginning of the protocol execution without any relevant information. Therefore, the attack success probability $P_{adv} = P_{adv_B} \leq 2^{-l}$. □

3.2.4.2 Formal Validation

To validate the protocol correctness in the symbolic model, we perform a formal verification using the TAMARIN prover [133], a powerful validation tool for security protocols. In our analysis, we begin with the evaluation of the confidentiality of the secret keys and nonces of Alice and Bob. Then, we evaluate an authentication property referred to as injective agreement that is presented in Subsection 2.1.5. This lemma verifies that the protocol guarantees to Alice that if she completes a protocol run with Bob to agree on a key K, then Bob has been apparently running the protocol with Alice and the two devices agreed on the same value. This property has been tested in both ways to guarantee a mutual authentication as mentioned in our code available in [1]. The multiple-session attack was not considered in our evaluation since we have no persistent secret during multiple protocol executions. These hypotheses reflect the consequences of a Man-in-the-Middle attack where the adversary performs the actions described in the previous subsection.

This tool adopts the Dolev-Yao intruder model on its public channel, which grants the attacker with a complete control over it. Thus, it satisfies our attacker model requirements on the In-Band channel. However, the authentic Out-of-Band channel is modeled in the tool such that it prevents the attacker from forging or replaying any messages. As for the *blocking* and the *delaying* actions, the adversary is already able to temporarily or permanently stop the process of sending an information, even on the authentic channel. As described in Section 3.2.3.1.1, the contextual block is modeled as a fuzzy commitment using the two functions $Encode(.)$ and $Decode(.)$, representing respectively the Reed-Solomon encoding and decoding schemes. This decision is motivated by the complete revelation of $r_c = r_{c_a} = r_{c_b}$ to the attacker at the beginning of the protocol, which facilitates the analysis for the TAMARIN prover and it enhances the adversary capabilities. Our sophisticated contextual attacker is represented as a Dolev-Yao intruder that has perfect knowledge of Bob's contextual information r_{c_b} which grants him a perfect reconstruction of the nonce r_b. Even though there is a lack of a modular exponentiation in the tool, we can model to a certain degree these operations just to reach the full capabilities of the intruder. Nonetheless, the XOR properties were recently modeled in TAMARIN v1.4.1 but the tool does not support the XOR of more than two terms, as required in message ❸, due to the exponential complexity. This computational burden is caused by the multiple algebraic properties of XOR such as the associativity, the commutativity, the cancellation and the neutral element. To ease the computation, we modeled our own approximation of the XOR operation using a constructor functions $xorc(.,.)$ to apply the operation on two variable inputs. The user notification after the first hash verification and the pairing validation on message ❺ are modeled as a message sent on a secure channel since the attacker has no power over it in practice.

To guarantee the correctness of the protocol execution, a set of restrictions must be indicated in the TAMARIN model. We enforced the use of an initialization rule that provides all the devices with the same contextual information. We imposed also the uniqueness of the private DH keys and of the authentication nonces to avoid any multi-session attack. Finally, we apply the hash equality restriction that stops the protocol run when the hash verification does not hold, which represents the case of an attack detection.

The results of the lemmas highlighted in Table 3.3 validate the robustness of our protocol in the symbolic model even in the presence of a sophisticated contextual attacker that can break the secrecy of the authentication nonces during the protocol run. The outcomes are either ✓ when the property is validated or ✗ when the property does not hold and an attack trace is provided by the tool. We use the automated proofs with the default heuristic and the default proof tree exploration. The validation lasted 84 minutes and was conducted on a computer with an Intel(R) Core^{TM} i5 – 9400H CPU @ 2.5GHz × 8 processor, 32 GB of RAM, running Ubuntu 18.04.4 LTS.

Table 3.3: COOB evaluated properties in the symbolic model

Property	**Result**	
	Ordinary contextual attacker	*Sophisticated contextual attacker*
Secrecy of r_c	✓	✗
Secrecy of r_a	✓	✗
Secrecy of r_b	✓	✗
Secrecy of Alice's key	✓	✓
Secrecy of Bob's key	✓	✓
Alice-to-Bob injective agreement	✓	✓
Bob-to-Alice injective agreement	✓	✓

Moreover, this analysis shows that an attacker is not able to mount an MitM attack resulting in the agreement on different keys on each device and guarantees the secrecy of the computed key has been validated for both Alice and Bob. Therefore, this analysis validates the mutual authentication property between the legitimate pairing parties chosen by the user and the secrecy of the communication link established for the post pairing phase. The case of multi-session attacks has not been addressed in this validation for two reasons. First of all, it adds significant computation cost due to the unbounded number of sessions that needs to be considered. Secondly, our scheme regenerates fresh parameters at the beginning of each session, which makes the assumption of having persistent security knowledge between two distinct protocol runs invalid. Therefore, relying on the security parameters from an earlier execution of the scheme is considered as a MITM attack where the adversary is trying to guess the appropriate nonce values, as explained in Section 3.2.4.1.

The main advantage of our protocol COOB is expressed as follows:

- **In the case of an ordinary contextual adversary**, COOB provides a $2l$ bit security level by using an efficient combination of a context-based scheme and an Out-of-Band channel.
- **In the case of a sophisticated contextual adversary**, any context-based pairing scheme can be compromised since the attacker is able to violate the safe zone requirement. However, COOB relies on the Out-of-Band channel to guarantee the authenticity of the exchanged DH keys by providing a l bit security level.

3.2.5 Usability Analysis

3.2.5.1 Experimental Setting

We have implemented COOB using C++ on two ESP32 microcontroller modules. This choice of cards is mainly motivated by the simplicity of the extraction and the manipulation of the CSI measurements using the WI-ESP tool [18]. The first ESP32 card is connected to a source of light, for example a LED, and the second one is connected to a photo-resistor in order to construct an authentic visual OoB channel. We use the Elliptic Curve Diffie-Hellman (ECDH) key exchange protocol based on the Koblitz curve secp256k1, SHA-256 for hashing and Wi-Fi as our In-Band channel. As for the choice of the elliptic curve domain parameters, we use by default in our implementation the recommended specifications provided in [35].

The Out-of-Band module apply an On-Off Keying (OOK) modulation and it takes 0.2 seconds to send a one-bit value. This transmission rate is explained by the choice of the

photo-resistor and the capacitor at the receiving side as shown in Figure 3.14. This RC light detection circuit is used because of the digital nature of the Raspberry Pi pins and their inability to read analog inputs. Therefore, the charging time of the RC circuit is used as a reference when applying an internal counter to detect the existence of a light pulse when compared with a threshold computed with regard to the ambient luminosity level at the time of pairing. For synchronization purposes, we added four bits "1110" at the beginning of the OoB bit sequence to announce the start of the transmission in the case where the message starts with a 0 bit-value, since it is represented by an off key. The reason behind the use of four bits instead of one is to reduce the transmission errors due to a late triggering of the detection that sometimes starts at the second or the third bit.

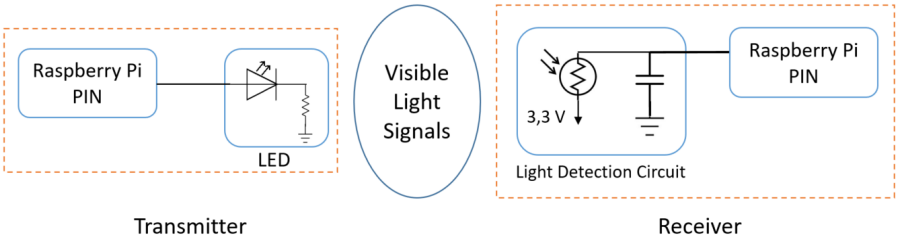


Figure 3.14: Visual Out-of-Band channel design

The contextual module is assumed to apply a reconstruction threshold that represents the maximum number of bits that can be corrected by the Reed-Solomon codes during the secret reconciliation phase. We fixed the value of the threshold to 20% of the total hidden value bit-length $|r_b| + |[g^{b-r_b}]_j^{j+l-1}| = 2 \times l$ to tolerate any encoding errors by the legitimate devices. This fault tolerance is expected to increase the contextual secret message bit-length $|Encode(r_b||[g^{b-r_b}]_j^{j+l-1})| = \lceil 2.4 \times l \rceil$ while providing a more reliable encoding scheme.

3.2.5.2 Performance Evaluation

In this part, we evaluate the performance of our scheme COOB by computing its pairing completion time in three different environments that cover all the security characteristics of the deployment area:

i. *Secure Environment:* In this deployment area, the territory is actively monitored and there are natural barriers that prevent the adversary from being in close proximity with the legitimate IoT devices. Thus, the threat model that is adopted in this scenario is the ordinary contextual intruder.

ii. *Hostile Environment:* In this deployment area, the territory is not under the control of the user and there are not natural barriers that prevent the adversary from being in close proximity with the legitimate IoT devices. Thus, the threat model that is adopted in this scenario is the sophisticated contextual intruder.

iii. *Unknow Environment:* In this deployment area, the territory is not actively monitored but there are natural barriers that might prevent the adversary from being in close proximity with the devices. In this scenario, we cannot define the adequate threat model that is suitable for modeling the attacker capabilities.

Based on these three deployment scenarios, we aim at maximizing the utility and usability of the chosen protocol in order to provide the optimal pairing time and the desired security guarantees.

3.2.5.2.1 Secure Environment

We start by accounting for the time needed by the chosen contextual module, in our case the TDS scheme [204], in order to compute the pairing time required by our hybrid protocol COOB. In order to clearly evaluate the performance of our scheme, we compare it to the same protocol design in terms of exchanges, key manipulation and cryptographic primitives but without the contextual module. This scheme is referred to as $2l$-OoB and it sends $2l$ bits on the OoB channel to match the same level of security of our hybrid protocol in this specific deployment scenario, as illustrated in Figure 3.15. This operation aims at assessing the advantage of applying the contextual model in order to reduce the pairing time by rapidly transferring half of the security bits. The pairing time of the two schemes have been averaged over 5 executions that were conducted for a number of bits l varying between 16 and 88 bits. The evaluated bit-length limit of 88 cannot not be exceeded due to the memory limitations of these cards when storing the collected CSI information prior to the S-Box computation process. The results were analyzed to provide a time percentage gain that reflects the added value of our modular hybrid design.

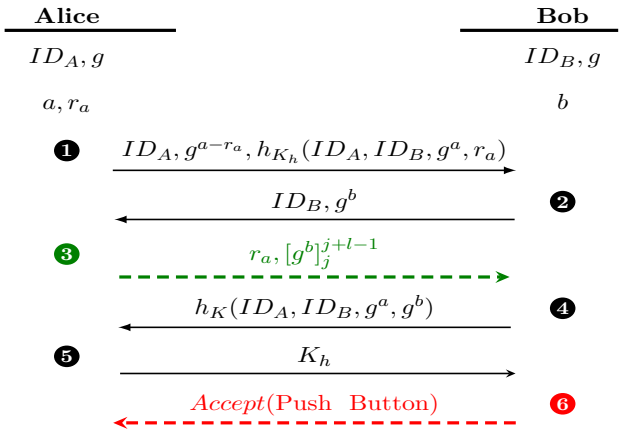


Figure 3.15: 2l-OoB pairing protocol

As highlighted in Figure 3.16, the pairing time imposed by a solely OoB-based scheme that sends $2l$ bits on the auxiliary channel grows rapidly to reach 37 seconds for a bit-length $l = 88$ bits. Our implemented OoB-based protocol achieves a better performance compared to the published usability results in the work of Kumar et al. [56] that take on average 28.8 seconds for $l = 15$ bits on a visual channel. Therefore, we use our $2l$-OoB pairing protocol performance results to conduct the comparative study. Our hybrid scheme takes advantage of the fast contextual agreement module to keep the required association time within a reasonable limit equal to 21 seconds. This comparison is better described using a time percentage gain that reflects COOB pairing time reduction while maintaining the same level of security. This time optimization ranges between 34 and 41 percent, as shown in Figure 3.17, for a nonce bit-length $l \in [16, 88]$.

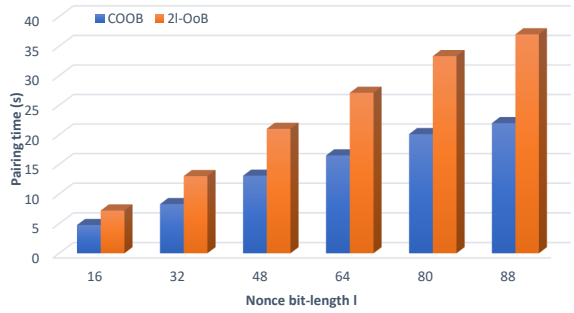


Figure 3.16: Pairing time performance comparison: COOB vs 2l-OoB scheme

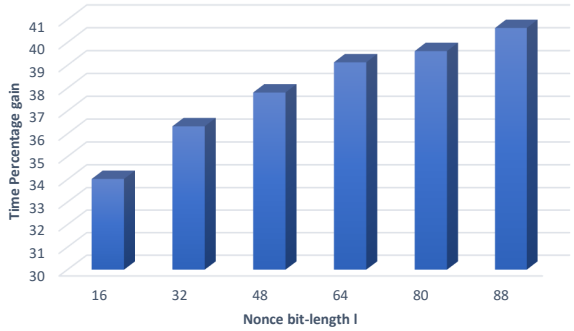


Figure 3.17: Average time percentage gain

In this first deployment scenario, the use of the hybrid protocol COOB optimizes the pairing time in comparison with the use of an OoB-based scheme. However, we recommend the use of a contextual protocol due to its usability advantages. Also, the adopted threat model renders the use of Out-of-Band channels as an unattractive option since the adversary is unable to attack the contextual protocol in the first place.

3.2.5.2.2 Hostile Environment

In this second deployment scenario, the use of context-based pairing protocols is not considered as a secure option anymore. This is explained by the adversarial ability to sense the same contextual features as the legitimate devices which would compromise the key agreement process. Thus, the use of an OoB-based protocol is considered as the only viable option to perform the ad-hoc pairing. The application of the hybrid protocol against the sophisticated intruder reduces the security level to l bits in comparison with the $2l$ bits of the Out-of-Band protocol. From the usability perspective and for the same level of security, the hybrid protocol only adds a negligible additional 7% delay to the pairing time that is taken by the $2l$-OoB scheme. This overhead represents the execution time of the contextual module by the hybrid protocol.

3.2.5.2.3 Unknown Environment

In this third deployment scenario, the adversary may or may not have the ability to violate the safe zone requirement due to the lack of active monitoring of the pairing environment. Thus, we would like to maximize the usability of the chosen pairing solution while maintaining an acceptable level of security against the possible threats. The contextual pairing techniques can be only used securely in the first environment. On the other hand, the OoB-based solutions can be adopted in all the deployment environments while preserving the same level of security but with a considerable pairing completion time. The third possible option is the use of the hybrid protocol that guarantees the resilience of the pairing operation against the two threat models while optimizing the execution time

compared to the OoB-based scheme in the secure environment and providing similar performances with the same level of security in the hostile scenario. Thus, the hybrid solution is the most suitable approach that maximizes the usability for the human operator and, at the same time, offers the desired security level.

3.3 Secure Pairing Design Recommendations & Future Challenges

One of the critical parts of designing a secure device pairing that is based on an Out-of-Band channel is the assessment of the security guarantees provided by this auxiliary communication medium. This is explained by the absence of any prior knowledge between the pairing parties and the lack of trust in the In-Band channel since it is under the control of a powerful Dolev-Yao Intruder. Therefore, the OoB channel presents the only source of security in the protocol. As a consequence, if the security properties, assumed guaranteed in the design phase, are somehow violated by the attacker then the protocol's security is in jeopardy. The Bluetooth Secure Simple Pairing protocol represents one of the most widely used security pairing scheme with its three variants: *PIN Entry* inspired from the MANA III protocol, described in Subsection 3.1.1.3, *Numerical Comparison* inspired from the MANA II protocol, described in Subsection 3.1.1.2, and *Out-of-Band channel* which uses the NFC technology. The most deployed ones are *PIN Entry* and *Numerical Comparison*. They rely on the user involvement to either enter a PIN into both devices or to compare and confirm the match between two six-digit number displayed on the objects. Many research works, such as [109, 186], pointed out numerous vulnerabilities related to the human-factor error resulting from the previously described user actions, e.g., the entry of a predictable PIN or the confirmation of mismatched authentication digits due to a rush behavior. Another existing design flaw among the secure device pairing schemes is the use of confidential Out-of-Band channels that are hard to reach due to eavesdropping and side-channel attacks. In the work of Han et al. [80], the authors propose a device pairing protocol between a smartphone and a vehicle, called MVSec, that is based on a confidential exchange of a nonce at the beginning of the execution. This confidential channel is unidirectional visible light communication from the car to the device inside the closed glove compartment. According to the attacker model adopted, the adversary can be inside the vehicle and the fact that the light transmission happens inside a close area makes it confidential. Due to the feasibility of the eavesdropping attack using the electromagnetic side channel [187] from a reasonable distance such as an attacker sitting inside the vehicle, the nonce confidentiality assumption no longer holds which compromises the security of the protocol.

The use of the formal or the computational security assessment techniques can be a powerful way to evaluate the confidentiality and the authentication properties provided by the device pairing protocols. However, the only drawback of these methods resides in the formulation of the assessed properties that may not reflect the desired degree of security. Therefore, we might end up with an incomplete security analysis or with conflicting results by evaluating two slightly different formulations of the same property as demonstrated in Table 3.1 in the case of the MANA II protocol. Accordingly, the formulation of these properties should be specified to mitigate the previously discussed issues as detailed in the work of Lowe [120]. Furthermore, we have noticed that the automated computational analysis using tools such as CryptoVerif [30] does not support the use of Out-of-Band channels which eliminates the feasibility of performing a complete computational evaluation of numerous device pairing protocols. This is considered as an issue in the device pairing context due to the common use of short authentication strings in the key confirmation phase which is not usually addressed in the symbolic model. Thus, any vul-

nerabilities that exploit the computational weaknesses of the protocol will not be disclosed and, consequently, mitigated. The conducted security evaluations, in both the symbolic and the computational model, demonstrate the necessity of conducting both verifications in order to confirm the resilience of a scheme. This is due to the aspects addressed by each model: the focus on the protocol structure and the exchanges in the symbolic analysis and, also, the focus on the computational robustness of the cryptographic primitives. Also, we noticed that the effectiveness of the formal analysis lies in the proper formulation of the security properties under investigation which will, consequently, permits the comparison of the protocol performances. Furthermore, we cannot stress enough the need for a normalized taxonomy in order to enhance the understanding of these security verifications and to better clarify the reasons behind any contradictions between the evaluation outcomes.

Another aspect, that should not be neglected by future work in the secure device pairing field, is the consideration of the advanced threat model, described in Subsection 2.1.2.2, in the security assessment. Also, there is an imminent need for a possible and a feasible mitigation against this imminent threat using context-based pairing solutions or distance-bounding techniques since the use of Out-of-Band channels does not provide the necessary security. Finally, with the growing demand for usable and secure device pairing protocols, we noticed the interest in using context-based schemes, also referred to as Zero-Interaction protocols [67]. However, the security analysis of these techniques are often limited to assessing the randomness of the collected measurements from the ambient environment which reflects the robustness against passive attacks. Such analysis cannot provide the necessary guarantees to formally or computationally validate the security of the pairing procedure as demonstrated in the work of Wu et al. [202] by disclosing a brute-force attack against the interlock protocol applied in the MagPairing protocol [92] that would have been detected using a computational security analysis. Therefore, there is a need for a proper modeling of these pairing schemes based on the security specifications of their chosen contextual features.

3.4 Conclusion

In this chapter, we have surveyed the formal and the computational security analysis that are conducted on a number of secure device pairing protocols by describing their threat models, their evaluated properties and their adopted verification models. Although every analysis tends to use its own terminologies and definitions, we have normalized the used taxonomy in order to enhance the understanding of these security verifications and to better clarify the reasons behind any contradictions between the evaluation outcomes. In addition, we have proposed a secure device pairing protocol, referred to as COOB, that efficiently combines the use of an Out-of-Band channel with a state-of-the-art fast contextual pairing scheme. This hybrid protocol enhances the security against a sophisticated contextual attacker that completely controls the ambient environment. Thus, this advanced threat model is not supported by the existing contextual pairing protocols.

Moreover, we have discussed the recently published misbinding attack that affects all SDP protocols by exploiting the combination of the lack of hardware protection and the human factor error to lure the user to pair with a malicious device. This section motivates the use of a formal or a computational security analysis to validate the correctness of the SDP schemes that will be proposed in the future.

4 | Secure Device Enrollment

Contents

In this chapter, we provide an in depth overview of the state-of-the-art model-based PUF enrollment protocols. We conduct a classification of the existing proposals based on two identified architectures. In addition, we describe the different components of the protocols and we discuss their respective weaknesses. Also, we evaluate the robustness of the identified enrollment protocols against an insider threat scenario that targets the secrecy of the PUF ML model. Moreover, we present our enrollment protocol, Water-PUF [102], that relies on the use of an ML watermarking technique to identify the use of a leaked PUF model. Thus, it represents a promising candidate solution that directly addresses the insider threat scenario.

In Section 4.1, we introduce the key components of an entity authentication protocol that relies on the use of a machine learning model and a physical unclonable function. In Section 4.2, we analyze a selection of ML-based PUF enrollment protocols in order to highlight their limitations and the potential improvements. Based on the findings of the conducted analysis, we propose, in Section 4.3, our enrollment protocol that exploits a ML watermarking technique to mitigate the risks related to a leaked critical information, such as the PUF model, to an adversary. In Section 3.4, we conclude the second part of the thesis.

4.1 Model-based PUF Authentication Procedure

In this section, we describe the authentication process of an IoT object based on the use of a mathematically clonable PUF based on a number of ML techniques. The procedure consists of multiple entities that participate in verifying the identity of this particular device. These entities constitute two generic architectures that represent the steps of an enrollment protocol. Each of these components is defined and characterized based on the roles and the modules that are specified by the protocol designer. The building block diagrams in these two architectures can help to design and assess independently the system components of these schemes with respect to the adopted threat model. Furthermore, it provides a global insight of the enrollment process and the components. This section introduces the insider threat model in the enrollment process that is usually overlooked by the designers. This model aims at assessing the robustness of the protocols against an information leakage scenario of the secret PUF model to an adversary.

4.1.1 Enrollment Architectures

The existing Model-based PUF authentication protocols can be classified based on two generic architectures that we refer to as Three-components Enrollment (3CE) and Four-components Enrollment (4CE) procedures. As the name states, the former approach require the existence of three main high-level roles:

- **Prover:** The IoT object that needs to be enrolled in the network of the user based on the PUF hardware onboard of it.
- **Communication Channel (CC):** The chosen communication channel between the other components.
- **Authentication Server (AS):** The entity that manages the storage and the accessibility to the PUF model. Furthermore, it performs the enrollment procedure with the Prover as the Root-of-Trust (RoT) [213] in the authentication process through the chosen communication channel.

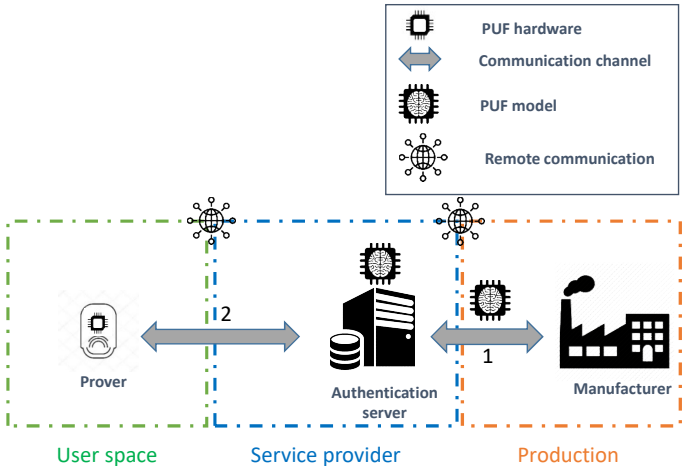


Figure 4.1: Three-Component enrollment procedure

This approach, typically, requires the unauthenticated IoT object to connect to the network of the user to remotely communicate with the authentication server, as illustrated in Figure 4.1. On the other hand, the latter architecture is slightly different since it exploits a delegated Root-of-Trust (RoT) [213] role, referred to as the Verifier. The four components of this approach are described as follows:

- **Prover:** The IoT object that needs to be enrolled in the network of the user based on the PUF hardware onboard of it.
- **Communication Channel (CC):** The chosen communication channel between the other components.
- **Verifier:** The designated entity that performs the enrollment procedure with the Prover on behalf of the RoT in the authentication process through the chosen communication channel. This role and the authentication server constitute the Chain-of-Trust in the enrollment procedure.
- **Authentication Server (AS):** The entity that manages the storage and the accessibility to the PUF model. Moreover, it adds the enrolled Prover to the list of authorized devices to join the network based on the validation of the Verifier.

The delegated Root-of-Trust acts as the local challenger of the IoT device, as shown in Figure 4.2. Therefore, it prevents the risks related to connecting an unauthenticated object to a poorly isolated network. Furthermore, it helps to decrease the communication and computation costs on the server side. Thus, the appliance of the Verifier role enhances the scalability of the enrollment procedure.

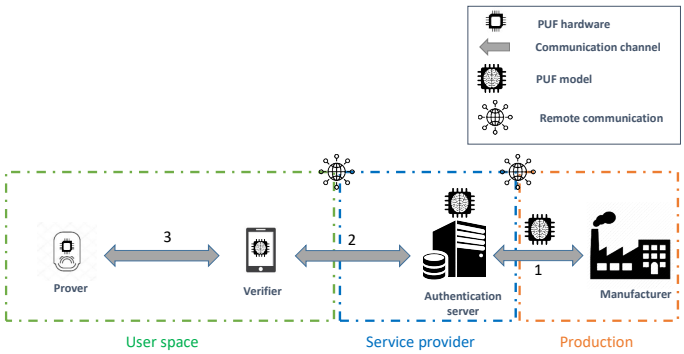


Figure 4.2: Four-Component enrollment procedure

4.1.2 Components Overview

In this subsection, we describe the roles and the modules that constitute each component. These generic elementary units serve as building blocks to the previously introduced architectures.

4.1.2.1 Prover

The Prover role represents the IoT object that holds the PUF hardware. This secure element represents a mean to perform the entity authentication procedure. Depending on the adopted enrollment architecture, the IoT device can be given the access to the network of the user prior to the authentication process to communicate with the AS, as illustrated in Figure 4.1. However, in the case of the 4CE approach, the Prover is limited to using local communications with the Verifier until the successful execution of the enrollment protocol.

The appliance of a PUF ML model in the protocol design is an admission that this secure element can be mathematically cloned when the adversary has a sufficient number of challenge-response pairs. Therefore, additional protection techniques should be implemented to prevent the attacker from constructing his own precise PUF model. Following

the specifications in the work of Maes [123], the added security measures classifies this PUF construction as *Controlled PUF*. The Prover role is established based on three main elementary units, as highlighted in Figure 4.3, that manage the Input-Output transformation. The three sub-components are described as follows:

- **Challenge Preparation (CP):** The CP unit is responsible for receiving and for preparing the received challenge from the Verifier. This part can be classified into three main categories:
  - *Direct Reception:* The received challenges can be fed directly to the PUF hardware.
  - *Mutual Construction:* The Prover and the Verifier collaborate to compute a common seed to generate the set of challenges. One simple example of this operation is to exchange nonces that are concatenated to create the shared seed value.
  - *Challenge Derivation:* The Prover receives a single l-bit challenge that is manipulated to extract a set of l challenges. As an example of this operation, the receiver can apply a linear-feedback shift register to the received root challenge.
- **Challenge Verification (CV):** The CV unit is responsible for verifying the validity of the challenges that are fed to the PUF hardware. For example, the verification process may aim at ensuring that the received challenges have not been executed before. This technique is considered a mitigation against the reliability attack that has been proposed in the work of Becker [25].
- **Controlled PUF (CPUF):** The CPUF unit constitutes the most important component on the Prover side. This part is responsible for generating and for obfuscating the PUF responses. The CPUF holds three main aspects to be described:
  i. *PUF Architecture:* The chosen PUF construction to be implemented on the Prover.
  ii. *Reconfigurability:* This aspect is only discussed in the case of FPGA. The integrated circuit onboard of the Prover can be reconfigured by the Verifier to impose a specific behavior of the PUF.
  iii. *Obfuscation Technique:* The specification of the chosen approach to hide the responses from the adversary to prevent any modeling attacks based on the collected CRPs.

4.1.2.2 Verifier

The Verifier role is considered as the local Root-of-Trust that initiates the challenge-response process with the Prover, as illustrated in Figure 4.2. This component plays a crucial role in generating the enrollment challenges and in verifying the validity of the received obfuscated responses. In this context, the Verifier takes advantage of the received PUF model from the authentication server to perform the enrollment process, as illustrated in Figure 4.4. The verification responsibility can be divided into two main parts:

- **Response Re-computation:** The Verifier applies the chosen challenges to the PUF model to extract a set of probably approximately correct responses.
- **Response Verification:** This process uses the received responses from the Prover and the re-computed values from the PUF model to validate the identity of the sender.

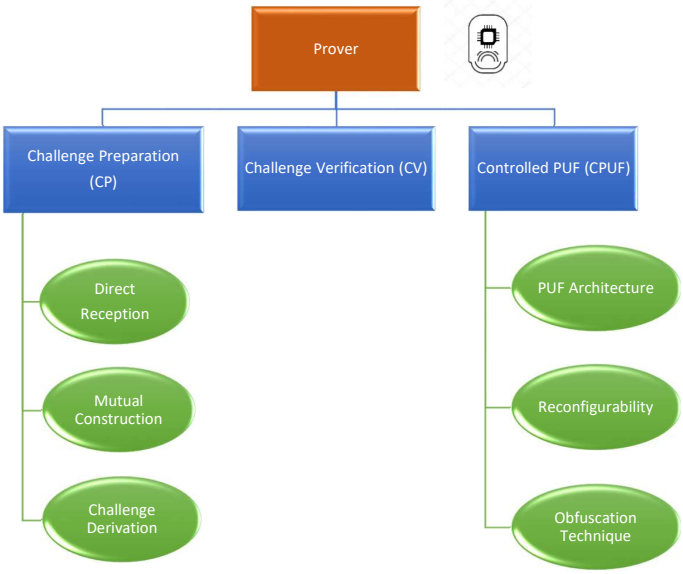


Figure 4.3: Key elements of the Prover role

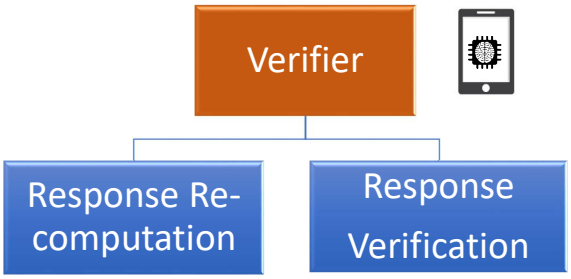


Figure 4.4: Key elements of the Verifier role

4.1.2.3 Authentication Server

The authentication server is considered as the primary Root-of-Trust in both architectures. This component guarantees the integrity and, in most case, the confidentiality of the PUF model depending on the required security properties of the enrollment protocol. Consequently, the AS can be classified into three categories based on these security guarantees, as shown in Figure 4.5. The classification of the AS operational mode is described as follows:

- **Public Database:** The authentication server has to guarantee the integrity of the PUF model that can be accessed publicly by any participant.
- **Private Database:** The authentication server has to guarantee the integrity and the confidentiality of the PUF model that can only be accessed by the authorized users.
- **Root Authenticator:** The authentication server stores the PUF model under one of the previous database modes. Furthermore, it fully plays the role of the Verifier as introduced in the 3CE architecture.

4.1.2.4 Manufacturer

The manufacturer plays the initial role of constructing the Prover hardware. He extracts enough challenge-response pairs to train the PUF ML model and he sends it securely to

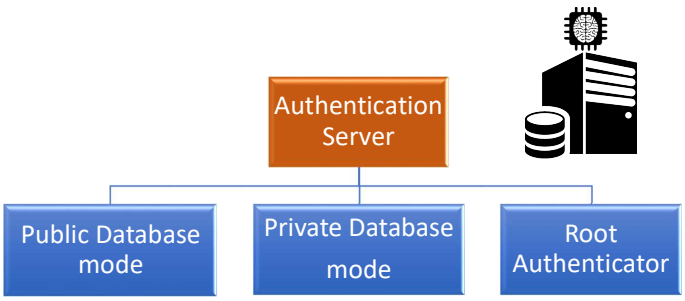


Figure 4.5: Key elements of the Authentication Server role

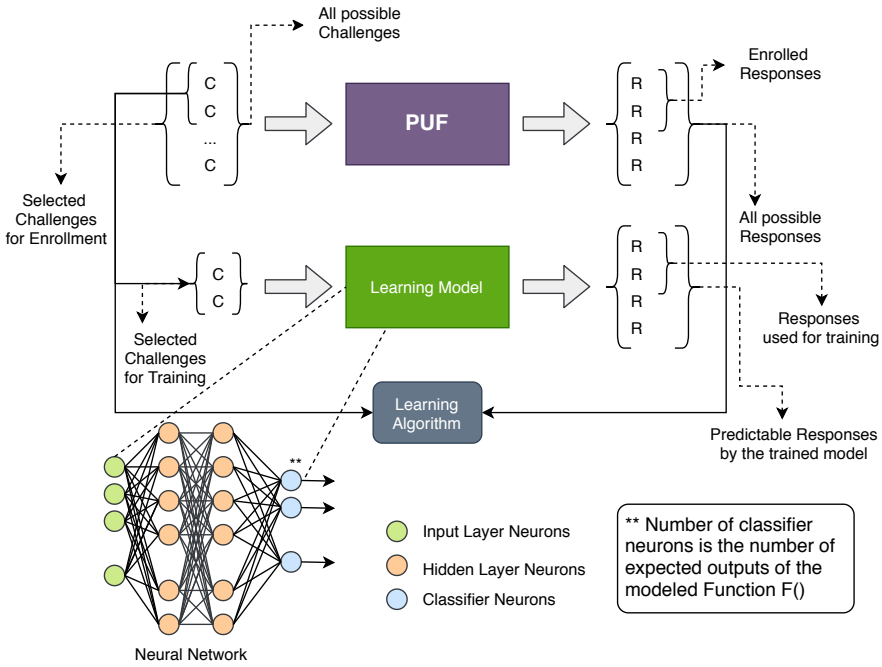


Figure 4.6: ML model training of PUF circuits [153]

the authentication server. Figure 4.6 illustrates the steps of the model training process based on the use of neural network. These actions mark the end of the participation of the manufacturer role in the enrollment process.

4.1.3 Threat Models

The adopted threat models in the existing ML model-based PUF authentication protocols can be categorized depending on the accessibility properties to the ML model in question. As described in Subsection 4.1.2.3, the *Private Database* and the *Root Authenticator* modes require the Authentication Server to keep the PUF model as a secret and to only provide access to the trusted users. Thus, the adversary cannot get hold of the PUF model and he can only attack the system through external actions such as eavesdropping or replaying the exchanged messages between the enrollment entities. This attacker falls into the **Outsider Threat** category. However, The *Public Database* mode assumes that any user can obtain the model without any restrictions. The security of this mode is assured by relying on additional assumptions on the attacker capabilities. Regarding the adversarial powers on the communication channel, he is able to eavesdrop on the exchanges between the Prover and the Verifier or the Authentication Server depending on the adopted enrollment architecture. Furthermore, he can actively query the PUF holder with his own challenges.

This action aims at collecting enough CRPs for the attempted model reconstruction attack in the case of the private operational modes. However, the adversary is assumed unable to conduct invasive attacks on the Prover software which guarantees the correctness of the enrollment protocol execution. This assumption can be assured through the use of lightweight integrity verification of IoT systems such as the remote attestation schemes [39, 14]. As a consequence, the adopted threat models are classified as follows:

- **Public Model Adversary (Pub-Adv)**: The goal of the adversary shifts from modeling the PUF hardware to attacking the additional security mechanisms in order to bypass the authentication process. For example, he can focus on reducing the response generation time using the public PUF model to bypass the time-bound assumption.
- **Private Model Adversary (Priv-Adv)**: The adversary aims at creating a precise PUF model based on the obfuscated challenge-response exchanges. This ML model serves as a tool to predict the correct responses to the challenges of the Verifier as a way to enroll malicious devices.

The two previously detailed attacker categories can be further extended to assess the robustness of the enrollment protocol against an adversary that can get hold of the PUF model that is used in the authentication process. This scenario is considered as an **Insider Threat** within the information system of a particular organization. The attack is based on an individual with sufficient access privileges who violates the non-disclosure policies by leaking sensitive information, such as the PUF models. These leaks should be impossible to be traced back to this particular individual. This scenario is only applicable in the context of the 4CE architecture where the verifier might be the source of the leakage since it represents the role with the least level of trust in comparison with the Authentication Server. On the other hand, the verifier is assumed to be able to properly perform the authentication process without the risk of fraudulently enrolling malicious devices. This is due to the fact that the enrollment process of a particular device can lead back to the responsible individual once the malicious object is discovered. However, it is not the case with the PUF model since it is shared between all the potential operators which eliminates any possibility to detect an information leakage incident or to discover its source.

4.2 Enrollment Protocols Analysis

In this section, we study a selection of model-based PUF enrollment protocols based on the previously identified architectures. The different modules that are applied in the components of these schemes are described and detailed. Afterwards, we provide a security overview of the identified weaknesses in the protocol design and we suggest the adequate mitigation.

4.2.1 Time-bounded Authentication Protocol

This enrollment scheme was proposed in the work of Majzoobi and Koushanfar [125, 126] to target the issue of having a public model architecture of the PUF. The security of the protocol is based on the assumption that the time required to generate the responses by a PUF hardware is significantly smaller than the time required to predict them using a machine learning model. Thus, it is possible to verify the origin of the responses that are received by the verifier to avoid any possible ML-based impersonation attacks. The main steps of the time-bound authentication process are illustrated in Figure 4.7. This proposal is based on the 4CE architecture and adopts the public adversary threat model that are described respectively in Subsection 4.1.1 and in Subsection 4.1.3.

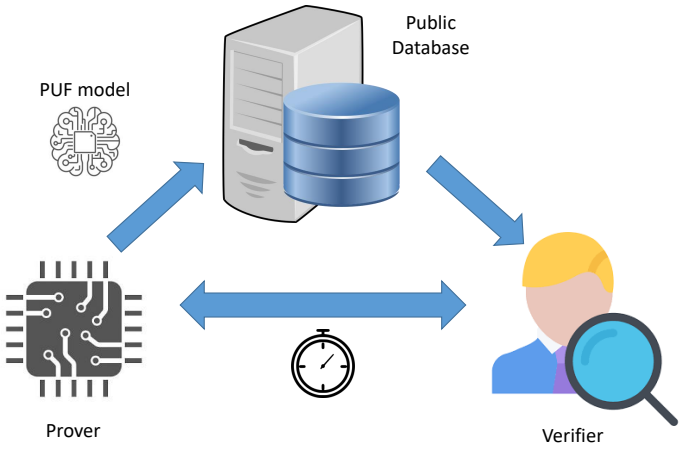


Figure 4.7: High-Level representation of the time-bound authentication protocol

4.2.1.1 Protocol Components

4.2.1.1.1 Prover

In this protocol, the operational characteristics of the Prover are systematically described based on the following three sub-components:

- **Challenge Preparation:** Direct Reception.
- **Challenge Verification:** The received challenges are not verified.
- **Controlled PUF:**
  i. *PUF Specifications:*
     – Nature: Electronic.
     – Architecture: C-RPUF (Subsection 2.3.2.3).
  ii. *Reconfigurability:* This option is fully supported.
  iii. *Obfuscation Technique:* This mechanism is not applied. The responses are returned to the Verifier without any modification.

4.2.1.1.2 Verifier

The operational characteristics of the Verifier are systematically described based on the following two sub-components:

- **Response Re-computation:** The Verifier applies the Public model, that is stored in the Authentication Server, and the desired reconfiguration to predict the responses of the Prover.
- **Response Verification:** The Verifier evaluates the execution time of the challenge-response process. The verification of the PUF output happens only if the responses are received within a pre-fixed time threshold. When the time-bound assumption is satisfied, the response verification process is conducted by a simple bitwise comparison.

4.2.1.1.3 Authentication Server

The AS in the protocol is playing the role of a *Public Database*. Therefore, the PUF model is also accessible to the adversary. However, the integrity of the stored PUF model is assumed guaranteed.

4.2.1.2 Security Assessment

Authentication Property. *The Verifier authenticates the Prover only if the time the Prover takes to generate the correct response is less than the time-bound threshold.*

To handle the public accessibility to the PUF model, the work of Majzoobi and Koushanfar [126] uses a time-bounded method that prevents the prover from applying a PUF model since it takes more time than just feeding the challenge as an input to the PUF hardware. In addition, the messages containing the configuration bitstream provides an insight about the placements of the specific PUF cells to be used in the case of a re-configurable PUF. However, the adversary is assumed to be unable to reverse-engineer this information which prevents him from knowing the used PUF configuration. This assumption suggests that the attacker does not have a perfect knowledge of the protocol structure which partially supports the *Security Through Obscurity (STO)* policy. Thus, this mechanism might be vulnerable to the attack on the distance-bounding protocols [81, 34]. Therefore, there is a need for a new method to guarantee that the source of the response is indeed the PUF hardware and not the used model. Since the manufacturer constructs the model of the PUF and he stores it in a publicly accessed database, the adversary is assumed to be able to obtain it as it is the case for any legitimate user. In order to prevent the attacker from using the PUF model to respond to the challenge, the Verifier applies a time-bound authentication proof to the challenge-response process based on the assumption that the time required for the response simulation is longer than the time required by the hardware PUF. This assumption is only valid if the minimum response simulation time, represented as t_{min}^{sim} , is larger than the upper bound delay for generating an authentic response by the hardware that is represented as Δ_{max}.

The time-bound assumption is based on the computational limitation of the adversary and the variation in the channel latency to guarantee the correctness of the authentication process. This explains the use of the additional STO assumption about the infeasibility of decoding the configuration bit-stream by the attacker that prevents him from efficiently simulating the behavior of the PUF. In addition, this particular protocol is mainly designed for FPGAs which makes it unsuitable for the application-specific integrated circuits such as the majority of the IoT devices. Thus, the reconfigurability technique cannot be applied to thwart the risks of bypassing the time-bound authentication.

4.2.2 Slender PUF Protocol

The Slender PUF protocol was proposed under two versions. The conference version was first introduced in the work of Majzoobi et al. [128] to present a new response hiding technique that is based on pattern matching. However, the journal version [162] represents an extension of the response obfuscation through a pseudo-random padding of the selected sub-string. These two proposals are based on the 3CE architecture and they adopt the private model adversary. The details of these two terminologies are described respectively in Subsection 4.1.1 and in Subsection 4.1.3.

4.2.2.1 Protocol Components

4.2.2.1.1 Prover

In this protocol, the operational characteristics of the Prover are systematically described based on the following three sub-components:

- **Challenge Preparation:** Mutual Construction through a nonce exchange.
- **Challenge Verification:** The received challenges are not verified.

- **Controlled PUF:**

  i. *PUF Specifications:*
  - Nature: Electronic.
  - Architecture: 4-XOR Arbiter PUF (Subsection 2.3.2.2).

  ii. *Reconfigurability:* This option is not supported.

  iii. *Obfuscation Technique:*
  - **Conference version [128]:** The Prover generates a random index $ind \in [0, l-1]$ that represents the first bit of the truncation. Afterwards, he extracts the l_{sub} bits sequence from the l bit PUF response to the sent challenges. Then, he sends it to the Verifier to validate the enrollment procedure.
  - **Journal version [162]:** The Prover conducts the same operations to find the substring response as in the conference version. Then, he generates an additional random $(l - l_{sub})$ bit sequence that serves as padding for the substring. Finally, he inserts the truncated response at a random index $ind_2 \in [0, l - l_{sub} - 1]$ of the generated circular padding sequence.

4.2.2.1.2 Verifier

The operational characteristics of the Verifier are systematically described based on the following two sub-components:

- **Response Re-computation:** The Verifier uses the PUF secret model, that is stored in the Authentication Server, to precisely compute the expected hardware response.
- **Response Verification:** The verification phase is the same for both versions of the protocol. The Verifier tries to find a match between the substring and the simulated PUF response through a maximum sequence alignment. The enrollment is validated under two conditions: the substring alignment should produce a match and the hamming distance between the two sequences should be less than a predefined threshold. The latter condition is applied to support the noisiness in the PUF responses.

4.2.2.1.3 Authentication Server

The AS in the protocol is playing the role of a *Root Authenticator*. Therefore, the PUF model is not accessible to the adversary. Consequently, the Authentication Server has to guarantee the confidentiality and the integrity of the PUF model.

4.2.2.2 Security Assessment

Authentication Property. *The authentication is successful if the Prover response substring matches at some location in the Authentication Server estimated response string within a predefined threshold.*

The two versions of the Slender PUF protocol have been put into test in the work of Becker [25]. In this experiment, the author has applied the CMA-ES [83] machine learning algorithm, detailed in Subsection 2.3.3.4. In the case of the attack on the Slender PUF protocol, Becker has targeted the main security assumption that the adversary can only compromise the protocol by guessing the truncation indexes, ind_1 and ind_2. This assumption aims to establish that the only possible technique to model the PUF hardware

is to map the substring response sequence to the corresponding challenges. The proposed attack counters this assumption by using a Pearson correlation coefficient $corr(.)$ [28] as a fitness test between the Hamming weights of the generated responses from the parent PUF instances, $HW(R_i)$, and the Hamming weights of the collected substrings, $HW(W_i)$. The choice of this fitness function is motivated by the assumption that the higher the computed correlation is, the more accurate the PUF instance. This technique has efficiently modeled the protected hardware PUF using different levels of noise and two constructions of PUFs (3-XOR and 4-XOR Arbiter PUF). The added noise has been applied to simulate the unreliability percentage of the collected hardware PUF responses. The accuracy of the modeled 4-XOR Arbiter PUF has reached 97.2% using 600000 noiseless CRPs. However, the additional 29% noisy responses have reduced the accuracy to 92.5% using 1200000 samples.

4.2.3 Noise Bifurcation Protocol

The Noise Bifurcation protocol was introduced in the work of Yu et al. [207] to present a novel response hiding technique. The scheme selects only specific responses to be returned to the Verifier. Thus, the attacker is assumed unable to associate the challenges and their corresponding responses. The proposal is based on the 3CE architecture and it adopts the private model adversary. The details of these two terminologies are described respectively in Subsection 4.1.1 and in Subsection 4.1.3.

4.2.3.1 Protocol Components

4.2.3.1.1 Prover

In this protocol, the operational characteristics of the Prover are systematically described based on the following three sub-components:

- **Challenge Preparation:** Mutual Construction through a challenge exchange. The master challenges are referred to respectively as C_p for the one generated by the Prover and C_v for the one generated by the Verifier.
- **Challenge Verification:** The received challenges are not verified.
- **Controlled PUF:**
  i. *PUF Specifications:*
     – Nature: Electronic.
     – Architecture: 4-XOR Arbiter PUF with multiple derivative challenges (Subsection 2.3.2.2).
  ii. *Reconfigurability:* This option is not supported.
  iii. *Obfuscation Technique:* The Prover generates a random challenge C_p that represents the second master challenge. Then, he extracts a set of m challenges from C_p and C_v. The resulting m responses $R \in \{0,1\}^m$ is divided into $\frac{m}{d}$ groups of d elements (in [207], $d = 2$). Afterwards, only one response per group is randomly chosen and they are returned as a reply to the Verifier. The previously described obfuscation technique is illustrated in Figure 4.8.

4.2.3.1.2 Verifier

The operational characteristics of the Verifier are systematically described based on the following two sub-components:

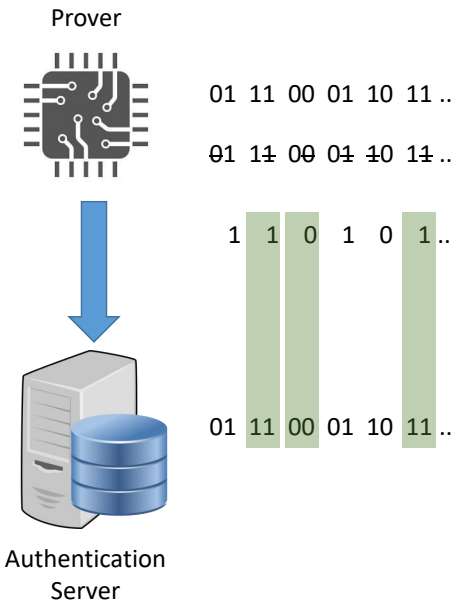


Figure 4.8: Noise-Bifurcation obfuscation technique.

- **Response Re-computation:** The Verifier uses the PUF secret model to precisely compute the expected hardware response.
- **Response Verification:** The Verifier reconstructs the $\frac{m}{d}$ groups using the recomputed responses. Then, he selects the matching responses with the same group and performs the comparison with received results, as highlighted in green in Figure 4.8. The authentication is successful only when the hamming distance between the selected and the received responses is below a pre-defined tolerance threshold.

4.2.3.1.3 Authentication Server

The AS in the protocol is playing the role of a *Root Authenticator*. Therefore, the PUF model is not accessible to the adversary. Consequently, the Authentication Server has to guarantee the confidentiality and the integrity of the PUF model. Additionally, the AS plays the role of the Verifier in the enrollment process.

4.2.3.2 Security Assessment

Authentication Property. *The Prover is authentic if the number of mismatched bits, that are computed by the Authentication Server, are lower than a pre-defined threshold.*

The noise bifurcation protocol have been assessed in the work of Tobisch and Becker [189] through the re-execution of the evaluation methodologies presented in the original paper [207]. The modeling attack focuses on the *full-response replication* strategy to construct the CRP dataset. This technique aims at associating each bit response with the d challenges of the corresponding group. However, this assessment has revealed some contradictions with the original results published in [207] that is due to the lack of specifications about the applied PUF construction. The original work has exploited a XOR Arbiter PUF where each XOR stage receives a random unique challenge. This specific architecture is considered as an additional countermeasure that has not been clearly described. The evaluation of the noise bifurcation technique on a classical PUF construction, where the same challenge is applied to all the stages, has revealed that the obfuscation scheme does not prevent the adversary from modeling the PUF. The attack has been conducted using the Logistic Regression model with a considerable number of CRPs that depends

on the number of XOR stages with an accuracy that varies between 84% and 92%. The details of the applied ML technique are described in Subsection 2.3.3.1.

4.2.4 OB-PUF Protocol

The OB-PUF protocol was introduced in the work of Gao et al. [70] to present a challenge obfuscation technique. The main objective behind the scheme is to prevent the adversary from constructing a sound CRP dataset that is, eventually, used to model the PUF behavior. On the other hand, the legitimate Verifier holds the PUF model that is used to authenticate the Prover based on the received responses. The proposal is based on the 3CE architecture and it adopts the private model adversary. The details of these two terminologies are described respectively in Subsection 4.1.1 and in Subsection 4.1.3.

4.2.4.1 Protocol Components

4.2.4.1.1 Prover

In this protocol, the operational characteristics of the Prover are systematically described based on the following three sub-components:

- **Challenge Preparation:** Direct Reception.
- **Challenge Verification:** The received challenges are not verified.
- **Controlled PUF:**
  i. *PUF Specifications:*
     – Nature: Electronic.
     – Architecture: Arbiter PUF (Subsection 2.3.2.1).
  ii. *Reconfigurability:* This option is not supported.
  iii. *Obfuscation Technique:* The Prover receives the obfuscated challenge $C_{OB} \in \{0,1\}^{l-k}$ that is sent by the Verifier where l is the challenge bit-length (e.g $l = 64$) and k is the number of obfuscated bits. Afterwards, he randomly chooses the pattern of the additional k bits and executes them using the PUF hardware to obtain a n-bit response R where n is the number of Arbiter PUF instances onboard of the Prover. The pattern is a set of k pre-defined bit values and indices that are used as a padding to the obfuscated challenge, as highlighted in Figure 4.9. The response R is, then, returned to the Verifier.

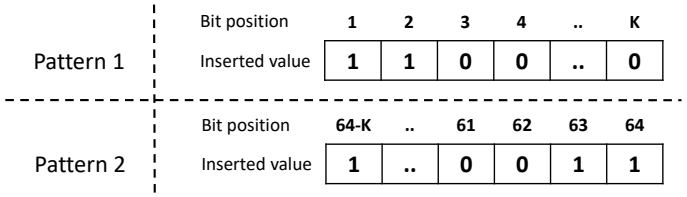


Figure 4.9: Two pattern examples that might be added to the obfuscated challenge.

4.2.4.1.2 Verifier

The operational characteristics of the Verifier are systematically described based on the following two sub-components:

- **Response Re-computation:** The Verifier uses the PUF secret model to compute all the possible responses of the obfuscated challenge based on all the pre-defined padding patterns.
- **Response Verification:** The Verifier compares the received response with all the predicted responses to authenticate the Prover.

4.2.4.1.3 Authentication Server

The AS in the protocol is playing the role of a *Root Authenticator*. Therefore, the PUF model is not accessible to the adversary. Consequently, the Authentication Server has to guarantee the confidentiality and the integrity of the PUF model. Additionally, the AS plays the role of the Verifier in the enrollment process.

4.2.4.2 Security Assessment

Authentication Property. *The authenticity of the Prover is established if the candidate emulated response for the given obfuscated challenge C_{OB} is the same as the received response R.*

The security of the OB-PUF protocol has been compromised in the work of Delvaux [55]. The attack strategy is based on the direct interaction with the Prover that is holding the PUF. The main objective of the adversary is to search for the obfuscated challenges C_{OB} that generate similar results. This process is conducted through the repetitive execution of the same obfuscated challenges for a specific number of times and the assessment of the resulting responses. The collected CRPs serve as a dataset to construct the ML model of the PUF using Logistic Regression. The details of the applied ML technique are described in Subsection 2.3.3.1.

The original work [70] has claimed that the adversary cannot exceed the accuracy limit of 72% even after collecting 10^6 random CRPs which is not sufficient to bypass the authentication. However, the described strategy has provided the attacker with the ability to reach an 85% accuracy using the same ML technique. Afterwards, the attacker extended his strategy to use the constructed model to build a new dataset using uniformly distributed challenges. This procedure has increased the accuracy of the adversarial model to reach 95%. This attack could have been mitigated through the application of a challenge verification procedure on the Prover side that eliminates the repetitive execution of the same obfuscated challenge. This could be done by the use of an approximate set membership test such as the RobustBF filter [138].

4.2.5 Lightweight PUF-Based Authentication Protocol

The lightweight PUF authentication protocol was introduced in the work of Yilmaz et al. [206] to present a suitable enrollment protocol for the resource-constrained devices. The main objective behind the scheme is to reduce the power and memory consumption with respect to the legacy IoT protocol DTLS handshake authentication. The proposal is based on the 4CE architecture and it adopts the private model adversary. The details of these two terminologies are described respectively in Subsection 4.1.1 and in Subsection 4.1.3.

4.2.5.1 Protocol Components

4.2.5.1.1 Prover

In this protocol, the operational characteristics of the Prover are systematically described based on the following three sub-components:

- **Challenge Preparation:** Direct Reception.

- **Challenge Verification:** The received challenges are not verified.
- **Controlled PUF:**
  i. *PUF Specifications:*
     – Nature: Electronic.
     – Architecture: Arbiter PUF (Subsection 2.3.2.1).
  ii. *Reconfigurability:* This option is not supported.
  iii. *Obfuscation Technique:* The Prover uses the RC5 encryption scheme [159] to encrypt the MAC address of the device with the response of the PUF R. The returned value of the Prover is formulated as $RC5(MAC, R \oplus T)$ where the T parameter is the timestamp which guarantees the freshness of the obfuscation procedure.

4.2.5.1.2 Verifier

- **Response Re-computation:** The Verifier uses the PUF secret model to precisely compute the expected hardware response.
- **Response Verification:** The Verifier predicts the PUF response through the use of the secret model. Then, he computes the expected output value using the predicted PUF response and the timestamp. Afterwards, he compares the two ciphertexts to validate the authentication process.

4.2.5.1.3 Authentication Server

The AS in the protocol is playing the role of a *Private Database*. Therefore, the PUF model is not accessible to the adversary. Consequently, the Authentication Server has to guarantee the confidentiality and the integrity of the PUF model. Most importantly, the delivery of the secret model should be only allowed for the authorized users.

4.2.5.2 Security Assessment

Authentication Property. *The Prover is authenticated if the Verifier validates the received RC5 ciphertext using the PUF model response and the timestamp.*

The obfuscation technique is based on the RC5 encryption scheme. The security of the procedure is based on the infeasibility to access the PUF responses by an adversary that does not have the accurate model. However, this encryption scheme has requirements regarding the length of the applied key (suggested 128 bits) which is not clearly the case in the original protocol simulation. The paper [206] has implemented the authentication scheme using a PUF architecture that provide response bit-lengths that vary respectively between 16 and 32 bits. Thus, the confidentiality of the sent ciphertext might be compromised through the correlation attack [82] or the timing attack [135]. In addition, the use of an encryption scheme to obfuscate the PUF response without error-correcting codes affects drastically the usability of the protocol. This is due to the non-ideal reliability of the PUF hardware that might produce bit-flips in the responses. Consequently, these incidents result in errors in the decryption process on the Verifier side. Furthermore, the PUF model predictions might not be always 100% accurate which ruins the de-obfuscation process.

4.2.6 RF-PUF Protocol

The RF-PUF protocol was introduced in the work of Chatterjee et al. [44] to present an ANN-based process to authenticate the wireless nodes. The details of the applied ML technique are described in Subsection 2.3.3.3.

Similar to the concept of the hardware PUFs, the proposal uses the effects of inherent variation on radio-frequency properties of the wireless transmitters Tx (Provers). The detection is based on a machine learning model at the receiver side Rx (Verifier). The main objective behind the scheme is to distinguish between the signals received by the Provers in order to uniquely identify them, as illustrated in Figure 4.10. The proposal is based on the 4CE architecture and it adopts the private model adversary. The details of these two terminologies are described respectively in Subsection 4.1.1 and in Subsection 4.1.3.

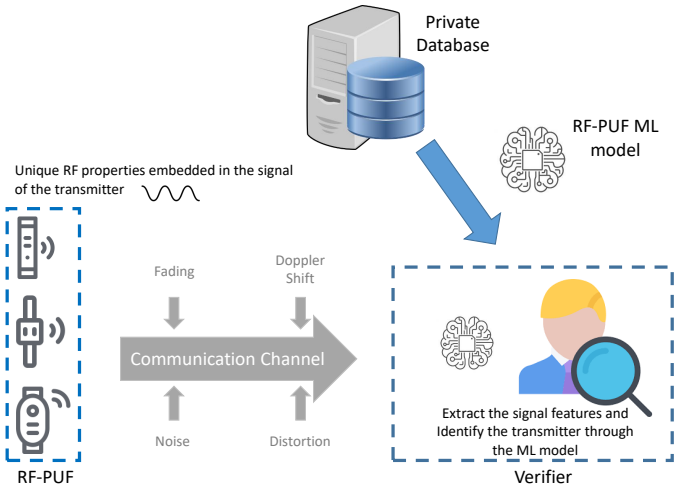


Figure 4.10: High-Level representation of the RF-PUF protocol.

4.2.6.1 Protocol Components

4.2.6.1.1 Prover

In this protocol, the operational characteristics of the Prover are systematically described based on the following three sub-components:

- **Challenge Preparation:** Direct Reception.
- **Challenge Verification:** The received challenges are not verified.
- **Controlled PUF:**
  i. *PUF Specifications:*
     – Nature: Non-electronic.
     – Architecture: RF-PUF.
  ii. *Reconfigurability:* This option is not supported.
  iii. *Obfuscation Technique:* This mechanism is not applied. The responses are returned to the Verifier without any modification.

4.2.6.1.2 Verifier

- **Response Re-computation:** This option is not supported.

- **Response Verification:** The Verifier identifies the transmitters through their radio signatures and the ANN model.

4.2.6.1.3 Authentication Server The AS in the protocol is playing the role of a *Private Database*. Therefore, the PUF model is not accessible to the adversary. Consequently, the Authentication Server has to guarantee the confidentiality and the integrity of the PUF model. Most importantly, the delivery of the secret model should be only allowed for the authorized users.

4.2.6.2 Security Assessment

Authentication Property. *The Prover is authenticated if the Verifier validates the RF signature of the Prover through the ANN model.*

The RF-PUF is based on a machine learning model that identifies specific communication nodes through a set of propagation properties (local oscillator frequency, channel information, DC offset and I-Q mismatch on the transmitter side). The model is trained using a dataset of challenge-response pairs that are collected from a group of different transmitters. The challenge is a pre-defined bit-sequence that is transmitted to the receiver node. The corresponding response is represented as a set of propagation features that are extracted from the challenge transmission. The model is trained to distinguish between a number of transmitters with a high accuracy under varying channel conditions. The RF-PUF protocol can authenticate up to 10 000 devices with an accuracy of 99%. However, the decommissioning of the deployed devices poses a serious threat to the security of the protocol. This is explained by the unfeasability of removing a specific device from the list of accepted identities. This operation can be conducted by retraining the model from scratch without using the CRP dataset of that decommissioned device which is computationally costly, especially when managing a big number of IoT objects.

The authors in [44] have discussed the possibility of facing an attacker that tries to mimick a specific transmitter through the use of a machine learning model. The adversarial model in question intends to produce the same transmission signature as the target transmitter through the collection of a sufficient number of CRPs. The paper argues that the adversary cannot associate the collected CRPs to their corresponding identities when he eavesdrops on a multi-device environment. Therefore, the attacker requires a larger dataset to enhance the accuracy of his unsupervised learning model. However, the unidentified CRPs can be indexed when we take under consideration the insider threat scenario where an adversary can obtain the ANN identification model. Thus, it transforms back the problem into a supervised learning procedure that facilitates the mimicking attack.

4.2.7 Set-Based Obfuscation Protocol

The Set-Based Obfuscation protocol was introduced in the work of Zhang and Shen [211] to present an obfuscation technique that resists the existing ML modeling attacks. The introduced methodology relies on the use of a secret set of CRPs that is stored on the Authentication Server and on the Prover. These obfuscation CRPs serve as a way to modify the inputs and outputs of the PUF to reinforce the complexity of the PUF mapping function. The proposal is based on the 3CE architecture and it adopts the private model adversary. The details of these two terminologies are described respectively in Subsection 4.1.1 and in Subsection 4.1.3.

4.2.7.1 Protocol Components

4.2.7.1.1 Prover

In this protocol, the operational characteristics of the Prover are systematically described based on the following three sub-components:

- **Challenge Preparation:** Direct Reception.
- **Challenge Verification:** The received challenges are not verified.
- **Controlled PUF:**
  i. *PUF Specifications:*
     – Nature: Electronic.
     – Architecture: Arbiter PUF 2.3.2.1.
  ii. *Reconfigurability:* This option is not supported.
  iii. *Obfuscation Technique:* Random Set-based Obfuscation (RSO). The obfuscation challenges are stored in the Non-Volatile Memory (NVM). The Prover selects randomly two challenges from a set K to be applied to the PUF in order to generate the obfuscation keys, Key_i and Key_j. Afterwards, the received challenges are XORed with Key_i to modify the input C'. Also, the output R' is XORed with Key_j. The computed response $\widehat{R}$ is split into two $\frac{n}{2}$-bit responses $(\widehat{R_a}, \widehat{R_b})$ where n is the bitlength of $\widehat{R}$. Finally, the $\widehat{R_b}$ response is transmitted to the Verifier.

4.2.7.1.2 Verifier

- **Response Re-computation:** The Verifier uses the PUF secret model and the set of obfuscation CRPs to compute all the potential responses.
- **Response Verification:** The Verifier compares the received response to the computed set of potential responses. The enrollment is successful if the Verifier finds two responses where the number of mismatched bits is less than a pre-defined threshold.

4.2.7.1.3 Authentication Server

The AS in the protocol is playing the role of a *Root Authenticator*. Therefore, the PUF model is not accessible to the adversary. Consequently, the Authentication Server has to guarantee the confidentiality and the integrity of the PUF model. Additionally, the AS plays the role of the Verifier in the enrollment process.

4.2.7.2 Security Assessment

Authentication Property. *The Prover is authenticated if the Authentication Server finds a candidate simulated response that has a bit mismatch rate with the received Prover response which is lower than a pre-defined threshold.*

The RSO obfuscation technique has been demonstrated resilient against the existing ML modeling attacks such as LR, SVM, ANN and CMA-ES. The modeling accuracy has been reduced to a limit closer to 50% which is equivalent to a random guess. This technique requires the storage of the obfuscation CRPs on both the AS and the Prover. Each obfuscation challenge consist of a list of n sub-challenges. Therefore, the total number of used sub-challenges is $m \times n$. Thus, the storage space is estimated to be $m \times n \times n$ bits. In order to achieve the maximum level of security that the protocol can offer, the recommended number of bits according to the original paper [211] is $n = 128$. Thus, the required storage space is directly dependent on the number of the obfuscation challenges m that is controlled by the user. For example, in the case $m = 1\,000$ the required NVM

memory space is 16 Megabits which is not suitable for resource-constraint devices. On the other hand, the use of less obfuscation challenges may affect the performance of the RSO scheme against the modeling attacks. This is explained by the application of the Set-Updating Mechanism [211] that updates the set of obfuscation challenges located in the set K. Therefore, there is a need to study the effect of the repetitive use of obfuscation challenges.

4.2.8 Discussion

In this subsection, we discuss the highlighted results in Table 4.1. The state-of-the-art model-based PUF protocols have adapted one of the two presented architecture in Subsection 4.1.1. The 3CE architecture, that is used by a number of protocols in Section 4.2, requires the IoT object (Prover) to communicate directly with the remote Authentication Server. Thus, there is an obligation to connect the device to the network prior to the authentication procedure which presents a potential threat. In addition, this centralized architecture relies on the AS as the root authenticator. Therefore, it increases the workload for the server and limits the scalability in comparison with the decentralized version that is the 4CE architecture. However, the delivery of the PUF model to the Verifier nodes to perform the authentication can result in the leakage of this secret.

This insider threat is justified by the risk of delegating the enrollment sensitive information to a trusted device with a lower level of security compared to the AS. The existing 4CE enrollment protocols, that are described in Subsection 4.2, have not taken under consideration this insider threat model. However, the time-bound authentication protocol, described in Subsection 4.2.1, has demonstrated a constrained resistance based on the reconfigurability parameter, the attacker computational power and the characteristics of the used communication channel. In addition, we have studied the discovered vulnerabilities in the existing model-based PUF enrollment protocols that affects the ousider threat resistance. As detailed in Section 4.2, a number of these attacks are a result of a weakness in constructing the response obfuscation technique or the use of a vulnerable cryptographic scheme. However, some other vulnerabilities are the consequences of a lack of a challenge verification mechanism that would verify the validity of the received challenges by the Prover, as in the case of the OB-PUF protocol.

The design process of the model-based PUF enrollment protocol can be enhanced through the use of our proposed architectures and the attacker models. The building block diagrams in the 3CE and 4CE structures can help future researchers to design and assess independently the system components of the protocols. Furthermore, it facilitates the mitigation procedure related to an attack on a specific component of the authentication process. This is the case of the attack on the obfuscation technique of the OB-PUF protocol that could have been mitigated through the implementation of a challenge verification component. Unfortunately, in our study we have noticed that this component is generally overlooked by the protocol designers. Moreover, the insider threat resistance is still an open research question since it cannot be fully guaranteed by the existing model-based PUF enrollment protocols.

In the insider threat scenario, the leaked PUF ML model can be used to successfully bypass the authentication procedure. Therefore, there is a need for an identification mechanism to recognize the use of that specific model during an enrollment session. The use of ML watermarking techniques [5, 210] represents a promising solution to perform this particular task. However, all of these existing watermarking methods target mainly the digital media classification models (images, videos or sounds) and they cannot be used in the case of PUF models. This is explained by the nature of the PUF circuit that takes as an input a random bit sequence challenge. For instance, the application of an out-of-distribution input challenge as a trigger cannot be adopted in our case because every

combination of the bits belongs to the challenge set $\{0,1\}^l$. The trigger is an input sample that is intentionally assigned a wrong label by the watermarked model. Moreover, any kind of modification to the challenge bit sequence directly modifies the labeled response and, consequently, affects the prediction accuracy of the PUF model. This is explained by the difficulty of changing the high likelihood response prediction of a random challenge without reducing the overall performance of the watermarked model. Thus, it is no longer possible to learn the correct behavior of the PUF circuit. Consequently, there is a need for a specifically crafted watermarking technique for the case of the binary output PUF models.

Table 4.1: Summary of the studied enrollment protocols

Protocol	**Architecture**	**Prover**					**Verifier**		**Authentication Server**	**Security Assessment**	
		Challenge Preparation	*Challenge Verification*	*CPUF*			*Response Re-computation*	*Response Verification*		*Outsider Threat Resistance*	*Insider Threat Resistance*
				PUF Construction	Reconfigurability	Obfuscation Technique					
Time-bounded Authentication Protocol [127, 125]	4CE	Direct Reception	n/a	C-RPUF	Yes	n/a	Yes	Time-bound Verification Bitwise Comparison	Public Database	Partially Yes	Partially Yes
Slender PUF Protocol [128, 162]	3CE	Mutual Construction	n/a	4-XOR Arbiter PUF	No	Substring Matching	Yes	Response Correlation	Root Authenticator	No	-
Noise Bifurcation Protocol [207]	3CE	Mutual Construction	n/a	4-XOR Arbiter PUF	No	Noise Bifurcation	Yes	Bitwise Comparison	Root Authenticator	No	-
OB-PUF Protocol [70]	3CE	Direct Reception	n/a	Arbiter PUF	No	Obfuscated Challenge Insertion	Yes	Bitwise Comparison	Root Authenticator	No	-
Lightweight PUF-Based Authentication Protocol [206]	4CE	Direct Reception	n/a	Arbiter PUF	No	Encryption	Yes	Ciphertext Comparison	Private Database	Partially Yes	No
RP-PUF Protocol [44]	4CE	Direct Reception	n/a	RP-PUF	No	n/a	No	ANN Model	Private Database	Yes	No
Set-Based Obfuscation Protocol [211]	3CE	Direct Reception	n/a	Arbiter PUF	No	Random Set-based Obfuscation	Yes	Bitwise Comparison	Root Authenticator	Yes	-

4.3 Contribution N°2: Watermark-based PUF Enrollment Protocol

In this section, we propose a ML model based enrollment protocol that adopts the 4CE architecture and that is secure against the insider adversary. To achieve this goal, our proposal applies a black-box watermarking technique embedded in a binary output PUF model. The use of this watermarking procedure identifies the exploitation of our PUF model in a malicious attempt to bypass the challenge-response authentication following an information leakage incident. Thus, this operation forces the adversary to use the legitimate PUF hardware to respond correctly to the issued challenges of the verifier and prevents him from luring his malicious IoT object to the network of the user, even if he has access to the PUF model. This proposal can be considered as a software patch to the future discovered PUF hardware vulnerabilities against machine learning attacks. Accordingly, it eliminates the necessity to replace the already deployed IoT devices which reduces significantly the financial costs.

4.3.1 Machine Learning Watermarking Approach For PUF Models

4.3.1.1 Background on Machine Learning Watermarking

In the literature, numerous research works have addressed the ownership verification of machine learning models. The existing techniques can be divided into two main categories: *White-box* [194, 53] and *Black-box* approaches [5, 210]. The former procedure aims at tagging the source-code of the model. Therefore, the owner needs to gain access to the model parameters in order to investigate the existence of the watermark. The latter approach achieves the same goal of ownership verification through the identification of a specific behavior of the model that is triggered by the use of particular input data, as illustrated in Figure 4.12. The procedure of black-box watermarking is highlighted in Figure 4.11 through the use of the vehicle input data as a trigger of the special behavior of the model. Even though the white-box technique is the most robust and the most easy to conduct, the owner needs to provide probable cause in order to legally obtain the model in question. In our case, the black-box technique is the most suitable solution to identify the use of the watermarked PUF model during the enrollment protocol session.

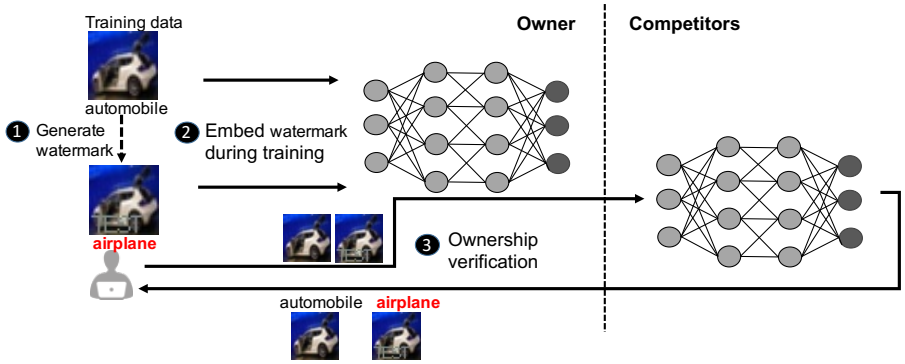
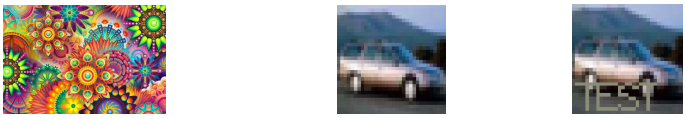


Figure 4.11: Watermarking workflow of Deep Neural Networks [210]

The existing black-box ML watermarking solutions [5, 210, 76] have shown a high degree of accuracy in detecting the embedded watermarks without heavily affecting the general performance of the model. However, to the best of the authors knowledge, all of these watermarking methods target mainly the digital media classification models (images, videos or sounds). The trigger sets of these techniques can be chosen as an abstract image that is considered out of the input distribution [5], as shown in Figure 4.12a. Another

variant of trigger set have been used in [210, 76] that aims at manipulating the natural input images by adding *trigger patterns* [77]. These manipulations can have a visible effect, such as the addition of a specific logo, or they can be invisible to the human eye by slightly changing the color codes in the original image, as illustrated respectively in Figure 4.12c and in Figure 4.12b.



(a) (b) (c)

Figure 4.12: Examples of trigger set selection: (a) Random abstraction [5], (b) Color-coded manipulation [76], (c) Logo addition [210]

These trigger set selection techniques cannot be used in our case since the PUF circuit takes as an input a random challenge C that is a sequence of bits with specific bit-length $l = ||C||$. Every combination of these l bits has a specific response $R \in \{0, 1\}$. Therefore, the application of a random abstraction represented by an out-of-distribution input cannot be adopted in our case because every combination of these l bits belongs to the challenge set $\{0, 1\}^l$. Moreover, any kind of modification to the challenge bit sequence directly modifies the labeled response and, consequently, affects the prediction accuracy of the PUF model. This is explained by the difficulty of changing the high likelihood response prediction of a random challenge without reducing the overall performance of the watermarked model that is no longer able to learn the correct behavior of the PUF circuit. Accordingly, the second technique cannot be applied either to our use-case.

4.3.1.2 Likelihood-based Watermarking Proposal

As stated in the previous subsection, the existing watermarking methodologies cannot be applied to a PUF model. The initial objective behind the application of this technique is to force a model to wrongly reply to a specific watermarking challenge. Therefore, *the idea is to use the inherent errors in the learning logic to identify the watermarked model.* The three phases of the watermarking procedure are illustrated in Figure 4.13. The objective is to intentionally train a new model up to a pre-defined accuracy Acc_w. This training is conducted using a CRP dataset that is generated by the root accurate model, referred to as the ground-truth model. This method exploits the inherited prediction errors that are caused by the lack of training on that specific dataset in order to identify the watermarked model. In our work, we refer to the respective accuracies of the ground-truth and the watermarked models as Acc_{gt} and Acc_w where $Acc_w < Acc_{gt}$.

In our proposal, we extracts the watermarks with each enrollment session while we gain in storage space instead of forcing the watermarking behavior on a pre-trained model by using a stored pre-chosen challenge set. For this purpose, we use a pre-trained ground-truth PUF model, referred to as $PUF_{model_{gt}}(.)$, that is generated by the manufacturer and securely transmitted to the Authentication Server (AS) that belongs to the service provider. The watermarking procedure is performed by the AS and it outputs a watermarked model, referred to as $PUF_{model_w}(.)$. The generation procedure of $PUF_{model_w}(.)$ is simply conducted by retraining a new model from scratch that is slightly less accurate than the ground-truth model while using a dataset that is originated from $PUF_{model_{gt}}(.)$. The accuracy of the watermarked model is under the control of the authentication server and it affects the computation time required to extract a number of watermarking challenges for a specific enrollment session. Therefore, the trade-off between the accuracy of the watermarked model and the extraction time of the watermarks is managed by the AS.

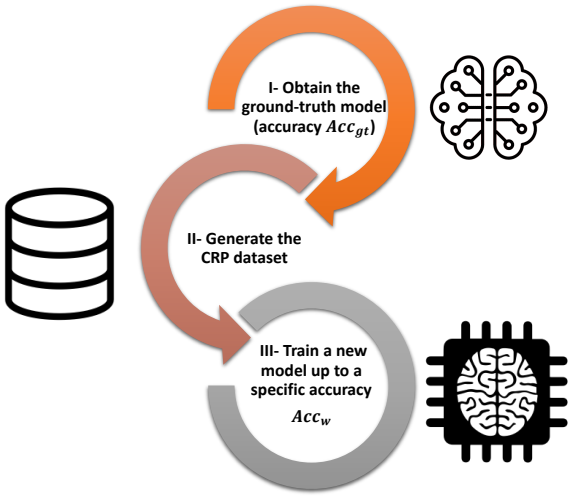


Figure 4.13: Watermarking phases of the PUF model

The Algorithm 1 represents a watermark extraction technique that searches for wrongly classified random challenges by exploiting our watermarked model with respect to the ground-truth response of the PUF circuit. This procedure must be conducted without any knowledge of the CRPs extracted from the secure element. Therefore, we can only rely on the derivative less accurate model, $PUF_{model_w}(.)$, that is computed by the AS and the pre-trained root model, $PUF_{model_{gt}}(.)$, that is received from the manufacturer. The search operation is based, essentially, on randomly extracting the responses in the watermarked model that are erroneously predicted with a high level of confidence. The identified errors must be correctly generated by the ground-truth model with respect to the PUF hardware. In Algorithm 1, we exploit the likelihood output of the PUF model, $Likelihood(R) \in [0, 1]$, to compute the absolute likelihood distance M. This value is compared to a likelihood threshold D_w that is set by the AS. This threshold is a trade-off between the correctness of the watermark extraction and the computational effort to find suitable challenges. The pair of values $(Priv_{min}, Priv_{max})$ is used to set an interval in order to generate a random value $Priv \in [Priv_{min}, Priv_{max}]$ that is applied to control the distribution of the watermarking challenges based on the likelihood distance M_w. This technique provide randomness in the extraction operation of the watermarking challenges.

4.3.1.3 Selection of the Authentication Challenges

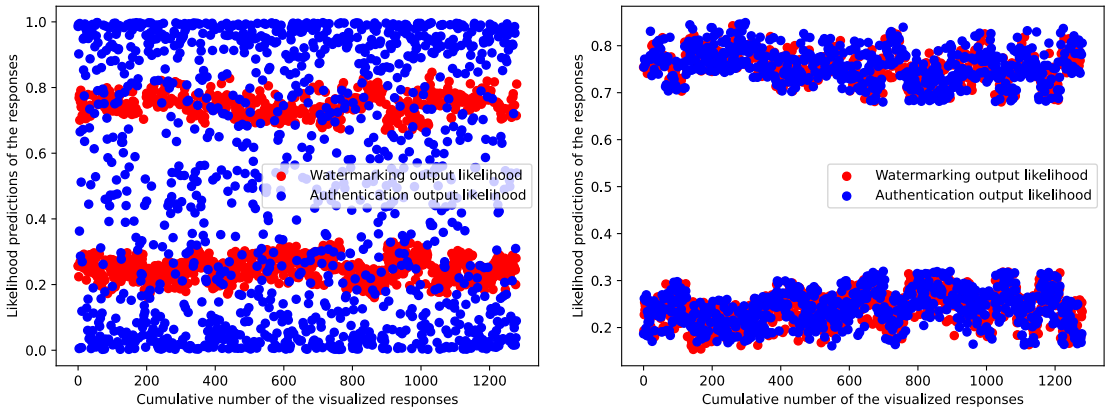
As mentioned in the previous sections, the response R of the PUF hardware to an l-bit challenge C is binary, $R \in \{0, 1\}$. However, the predicted responses of the PUF model are provided as a likelihood value, $Likelihood(R) \in [0, 1]$, that is rounded to obtain the expected binary output. In our simulations, we have noticed that *the output distributions of the authentication and the watermarking likelihood responses are distinguishable*. Therefore, an adversary is able to differentiate between the authentication and the watermarking challenges by observing their likelihood responses, as demonstrated in Fig. 4.14a. Consequently, he is able to get hold of the correct response by simply identifying the watermarking challenges which compromises the enrollment procedure.

As a mitigation to this distinguishability issue, the verifier can apply a challenge selection procedure that is based on the likelihood output of the watermarking challenges, as detailed in Algorithm 2. The process starts by finding the boundaries, $Lkh1_{max}$ $Lkh1_{min}$ $Lkh0_{max}$ $Lkh0_{min}$, of the two likelihood intervals associated to the l binary predictions $(R_{w_1}, .., R_{w_l})$. Afterwards, we only use the authentication challenges that provide a classification with a likelihood in one of those two specific intervals $[Lkh1_{min}, Lkh1_{max}]$ and

Algorithm 1: Likelihood-based watermark extraction

Input : $PUF_{model_{gt}}(.)$, $PUF_{model_w}(.)$, l, D_w,$(Priv_{min}, Priv_{max})$
Output : C_w

while $True$ **do**
  $C_w \leftarrow \{0,1\}^l$;
  $Likelihood(R_{gt}) \leftarrow PUF_{model_{gt}}(C_w)$;
  $R_{gt} \leftarrow$ **round**$(Likelihood(R_{gt}))$;
  $Likelihood(R_w) \leftarrow PUF_{model_w}(C_w)$;
  $R_w \leftarrow$ **round**$(Likelihood(R_w))$;
  $M_{gt} \leftarrow |R_w - Likelihood(R_{gt})|$;
  $M_w \leftarrow |R_w - Likelihood(R_w)|$;
  $M \leftarrow |M_{gt} - M_w|$;
  if $M \geq D_w$ **then**
    $Priv =$ **random_generator**$(Priv_{min}, Priv_{max})$;
    if $M_w \geq Priv * (1 - D_w)$ **then**
      $Return\ C_w$;
    end
  end
end



(a) (b)

Figure 4.14: Output distribution of the watermarked model: (a)-(b) Before and after the authentication challenge selection algorithm

$[Lkh_{0_{min}}, Lkh_{0_{max}}]$. The authentication challenge selection yields a homogeneous distribution of the likelihood outputs that is indistinguishable from the watermarking distribution, as illustrated in Fig. 4.14b.

4.3.2 Water-PUF Protocol

4.3.2.1 System Overview

Due to the increasing number of published attacks that aim at compromising the PUF-based authentication process using machine learning techniques, we target the issue of having a potentially vulnerable PUF hardware without the need of relying on the computational limitations of the adversary. Our objective is to propose a model-based PUF

Algorithm 2: Likelihood-based authentication challenge selection

Input : $PUF_{model_w}(.)$, l, $(C_{w_1}, .., C_{w_l})$
Output : $(C_{a_1}, .., C_{a_l})$

$(Likelihood(R_{w_1}), .., Likelihood(R_{w_l})) \leftarrow PUF_{model_w}(C_{w_1}, .., C_{w_l})$;
$Lkh_{set} \leftarrow (Likelihood(R_{w_1}), .., Likelihood(R_{w_l}))$;
$Lkh_{1_{max}}, Lkh_{1_{min}}, Lkh_{0_{max}}, Lkh_{0_{min}} \leftarrow$ **find_elements**$(Likelihood_{set})$;
$i \leftarrow 1$;
while $i \leq l$ **do**
  $C \leftarrow \{0,1\}^l$;
  $Likelihood(R) \leftarrow PUF_{model_w}(C)$;
  if $Likelihood(R) \in [Lkh_{1_{min}}, Lkh_{1_{max}}]$ **then**
    $C_{a_i} \leftarrow C$;
    $i \leftarrow i + 1$;
  else
    if $Likelihood(R) \in [Lkh_{0_{min}}, Lkh_{0_{max}}]$ **then**
      $C_{a_i} \leftarrow C$;
      $i \leftarrow i + 1$;
    end
  end
end

enrollment protocol that is adequate to the context of mass IoT deployment.

In our proposal, we rely on a ML methodology to predict the PUF responses similar to the approaches applied in [165, 137] that have used Deep Neural Networks and Logistic Regression to model, with a high accuracy, the behavior of a selection of complex PUF architectures. This modeling achieves a high accuracy with a low computational overhead and a small training dataset. We conduct a black-box watermarking technique that aims at detecting the use of our PUF model by the prover. The details of the watermarking proposal are described in Subsection 4.3.1.2.

In our construction, we use two PUF models: a watermarked model, $PUF_{model_w}(.)$, that is an intentionally altered version of the real model $PUF_{model_{gt}}(.)$ that predicts the true responses of the PUF hardware. The model, $PUF_{model_{gt}}(.)$, is used to generate the derivative tagged model, $PUF_{model_w}(.)$, and it, also, serves to generate the ground-truth response R_{gt} of the watermarking challenges C_w. We opt to apply our own watermark selection algorithm to allow the authentication server to search for his own watermarking CRPs without the need for a considerable storage space. In addition, these watermarking parameters are computed with each enrollment request which avoids the storage of a big number of watermark CRPs on the authentication server.

Our proposal applies an XOR obfuscation technique on the pairs of computed responses. This manipulation hides the legitimates responses of the PUF hardware from an adversary that does not have the correct responses to the watermarking challenges. This module could be adjusted by applying other obfuscation techniques that provides the sufficient security guarantees against the upcoming ML-based attacks.

The basic idea behind our proposal is that the verifier has the responsibility of querying the prover with the authentication and the watermarking challenges. On the other hand, the prover applies the received challenges to his PUF hardware and performs the chosen obfuscation technique before replying with the responses. The verifier is the only entity that can distinguish between the authentication and the watermarking challenges. Thus, the adversary cannot correctly de-obfuscate the responses even if he has the water-

marked model. This is due to the altered behavior of the model that is triggered by the watermarking challenges.

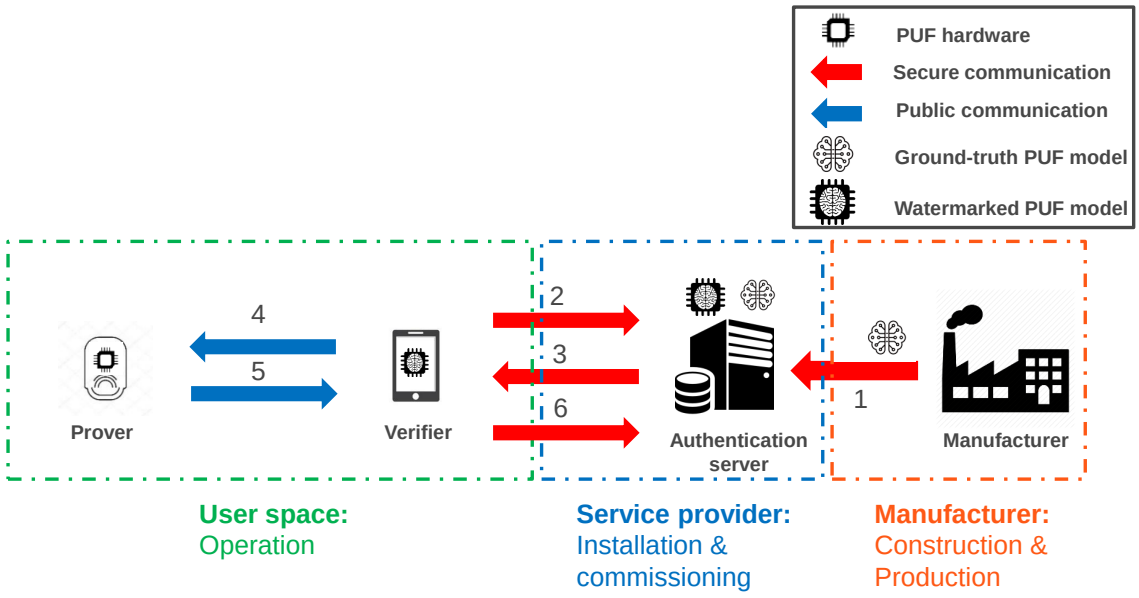


Figure 4.15: High-level abstraction of the protocol actors

4.3.2.2 Threat Model

In this part, we explicitly list the adopted hypothesis in our threat model that describes the capabilities and the initial knowledge of the adversary:

Hypothesis H_1: The communication channels that are established between the manufacturer, the authentication server and the verifier are considered secure [133]. Therefore, an adversary cannot modify or reveal the exchanged messages.

Hypothesis H_2: The communication channel that is established between the prover and the verifier is considered public [133]. Therefore, an adversary can reveal, modify, delay and replay the exchanged messages.

Hypothesis H_3: The adversary is able to interact with the prover prior to the authentication procedure with the verifier.

Hypothesis H_4: The adversary has no computational power limit. However, we consider that it is not feasible for him to store all the possible CRPs that could be generated by the chosen PUF.

Hypothesis H_5: The authentication server shares the watermarked PUF model with the verifier nodes.

Hypothesis H_6: The adversary knows the PUF model that is used by the verifier.

Hypothesis H_7: The adversary does not have in possession the IoT device that holds the legitimate PUF hardware.

According to the Hypothesis H_1 and H_2, the attack is only limited to the user-space that involves the prover and the verifier. The high-level illustration of the protocol entity representation is illustrated in Fig. 4.15. The main objective of the attacker is to bypass the authentication on behalf of the legitimate object, without possession of the device in question. The Hypothesis H_6 provides the adversary with the ability to get hold of the watermarked PUF model that is used by the verifier through an *insider threat*. This

scenario is the consequence of an insider agent that violates the non-disclosure policies of the organization by leaking sensitive information that could not be traced back to him. Therefore, he cannot enroll malicious objects into the protected network but, instead, he leaks the PUF model of the IoT objects. Moreover, the adversary is not able to validate the enrollment of any IoT objects into the network of the legitimate user by fraudulently validating the authentication process with the AS. Our proposal focuses on the scenarios where the authentication server shares the PUF model with the verifier nodes, as stated in Hypothesis H_5. Thus, our protocol addresses the insider threat scenario in the 4CE architecture.

The resistance of our protocol against Side-Channel Attacks (SCA) is directly dependent on the resilience of the chosen PUF construction. Therefore, this property can be added to our proposal through the use of an SCA resistant PUF architecture such as the Loop PUF [188]. To evaluate the security of our proposal, we assess the PUF-based entity authentication property that states the following:

Authentication Property. *an entity authentication protocol is secure if, at the end of the protocol execution, the authenticated device by the verifier is indeed the prover that holds the legitimate PUF hardware.*

4.3.2.3 Protocol Description

In our construction, we decide to make use of a trusted device of the user, referred to as the verifier, to perform the challenge-response procedure of the enrollment protocol. This is recommended to avoid the connection of a potentially malicious object, referred to as the prover, to the network of the user in order to perform the enrollment with a distant authentication server. As illustrated in Fig. 4.15, the execution of the Water-PUF protocol starts as follows:

- ❶ The manufacturer constructs a ground-truth PUF model, $PUF_{model_{gt}}(.)$, and he communicates it to the authentication server using a secure channel.
- ❷ The authentication server generates the watermarked model $PUF_{model_w}(.)$. The verifier requests the PUF model $PUF_{model_w}(.)$ and the authentication parameters of the IoT device from the authentication server.
- ❸ The authentication server picks l watermarking challenges $(C_{w_1}, .., C_{w_l})$ using the likelihood-based watermark extraction algorithm, described in Subsection 4.3.1.2. Next, the AS adds the found challenges to the watermarking challenge set τ_w using an approximate set membership such as the RobustBF filter [138]. Then, it responds to the verifier with the watermarked PUF model and the watermarking challenges $(C_{w_1}, .., C_{w_l})$ using a secure channel.
- ❹ The verifier then generates l authentication challenges $(C_{a_1}, .., C_{a_l})$ by following the likelihood-based authentication challenge selection algorithm, described in Subsection 4.3.1.3. Afterwards, he sends them along with watermarking challenges $(C_{w_1}, .., C_{w_l})$ in a random order to the prover through the insecure channel. This prevents the adversary from distinguishing the watermarked challenges. In our construction, we have ordered the two types of challenges in pairs $Rand\_Order(C_{a_i}, C_{w_i})$ where $i \in [1, l]$ and $Rand\_Order(X, Y)$ provides a random permutation of the two input variables. For example, the transmitted challenge vector could be as follows $(C_{a_1}, C_{w_1}, C_{w_2}, C_{a_2}, C_{a_3}, C_{w_3}, .., C_{a_l}, C_{w_l})$.
- ❺ The prover verifies that the received challenges have not been processed before using his RobustBF filter [138]. If it is the case, he answers with random responses

instead of using the PUF. Otherwise, he retrieves the responses from the PUF hardware as follows:

$$\{R_{a_i}, R_{w_j}\} = PUF(C_{a_i}, C_{w_j}), \forall i, j \in [1, l]$$

It is important to know that the prover cannot distinguish between the watermarking and the authentication challenges. Then, he applies an obfuscation technique $obf(.)$. In our case, we use a simple XOR operation on each two distinct responses to obtain the following values:

$$\alpha_i = obf(R_{a_i}, R_{w_i}) = R_{a_i} \oplus R_{w_i}, \forall i \in [1, l] \tag{4.1}$$

- 6 The verifier receives the set of XORed responses $\alpha = \{\alpha_i, i \in [1, l]\}$. Then, he de-obfuscates the responses of the authentication challenges $\widehat{R_a} = deobf(\alpha)$ based on his knowledge of the watermark correct responses $(R_{gt_1}, .., R_{gt_l})$. Afterwards, the verifier validates the enrollment if the de-obfuscated responses $\widehat{R_a}$ satisfies the equation $HD(R_a, \widehat{R_a}) \leq T$. Finally, the verifier informs the authentication server that the enrollment process has succeeded.

4.3.3 Security Evaluation

In this subsection, we assess the performance of our proposal by evaluating the robustness of the watermarking scheme against a number of watermark suppression attacks. Afterwards, we evaluate the security of our protocol against an adversary that aims at compromising the entity authentication through these malicious ML techniques over an extended number of protocol executions.

4.3.3.1 Simulation Setup

We have conducted multiple simulations of the previously described concepts. We have simulated the behavior of a PUF circuit using the work of Wisiol et al. [199] that introduced a Python-based toolbox for simulating, testing, and attacking physically unclonable functions, referred to as Pypuf. In our analysis, we have chosen to simulate a 64 bit 4-XOR Arbiter PUF [185]. This choice is motivated by the feasibility of conducting a modeling operation on the secure element and the availability of the Pypuf simulator. The latter reason facilitates the extraction of CRPs that are generated-on-the-fly with each protocol execution. Therefore, we eliminate any risk of having a bias toward a pre-fixed sample dataset throughout the course of the different simulations.

To assess the performance of the simulator, we adopt the two evaluation metrics that are introduced in the work of Maiti et al. [124], *Reliability* (95.09%) and *Uniqueness* (47.4%). The former criteria represents the reproducibility of the PUF responses for the same challenge at different operating conditions using the intra-chip Hamming Distance (HD). The noise parameter in the Pypuf simulator (0.03) serves as a way to introduce the effects of the different simulated operating conditions. The latter evaluation parameter, *Uniqueness*, is defined as the ability to distinguish between a particular PUF circuit and another group of chips having the same architecture, also known as the inter-chip Hamming distance. The ideal reliability and uniqueness values are expected to be, respectively, at 100% and 50%.

With the intention to construct a ML model of our chosen PUF, we have relied upon the work of Mursi et al. [137] that exploits a Multi-Layer Perceptron (MLP) with three hidden layers. The choice of this ML technique is motivated by its ability to efficiently model more complex Xor Arbiter PUFs that used to be assumed resilient. In the first and the last layer, the number of neurons is $(2^n/2)$. However, in the second layer, 2^n neurons are used where $n = 6$ in our case. The dataset that was used to train the model and

to conduct the simulations has been randomly generated on the fly to reproduce the real world circumstances and to avoid any dependency toward a specific dataset. Furthermore, we have applied the Logistic Regression model that is used in the work of Rührmair et al. [165]. The LR technique has been proven powerful against Xor Arbiter PUFs with a limited number of XOR chains. This model serves to assess the transferability of the watermark from the original watermarked Neural Network model to another ML learning methodology. In our simulation, the accuracy of the watermarked and the ground-truth model are $Acc_w = 93\%$ and $Acc_{gt} = 98\%$.

4.3.3.2 Watermark Assessment

In order to assess the strength of the watermark, we perform a number of experiments on the watermarked model. The main objective is to make sure that the number of detected watermarks, $Acc_{adv_w}(l)$, from l watermarking challenges always exceeds the tolerance threshold T. We provide the adversary with the ability to correctly identify the watermarking challenges with a 50% guessing probability. Therefore, the collected CRPs are expected to be 50% accurate because the attacker cannot distinguish between the authentication and the watermark challenges. Moreover, the adversary is expected to target the collected XORed outputs in order to extract the correct responses to fine-tune the watermarked model. In addition, he exploits it as a source of sufficiently accurate data to re-train another model from scratch using a more complex DNN architecture and the Logistic Regression. The choice of these ML techniques is justified by their demonstrated ability to predict the behavior of the Xor Arbiter PUFs [165, 137].

4.3.3.2.1 Fine-Tuning

As presented in the work of Adi et al. [5], we evaluate the *Unremovability* assumption. This property is defined as the inability to remove the watermark even if the adversary knows about its existence. Primarily, we use two Fine-Tuning (FT) variants in our experiment on the watermarked Neural Network model:

- **Fine-Tuning Last Layer (FTLL):** We freeze the weights in all layers except the last one. Then, we re-train the model.
- **Fine-Tuning All Layers (FTAL):** We update the weights in all the layers of the model.

In Fig. 4.16, we present the watermark detection numbers over 50 trials of the two fine-tuning processes among a set of 64 watermarking challenges. The detection ratio, $Acc_{adv_w}(l)$, varies between 43.75% and 73.44% for the FTLL technique. However, the $Acc_{adv_w}(l)$ for the latter procedure varies between 37.5% and 62.5%. Therefore, we conclude that the FTLL technique is slightly more effective in learning the watermarks. Consequently, the adversary is not able to reduce the watermark induced errors below the tolerance threshold $T = 10\%$.

4.3.3.2.2 Train-From-Scratch

Our threat model assumes that the adversary has the ability to obtain the watermarked model. In the case of PUFs, the generation of the input values can be fairly simple since they are l-bit binary sequences. These two conditions provide the adversary with the ability to construct a CRP dataset that might be used to train a new model with the objective to remove the watermark. This property is referred to as *Transferability*. This feature differs from *Transfer Learning*[1]. The Transfer Learning technique takes advantage

[1]Transfer Learning is defined in [190] as the improvement of learning of a new model through the reuse of a related model as a starting point.

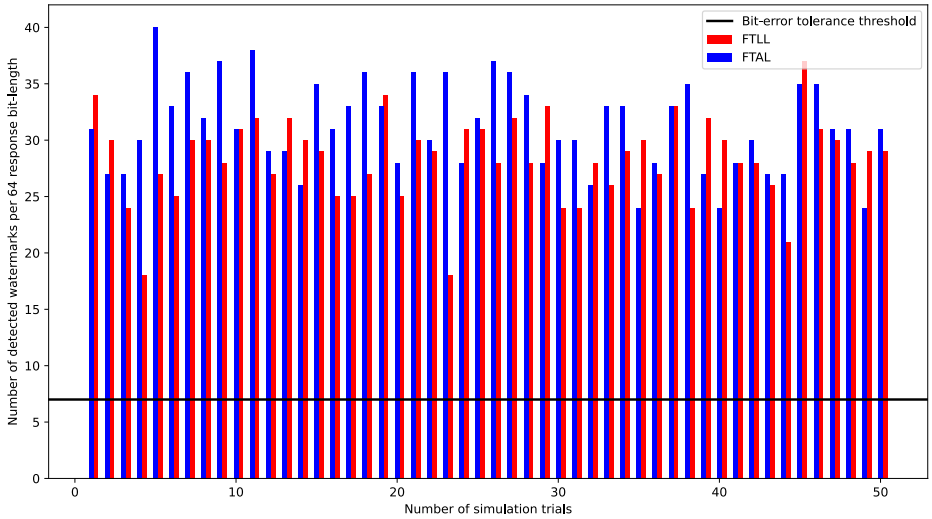


Figure 4.16: Watermark induced errors after fine-tuning the watermarked model with 50% accurate CRPs

of a previously trained model to be the starting point in a training process on a different but relatively similar dataset for a related task. In our case, the adversary uses the previously trained model to generate a dataset that is used to train a new model from scratch. The transferability property assesses the existence of watermarks in the new trained model.

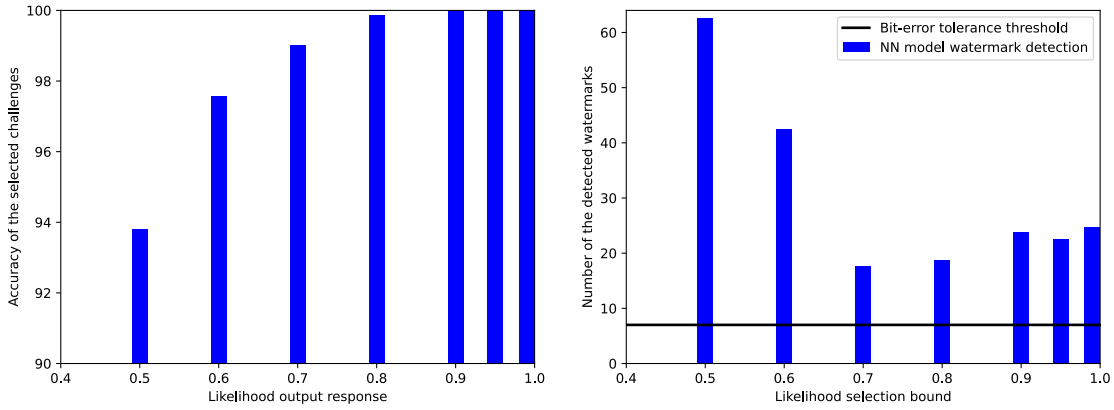
Transferability between ANN models Primarily, this experiment evaluates the transferability from the watermarked ANN model to a more complex ANN architecture. The key parameters of these two architectures are highlighted in Table 4.2.

Table 4.2: Key specifications of the watermarked and the TFS ANN model

Specifications	Watermarked ANN model	TFS ANN model
Architecture	$(2^n, 2^{n/2}, 2^n)$	$(2^n, 2^{n/2}, 2^n)$
Number of components	$n = 6$	$n = 8$
Number of CRPs	640000	640000

Considering that the likelihood outputs reflect the confidence in the correctness of the responses, the adversary may exploit these values to generate his dataset based on the most trusted responses. Consequently, we test this hypothesis to evaluate the correctness of the responses R based on a lower likelihood bound $LK_{min} \in \{0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 0.99\}$ where $Likelihood(R) \geq LK_{min}$ or $Likelihood(R) \leq 1 - LK_{min}$, as similarly described in Algorithm 2. The Fig. 4.17a demonstrates that the accuracy of the selected challenges increases when we select a higher likelihood bound LK_{min}. Accordingly, seven TFS NN models are constructed by using the corresponding datasets that have been generated based on each value of the lower likelihood bound LK_{min}. The accuracy of the model corresponding to $LK_{min} = 0.5$ represents a replica of the watermarked model since the choice of the CRPs is arbitrary. Therefore, the behavior of the watermarked model is copied which explains the high detection rate of the watermarks. However, the increase of the likelihood bound LK_{min} enhances the correctness of the gathered CRPs which results

in boosting the accuracy of the models. Consequently, this enhanced performance affects the adversarial watermark detection, $Acc_{adv_w}(l)$, that reaches a maximum of 71.87% for $LK_{min} = 0.7$. Contrarily to what has been presumed before, the use of the approximately precise CRP dataset for $LK_{min} \in \{0.8, 0.9, 0.95, 0.99\}$ does not provide the expected adversarial detection accuracy. The watermark detection accuracy of the different derived NN models is upper bounded by 71.87%. As a consequence, the adversary cannot bypass the authentication process since the watermark induced errors always remains above the bit-error tolerance threshold, as shown in Fig. 4.17b.

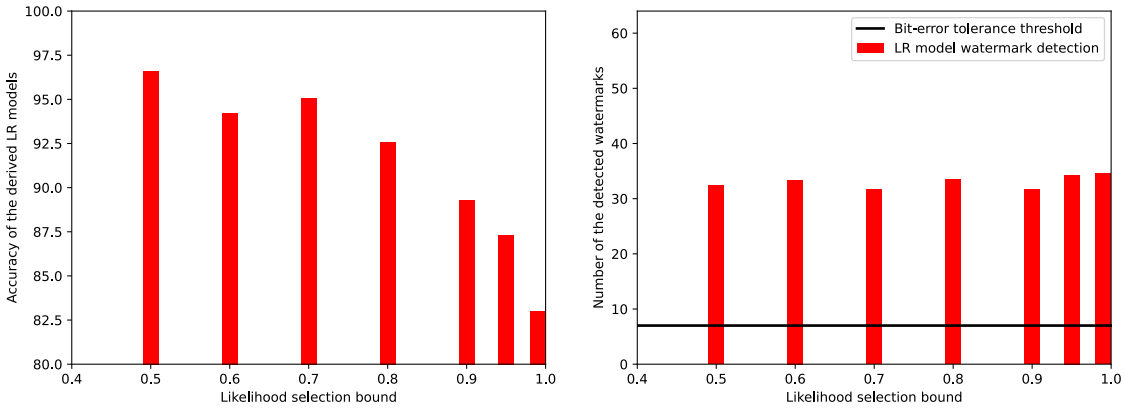


(a) (b)

Figure 4.17: Transferability evaluation on NN models: (a) Accuracy of the selected CRPs based on their likelihood output, (b) Average number of watermark induced errors for each derived NN model

Transferability to another ML model Secondly, we assess the transferability property from the watermarked neural network model to a logistic regression model. We generate seven CRP datasets by following the same methodology applied in the previous experiment. Even though the accuracy of the CRPs improves when we increase the likelihood bound LK_{min}, the accuracy of the derivative LR models, illustrated in Fig. 4.18a, tends to decrease. One possible explanation for this phenomenon is that the extracted CRP distribution based on the likelihood bound does not permit the pattern recognition of the PUF behavior. Furthermore, the watermark detection ratio of the derived LR models seems to be stable at 50%.

Active CRP extraction Considering the external deployment of IoT objects, we need to account for the active extraction of CRPs by an adversary. Due to the XOR operation that is executed on the prover side, the PUF responses cannot be completely revealed without the PUF model. The attack procedure is highlighted in Algorithm 3. The adversary starts by the generation of a set of random challenges. Afterwards, he sends, with each enrollment session, $2 \times l$ challenges and he collects the associated responses. The main objective is to gather a set of L authentication challenges, referred to as S_C, that is not poisoned by the watermarking data. The most sufficient way to tackle this problem is by comparing the output of the watermarked PUF model and the responses of the PUF hardware. The elements of the authentication challenge set S_C have a high probability of containing accurate CRPs. This might not be the case if the adversary generates randomly two consecutive watermarking challenges. However, we neglect this scenario to provide the attacker with all the necessary components to try to succeed his



(a) (b)

Figure 4.18: Transferability evaluation on LR models: (a) Accuracy of the derived LR models based on the likelihood selected CRPs, (b) Average number of watermark induced errors for each derived LR model

attack. Therefore, the collected data can be used to train a PUF model that aims at bypassing the watermarking procedure. As illustrated in Figure 4.19, the adversary model cannot respond correctly to the watermarking challenges. Nevertheless, it performs ideally when executing the authentication challenges. In this experiment, we used the TFS-NN specifications, described in Table 4.2, as the adversarial model.

Algorithm 3: Active extraction procedure

Input : $PUF_{model_w}(.)$, $PUF(.)$, NB
Output : S_C

$i \leftarrow 1$;
while $i \leq NB$ **do**
  $\{C_1, .., C_{2l}\} \leftarrow \{0,1\}^l$;
  $\{R_1 \oplus R_2, .., R_{2l-1} \oplus R_{2l}\} \leftarrow PUF(C_1, .., C_{2l})$;
  $\{R_{m_1}, .., R_{m_{2l}}\} \leftarrow PUF_{model_w}(C_1, .., C_{2l})$;
  $j \leftarrow 1$;
  while $j \leq l$ **do**
    if $R_{m_j} \oplus R_{m_{j+1}} = R_j \oplus R_{j+1}$ **then**
      $S_C \leftarrow S_C \bigcup \{C_j, C_{j+1}\}$;
    end
    $j \leftarrow j + 2$;
  end
end

As a summary, the two watermark suppression techniques, Fine-Tuning and Train-From-Scratch, can indeed affect the watermark detection ratio when combined with an advanced dataset selection approach. However, the conducted experiments have concluded that the detection ratio cannot be decreased below the tolerance threshold values. Therefore, the adversary cannot overcome the effect of the likelihood-based watermarking scheme which, ultimately, prevents him from bypassing the enrollment process.

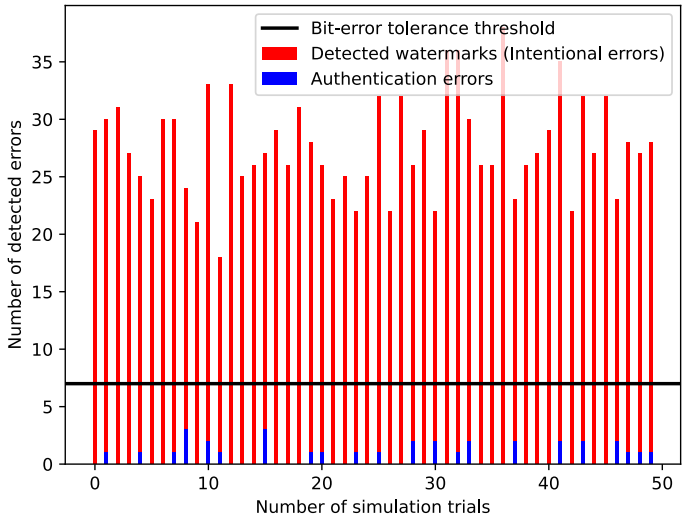


Figure 4.19: Watermark induced errors after the active extraction procedure

4.3.3.3 Water-PUF Security Evaluation

4.3.3.3.1 Attack Probability Formulation

In this part, we discuss the security of the Water-PUF protocol based on the threat model that is presented in Subsection 4.3.2.2. The adversary aims at compromising the protocol through the enrollment of a malicious device that holds a different PUF hardware from the legitimate one. As described in the Hypothesis H_6, the adversary has knowledge of the PUF model that is used by the verifier. In our proposal, the verifier authenticates the adversary if the number of bit errors in the received response R_{adv} in comparison with the predicted response R_v of the verifier does not exceed the tolerance threshold $T = 10\%$. Thus, the protocol validates the enrollment of the attacker if the number of correct bits m_{adv} is expressed as follows:

$$m_{adv} = l - HD(R_{adv}, R_v) \geq l \times (1 - T) \tag{4.2}$$

In this context, the main objective of the adversary is to reduce the Hamming distance $HD(R_{adv}, R_v)$. This distance is expressed by using the accuracy of the adversarial model in the process of correctly predicting the responses for l watermarking challenges, $Acc_{adv_w}(l)$. This is explained by the ability of the attacker to respond correctly to the authentication challenges since he possesses the same PUF model as the verifier. This accuracy metric is formulated as $HD(R_{adv}, R_v) = (1 - Acc_{adv_w}(l)) \times l$. Therefore, the adversary needs to satisfy the Equation 4.2 by enhancing the accuracy of his watermark detection model up to the following bound:

$$Acc_{adv_w}(l) \geq (1 - T)$$

The adversarial attack probability is dependent on the feasibility of achieving the required accuracy to respond correctly to the watermarking challenges. Accordingly, the probability of a successful attack on our protocol, P_{adv}, can be expressed as:

$$P_{adv} = P[Acc_{adv_w}(l) \geq (1 - T)] \tag{4.3}$$

In the following paragraphs, we assess the feasibility of increasing the watermark detection accuracy $Acc_{adv_w}(l)$ by the adversary.

Impersonation Attacks

The Water-PUF protocol exploits a PUF circuit that is embedded on the Prover. The uniqueness property of the PUF responses, described in Subsection 4.3.3.1, prevents the adversary from using another PUF circuit to bypass the authentication. This attack is limited to a random guessing attempt of the PUF responses. Thus, the adversarial attack probability can be expressed as follows:

$$P_{adv} = \frac{1}{2^{m_{adv}}} \leq \frac{1}{2^{(1-T)\times l}} \tag{4.4}$$

Distinguishability Attacks

The adversary can enhance his ability to detect the watermark challenges $Acc_{adv_w}(l)$ through the distribution analysis of the likelihood responses. As highlighted in Fig. 4.14a, there is a noticeable difference between the likelihood output distribution of the watermarked model when applying the authentication and the watermarking challenges. As a result, the adversary is able to successfully identify the watermarks which compromises the security of the protocol. Therefore, Water-PUF applies the Algorithm 2 on the verifier side to prevent this attack by selecting specific authentication challenges for the enrollment procedure. Accordingly, the distribution of the selected watermarking and authentication challenges have been rendered homogeneous, as illustrated in Fig. 4.14b. Thus, the adversary cannot successfully distinguish between them which makes the attack equivalent to a random guess.

Watermark Suppression Attacks

The accuracy of the adversarial model can be increased through the application of a number of watermark suppression techniques. The fine-tuning and the train-from-scratch methods have been extensively simulated in Subsection 4.3.3.2. The fine-tuning technique has yielded a maximum watermark detection accuracy of 73.44%. The train-from-scratch approach focuses on the use of the watermarked model by the adversary to extract approximately correct CRPs to retrain a more complex model. Furthermore, the TFS technique assesses the robustness of the watermarking scheme through the exploitation of another ML methodology that is capable of modeling the PUF behavior. This technique has provided the adversary with a maximum watermark detection accuracy of 71.87% when applying the transferability approach between two deep neural network models.

The active extraction scenario exploits the possibility of having an adversary that is able to query the legitimate PUF hardware prior to the enrollment with the verifier, as described in the Hypothesis H_3. This attacker capability has been studied to evaluate the consequences of providing the adversary with this knowledge and to assess its effect on the correctness of the watermarking procedure. The generated adversarial model has achieved a maximum accuracy of 75% which is not sufficient to bypass the authentication procedure. However, the average watermark detection accuracy throughout the conducted 50 trials is around 61%. Furthermore, we investigate the possibility of using the XORed responses to construct directly the adversarial model. The objective of the model evaluates the possibility of predicting the XORed response based on the knowledge of the two used challenges. The overall accuracy of this model is limited to 75% with considerable loss values. However, we demonstrate that the performance of the model drops to 50% when we apply the watermarking challenges.

The attacks on the watermarking procedure have been experimentally proven infeasible because the adversary cannot enhance the accuracy of the watermark detection. Therefore, the attacker cannot reduce the number of bit-errors below the tolerance threshold T. This proposal prevents the risk of facing an adversary that is running a PUF model instead of using the legitimate PUF hardware to provide the correct responses. Accordingly, the

adversary has to use the PUF hardware to generate the correct responses of the received challenges without the need to identify the watermarks. This action is not possible because of the Hypothesis H_7 that prevents the adversary from querying at will the legitimate PUF hardware once the IoT device is authenticated by the verifier. Thus, the attacker cannot enroll another malicious IoT device by impersonating the legitimate PUF hardware. Accordingly, our proposal satisfies the entity authentication property that is defined in Subsection 4.3.2.2. Moreover, to the best of the authors knowledge, the combination of the Hypothesis H_4 and H_6 is not supported by the existing model-based PUF enrollment protocols [126, 128, 206, 116, 211]. This is due to the insider threat capability and the granted computational power that provides the adversary with the ability to bypass any time-bounded authentication.

4.3.4 Discussion

Our proposal has been simulated over 10^4 trials for each of the previously described adversarial techniques. The simulations have yielded an average successful execution rate of 99.44% with a 0% attack success rate with a tolerance threshold $T = 10\%$. The average bit-error on the verifier side is 2.46 bit per enrollment session which allows us to reduce the tolerance threshold to $T = 1 - Acc_w$. Thus, this operation makes it more difficult for the adversary to bypass the enrollment procedure. Moreover, the Water-PUF protocol eliminates the risk related to the reliability-based attack that was introduced by Becker [25] through the use of an approximate set membership test such as the RobustBF filter [138] to avoid the repetitive execution of the same challenge. Thus, the evaluation of the response reliability by an adversary cannot be conducted. This filter serves as a data structure that is designed to efficiently verify the presence of an element in a particular set. Therefore, it eliminates the possibility of assessing the reliability of a specific response through the repetitive execution of the same challenge.

The Water-PUF protocol could be applied in the case of a recently discovered ML vulnerability in a deployed PUF architecture. Typically, the ML resistant PUF circuits tend to avoid using an obfuscation technique since the mapping function between the challenges and the responses is relatively complex. Thus, this eliminates the risk of revealing it using a machine learning model. In this threat scenario, the service provider can extract a sufficient number of CRPs to construct the ground-truth model $PUF_{model_{gt}}(.)$ that is used for the watermarking procedure. Afterwards, a simple software patch of the IoT object should take place to add the chosen response obfuscation technique. This solution eliminates the need to withdraw the deployed devices and to replace the vulnerable PUF circuits with a suitable and ML resistant architecture. Thus, our protocol prevents a considerable financial loss due to these actions through the appliance of a software patch to the identified hardware vulnerability.

Our proposal relies, mainly, on the watermark extraction algorithm to find the needed watermarking challenges. The search process is based on the random generation of a challenge set and the verification of the likelihood selection criteria. Therefore, the higher the likelihood bound is set, the longer we need to look for the adequate watermarking challenges, as shown in Figure 4.20. The previous experiment has been conducted using a laptop with an Intel(R) Core^{TM} i5 – 9400H CPU @ 2.5GHz × 8 processor, 32 GB of RAM, running Ubuntu 20.04.1 LTS. However, the simulation has been executed using a Python script that is running on a single core. To optimize the computation time, we can proceed by limiting the likelihood distance to a specific interval with respect to the desired computation time. However, the reduction of the likelihood distance affects the correction of the watermark extraction process. Another possible solution is the appliance of a parallel computation technique where we run the procedure simultaneously on multiple cores. Afterward, we eliminate any redundancies in the extracted watermarking challenges. This

solution will reduce significantly the required execution time of the Water-PUF protocol.

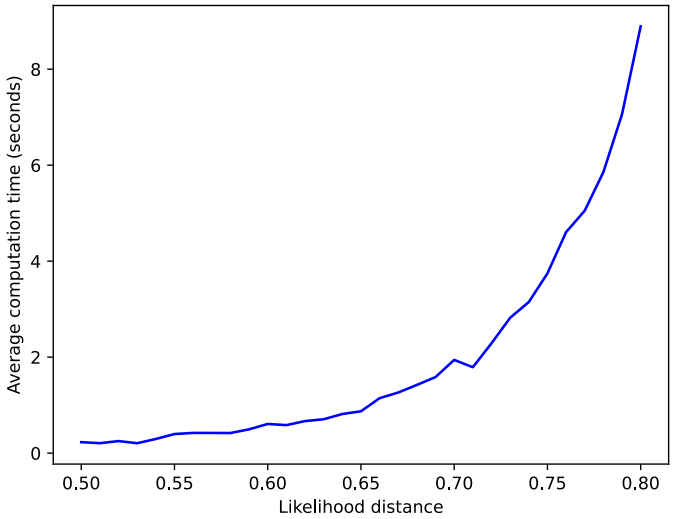


Figure 4.20: Average time execution of the watermark extraction algorithm for different likelihood distances

Furthermore, Our protocol relies on the RobustBF filter to eliminate the risk of replay attacks. The drawback of these membership verification techniques is that they require a storage space that depends on the number of inserted elements. However, the chosen filter reduces significantly this requirement to provide a high verification accuracy along with an optimal storage space on the prover. The RobustBF filter requires a $1,382$ bit of storage per inserted elements. For an enrollment session where we use $2 \times l$ challenges, the size of the filter that is used to store N elements during the life-cycle of the prover is $2 \times l \times N \times 1,382$ bits. For example, if we account for $N = 100\,000$ enrollment sessions and $l = 64$ challenges, the size is approximately $2,21$ MegaBytes. The NVM size of an ESP32 card, that has been previously used to represent a constrained IoT device, is 4 MegaBytes. Thus, we believe that we can apply the RobustBF filter in our enrollment proposal, Water-PUF, in the context of constrained IoT objects.

4.4 Conclusion

In this chapter, we have focused on the usage of a ML model of the PUF circuit to perform the enrollment process due to its scalability advantages. However, the use of a mathematically clonable PUF requires the adoption of additional security measures to prevent the modeling attacks through the collection of CRPs. This operation is considered quite complex to address without having a clear insight of the different entities of the protocol and their respective components. Therefore, we have introduced two enrollment architectures that map the different participating nodes in the authentication process of the IoT device. We have studied a selection of model-based PUF enrollment protocols and we have outlined their security limitations with respect to the identified design flaws.

The proposed architectures have facilitated the mitigation process of some highlighted weakness through the modification of a vulnerable component in the protocol design. The resiliency of the selected enrollment schemes have been assessed against an insider threat within the organization. This study has yielded that these protocols cannot fully guarantee the security of the enrollment procedure when the PUF model is leaked to the adversary.

This attack scenario is generally overlooked by the protocol designers.

As a consequence, we have designed a PUF-based enrollment protocol that exploits a black-box ML watermarking technique which prevents an adversary from bypassing the authentication even if he has the same PUF model as the legitimate verifier. Thus, our design has been demonstrated resilient against this insider threat scenario through the robustness of the proposed watermarking technique. The protocol has been simulated over 10^4 trials that have yielded a 99.44% successful executions with a 0% attack success probability. The insider adversary represents a serious threat regarding the security of the existing PUF-based authentication protocols. Thus, our enrollment scheme, Water-PUF, represents an efficient solution to tackle this issue through a specifically crafted ML watermarking technique.

5 | Conclusion and Future Directions

5.1 Conclusion

The Internet of Things services are being increasingly adopted by the general public and numerous industries. These objects have helped to facilitate the everyday life of the users and to improve the business performances in multiple fields. However, these devices can constitute a serious threat that targets the safety of the end users data. This is due to their resource-constrained nature that prevents the use of the traditional heavy-computing safeguards. The deployment of IoT objects within the network of a user or a business without any security measures can create the perfect point of entry for the adversaries. Therefore, the association of these devices to a network requires the use of lightweight security measures that primarily guarantees the confidentiality and the integrity of the collected data. Furthermore, there is a critical need to authenticate the IoT object in question to avoid the association of a malicious device to the secure network of the user. This way, the IoT oriented businesses can safely exploit the collected data to offer accurate and trusted services to their users.

In this thesis, we have focused on the problem of secure association of resource-constrained IoT devices into the network of the user. We have addressed the issue of secure bootstrapping by dividing the process into two separate phases that have two security objectives. The primary objective is to guarantee the confidentiality and the integrity of the exchanges between the user and the IoT device without the need for pre-shared knowledge by the use of an ad-hoc pairing protocol. This requirement eliminates the necessity for digital certificates for two reasons: the lack for support of asymmetric encryption and the difficulty to properly manage them in massive deployment scenarios. The ad-hoc secure device pairing protocols rely mainly on the use of two techniques to perform the key agreement process: Out-of-Band channels and contextual environment.

In the first phase of our work, we have evaluated the formal and computational security analysis that have been conducted on a selection of state-of-the-art pairing protocols. This study presents a systematic description of the adopted assumptions, the threat model and an assessment of the verification results. We discovered contradictory formal verification outcomes that are related to a lack of formalization of the assessed properties. In addition, we have normalized the used taxonomy in order to enhance the understanding of these security validations. We also discuss the consequences of a recent adversary model that provide the attacker with the ability to partially compromise one of the participating devices.

Subsequently, we have proposed a secure device pairing protocol, COOB, that efficiently combines these two techniques to enhance the security of the key agreement in a hostile environment that is controlled by the adversary. This advanced threat model compromises the security of the state-of-the-art context-based schemes. In addition, we reduce the communication time that is required by the Out-of-Band channel through the

use of a fast contextual commitment approach. Our protocol has been formally validated by the TAMARIN verification tool. COOB guarantees the confidentiality of the shared Diffie-Hellman key at the end of the protocol execution and it offers the injective agreement on the public DH keys that are exchanged on the insecure In-Band channel. The latter property guarantees that the pairing participants agree on the exchanged public keys which mitigates the risks related to a Man-in-the-Middle attack.

In the second phase of our work, we have focused on addressing the entity authentication issue of the associated IoT objects through the use of a secure device enrollment protocol. This process verifies the identity and the origin of the object in question. We have studied the use of physical unclonable functions in the state-of-the-art enrollment protocols. Numerous of these schemes rely on machine learning techniques to model the behavior of the PUF in order to reduce the storage requirement of the challenge-response pairs. We have discussed the importance of guaranteeing the secrecy of the PUF model and the risks related to the leakage of this sensitive information to an adversary due to an insider threat within the organization.

To mitigate this raised issue, we have constructed a watermarking technique of the PUF model that identifies the use of the leaked model by the adversary. This procedure prevents an adversary from relying on the watermarked model in question or another derivative model to bypass the authentication. Therefore, any leakage of the watermarked PUF model that is used for the enrollment does not affect the correctness of the protocol. Our enrollment scheme, Water-PUF, has been validated by a number of simulations against numerous watermark suppression attacks to assess the robustness of our proposal.

5.2 Open Issues and Future Directions

Hereafter we shed some light on a number of future directions and open issues relating to securing the bootstrapping process of Internet of Things:

- In Chapter 3, we have applied a state-of-the-art fast contextual protocol in our proposal COOB. This technique uses channel state information from a number of nearby Wi-Fi access points. In order to enhance the ease-of-adoption of our protocol, we need to rely on other contextual features that are available outside of urban areas. A possible direction is to study the use of magnetometer measurements to perform this operation.

- Throughout the thesis, we assume that the firmware of the legitimate IoT devices has not been compromised. Our two-phased bootstrapping procedure requires a lightweight verification technique of the executed software by the objects. The use of the PUF circuit could be a promising perspective to provide a root key that is used for the software validation. However, in the case of resource-constrained devices, the PUF key generation procedure requires the use of lightweight error correcting codes that can be suitable for this context. Thus, the optimization of these error correcting techniques can be a promising direction to help the adoption of the PUF-based secure bootstrapping solutions.

- In Chapter 4, the PUF model-based enrollment protocols require the use of an obfuscation technique to provide an additional complexity to the mapping function between the challenges and the responses. The correctness of the response obfuscation technique is crucial to guarantee the security of the enrollment process. This is due to the necessity of limiting the correct response generation process to two entities: the Prover and the authentication server. Thus, the adversary would not be able to correctly respond to the watermarking challenges. In our proposal, we

have relied upon a simple XOR-based obfuscation technique to validate our concept. However, this approach can be enhanced by the application of the Shamir Secret Sharing [179] as demonstrated in the work of Chen et al. [45]. This obfuscation technique has reduced the effectiveness of the LR, the ANN and the CMA-ES modeling from an approximately correct prediction rate to a random guess. The reported approach has been designed to be error-tolerant with respect to the noisy responses of the PUF circuit. Furthermore, it is considered as a resource efficient technique that can be applied on IoT constrained devices.

- In Chapter 4, we assume, in our adopted threat model, that the adversary cannot retrieve the keys that are stored in the IoT devices. This requirement can be satisfied through the use of a Trusted Platform Module (TPM) [69]. Unfortunately, this security feature is not always supported by these objects. The ESP32 is a popular example of a widely used chip in IoT devices that does not support a TPM. Thus, resulting in the vulnerability CVE-2019-17391 [52] that aims at extracting the cryptographic keys of the secure boot and the flash encryption features. Therefore, the adversary can indeed reveal any secret keys that have been previously exchanged during the device pairing process. Hence, in the future, we aim at extending the threat model to grant the adversary the power to reveal any information that is not securely stored. Consequently, this adversarial capability would compromise any traditional cryptographic mechanism that aims at authenticating the devices based on the confidentiality and the integrity of the stored keys.

  In order to tackle this issue, we have envisioned to use the PUF hardware on the Prover and the watermarked PUF model on the Verifier to produce the needed session keys. Evidently, the use of the watermarking scheme in Water-PUF would eliminate the risk related to the key re-computation by an adversary with the leaked ML model. Furthermore, the correctness of the PUF-based key generation procedure requires the use of error correcting codes which results in a higher computational cost. Thus, we will focus on the devices with enough resources to handle these operations. This procedure can also be applicable in the case of IoT objects with a TPM that cannot perform a secure pairing due to the lack of I/O interfaces to construct an Out-of-Band channel. These devices are generally covered by the *Just Works* variant of the Bluetooth standard Secure Simple Pairing protocol [32]. This scheme perform a key exchange on the paired devices without any protection against the man-in-the-middle attack.

  Moreover, we would like to study in depth the effectiveness of applying a distance-bounding protocol, such as the time-bounded authentication in Subsection 4.2.1, to counter the relay attacks on the enrollment process. Thus, we are required to adopt the 4CE architecture by sharing the PUF model with Verifier nodes to routinely verify the existence of the legitimate IoT object within the demanded distance-bound. Thus, the protection against an insider information leakage threat is crucial to guarantee the correctness of the device authentication. This decentralized procedure would reduce the computational and the communication cost on the authentication server which significantly increases the scalability of the solution. The combination of these future enhancements with our proposed watermark-based protocol would provide the necessary security to protect these systems with optimal costs.

Finally, we hope that the concepts and the ideas that were presented in this thesis will help pave the way to a reliable and safe deployment of Internet of Things devices.

List of Acronyms

Author Publication List

International Journals

1. Khalfaoui, S., Leneutre, J., Villard, A., Ma, J., & Urien, P. (2021). *Security Analysis of Out-of-Band Device Pairing Protocols: A Survey.* Wireless Communications and Mobile Computing, 2021.

2. Khalfaoui, S., Leneutre, J., Villard, A., Gazeau, I., Ma, J., & Urien, P. (2021). *Security Analysis of Machine Learning-Based PUF Enrollment Protocols: A Review.* Sensors, 21(24), 8415.

International Conferences

1. Khalfaoui, S., Leneutre, J., Villard, A., Ma, J., & Urien, P. (2020, October). *COOB: Hybrid Secure Device Pairing Scheme in a Hostile Environment.* In International Conference on Security and Privacy in Communication Systems (pp. 419-438). Springer, Cham.

2. Khalfaoui, S., Leneutre, J., Villard, A., Gazeau, I., Ma, J., Danger, J. L., & Urien, P. (2021, November). *Water-PUF: An Insider Threat Resistant PUF Enrollment Protocol Based on Machine Learning Watermarking.* In 2021 IEEE 20th International Symposium on Network Computing and Applications (NCA) (pp. 1-10). IEEE.

Patents

1. Khalfaoui, S., Villard, A., Ma, J., & Leneutre, J. (2020, May). *Procédé d'appairage.* EP 3913951A1.

2. Khalfaoui, S., Villard, A., Ma, J., & Leneutre, J. (2021, June). *Procédé et système d'authentification d'un dispositif équipé d'un circuit PUF.* FR 2107112.

Talks and Presentations

1. Sameh Khalfaoui, *Bootstrapping : mécanismes d'amorçage de la confiance pour la sécurité de l'IoT*, Workshop SEIDO, February 2020.

2. Sameh Khalfaoui, *COOB: Hybrid Secure Device Pairing Scheme in a Hostile Environment*, 16th EAI International Conference on Security and Privacy in Communication Networks (SecureComm 2020), October 2020.

3. Sameh Khalfaoui, *COOB: Hybrid Secure Device Pairing Scheme in a Hostile Environment*, ACES Team Seminar (Télécom Paris), November 2020.

4. Sameh Khalfaoui, *Water-PUF: An Insider Threat Resistant PUF Enrollment Protocol Based on Machine Learning Watermarking*, 20th IEEE International Symposium on Network Computing and Applications (NCA 2021), November 2021.

Bibliography

[1] COOB: Tamarin model, https://github.com/samehkhalfaoui/COOB-TAMARIN-model.git

[2] Sibeam captures world's first 60ghz millimeter-wave smartphone design win in letv's flagship smartphone, le max (2015), https://www.businesswire.com/news/home/20150519005350/en/SiBEAM-Captures-World%E2%80%99s-60GHz-Millimeter-Wave-Smartphone-Design

[3] Hp elite x2 1011 g2 - connecting to the wireless dock (2019), https://support.hp.com/id-en/document/c04587366#AbT1

[4] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., Levkowetz, H., et al.: Extensible authentication protocol (eap) (2004)

[5] Adi, Y., Baum, C., Cisse, M., Pinkas, B., Keshet, J.: Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In: 27th {USENIX} Security Symposium ({USENIX} Security 18). pp. 1615–1631 (2018)

[6] Ahmed, Z., Danish, S.M., Qureshi, H.K., Lestas, M.: Protecting iots from mirai botnet attacks using blockchains. In: 2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD). pp. 1–6. IEEE (2019)

[7] Akter, S., Chakraborty, T., Khan, T.A., Chellappan, S., Al Islam, A.A.: Can you get into the middle of near field communication? In: 2017 IEEE 42nd Conference on Local Computer Networks (LCN). pp. 365–373. IEEE (2017)

[8] Ali, A., Khan, F.A.: Key agreement schemes in wireless body area networks: Taxonomy and state-of-the-art. Journal of medical systems **39**(10), 1–14 (2015)

[9] Aljafar, M.J., Acken, J.M.: Survey on the benefits of using memristors for pufs. International Journal of Parallel, Emergent and Distributed Systems **0**(0), 1–28 (2021). https://doi.org/10.1080/17445760.2021.1972295, https://doi.org/10.1080/17445760.2021.1972295

[10] Alliance, W.F.: Wigig® and the future of seamless connectivity. Wi-Fi Alliance (2013)

[11] Alliance, W.F.: Wi-fi simple configuration technical specification, version 2.0. 5 (2014)

[12] Alliance, W.F.: Device provisioning protocol specification v1. 1 (2018)

[13] Alpern, B., Schneider, F.B.: Recognizing safety and liveness. Distributed computing **2**(3), 117–126 (1987)

[14] Ammar, M., Washha, M., Crispo, B.: Wise: Lightweight intelligent swarm attestation scheme for iot (the verifier's perspective). In: 2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). pp. 1–8. IEEE (2018)

[15] Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., Durumeric, Z., Halderman, J.A., Invernizzi, L., Kallitsis, M., et al.: Understanding the mirai botnet. In: 26th {USENIX} security symposium ({USENIX} Security 17). pp. 1093–1110 (2017)

[16] Asim, M., Guajardo, J., Kumar, S.S., Tuyls, P.: Physical unclonable functions and their applications to vehicle system security. In: VTC Spring 2009-IEEE 69th Vehicular Technology Conference. pp. 1–5. IEEE (2009)

[17] Association, I.S., et al.: Ieee std 802.11 ad-2012,"part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications," amendment 3: Enhancements for very high throughput in the 60 ghz band, ieee standard for information technology—telecommunications and information exchange between systems—local and metropolitan area networks—specific requirements. IEEE Computer Society (2012)

[18] Atif, M., Muralidharan, S., Ko, H., Yoo, B.: Wi-ESP—A tool for CSI-based Device-Free Wi-Fi Sensing (DFWS). Journal of Computational Design and Engineering **7**(5), 644–656 (05 2020). https://doi.org/10.1093/jcde/qwaa048, `https://doi.org/10.1093/jcde/qwaa048`

[19] Aura, T., Sethi, M.: Nimble out-of-band authentication for eap (eap-noob). draft-aura-eap-noob-03 (work in progress) (2018)

[20] Aura, T., Sethi, M., Peltonen, A.: Nimble Out-of-Band Authentication for EAP (EAP-NOOB). RFC 9140 (Dec 2021). https://doi.org/10.17487/RFC9140, `https://www.rfc-editor.org/info/rfc9140`

[21] Balfanz, D., Smetters, D.K., Stewart, P., Wong, H.C.: Talking to strangers: Authentication in ad-hoc wireless networks. In: NDSS. Citeseer (2002)

[22] Bansal, N.: Iot applications in retail. In: Designing Internet of Things Solutions with Microsoft Azure, pp. 217–238. Springer (2020)

[23] Barbareschi, M., Bagnasco, P., Mazzeo, A.: Authenticating iot devices with physically unclonable functions models. In: 2015 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC). pp. 563–567. IEEE (2015)

[24] Barker, E., Smid, M., Branstad, D., et al.: A profile for us federal cryptographic key management systems. NIST Special Publication **800**, 152 (2015)

[25] Becker, G.T.: The gap between promise and reality: On the insecurity of xor arbiter pufs. In: International Workshop on Cryptographic Hardware and Embedded Systems. pp. 535–555. Springer (2015)

[26] Bellare, M., Rogaway, P.: Entity authentication and key distribution. In: Annual international cryptology conference. pp. 232–249. Springer (1993)

[27] Bellare, M., Rogaway, P.: Provably secure session key distribution: the three party case. In: Proceedings of the twenty-seventh annual ACM symposium on Theory of computing. pp. 57–66 (1995)

[28] Benesty, J., Chen, J., Huang, Y., Cohen, I.: Pearson correlation coefficient. In: Noise reduction in speech processing, pp. 1–4. Springer (2009)

[29] Blake-Wilson, S., Menezes, A.: Unknown key-share attacks on the station-to-station (sts) protocol. In: International Workshop on Public Key Cryptography. pp. 154–170. Springer (1999)

[30] Blanchet, B.: Cryptoverif: A computationally-sound security protocol verifier. Tech. Rep. (2017)

[31] Blanchet, B., Smyth, B., Cheval, V., Sylvestre, M.: Proverif 2.00: Automatic cryptographic protocol verifier, user manual and tutorial. Version from pp. 05–16 (2018)

[32] Bluetooth, S.: Bluetooth core specification v5. 0. Bluetooth Special Interest Group: Kirkland, WA, USA (2016)

[33] Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proceedings of the fifth annual workshop on Computational learning theory. pp. 144–152 (1992)

[34] Brelurut, A., Gerault, D., Lafourcade, P.: Survey of distance bounding protocols and threats. In: International symposium on foundations and practice of security. pp. 29–49. Springer (2015)

[35] Brown, D.R.: Recommended elliptic curve domain parameters. Standards Efficient Cryptogr. Group Ver **1** (2010)

[36] Bundesnetzagentur: Bundesnetzagentur removes children's doll "cayla" from the market (2017), `https://www.bundesnetzagentur.de/SharedDocs/Pressemitteilungen/EN/2017/17022017_cayla.html`

[37] Burrows, M., Abadi, M., Needham, R.M.: A logic of authentication. Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences **426**(1871), 233–271 (1989)

[38] Cagalj, M., Capkun, S., Hubaux, J.P.: Key agreement in peer-to-peer wireless networks. Proceedings of the IEEE **94**(2), 467–478 (2006)

[39] Cam-Winget, N., Sadeghi, A.R., Jin, Y.: Can iot be secured: Emerging challenges in connecting the unconnected. In: 2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC). pp. 1–6. IEEE (2016)

[40] Castelluccia, C., Avoine, G.: Noisy tags: A pretty good key exchange protocol for rfid tags. In: International Conference on Smart Card Research and Advanced Applications. pp. 289–299. Springer (2006)

[41] Chadha, R., Cheval, V., Ciobâcă, Ş., Kremer, S.: Automated verification of equivalence properties of cryptographic protocols. ACM Transactions on Computational Logic (TOCL) **17**(4), 1–32 (2016)

[42] Chandy, A.: A review on iot based medical imaging technology for healthcare applications. Journal of Innovative Image Processing (JIIP) **1**(01), 51–60 (2019)

[43] Chang, R., Shmatikov, V.: Formal analysis of authentication in bluetooth device pairing. FCS-ARSPA07 p. 45 (2007)

[44] Chatterjee, B., Das, D., Maity, S., Sen, S.: Rf-puf: Enhancing iot security through authentication of wireless nodes using in-situ machine learning. IEEE Internet of Things Journal **6**(1), 388–398 (2018)

[45] Chen, S., Li, B., Chen, Z., Zhang, Y., Wang, C., Tao, C.: Novel strong-puf-based authentication protocols leveraging shamir's secret sharing. IEEE Internet of Things Journal pp. 1–1 (2021). https://doi.org/10.1109/JIOT.2021.3065836

[46] Choi, Y.J., Kang, H.J., Lee, I.G.: Scalable and secure internet of things connectivity. Electronics **8**(7), 752 (2019)

[47] Classen, J., Chen, J., Steinmetzer, D., Hollick, M., Knightly, E.: The spy next door: Eavesdropping on high throughput visible light communications. In: Proceedings of the 2nd International Workshop on Visible Light Communications Systems. pp. 9–14. ACM (2015)

[48] Claycomb, W.R., Shin, D.: Extending formal analysis of mobile device authentication. J. Internet Serv. Inf. Secur. **1**(1), 86–102 (2011)

[49] Cooper, G., Behm, B., Chakraborty, A., Kommalapati, H., Mandyam, G., ARM, H.T., Bartsch, W.: Fido device onboard specification 1.1 (2021)

[50] Corporation, L.M.: Iot is transforming modern warfare (2017), `https://www.lockheedmartin.com/en-us/news/features/2017/internet-of-thingstransofrming-modern-warfare.html`

[51] Council, N.R.: Counterfeit Deterrent Features for the Next-Generation Currency Design. The National Academies Press, Washington, DC (1993). https://doi.org/10.17226/2267, `https://www.nap.edu/catalog/2267/counterfeit-deterrent-features-for-the-next-generation-currency-design`

[52] Security Advisory concerning fault injection and eFuse protections. National Vulnerability Database (November 2019), `https://nvd.nist.gov/vuln/detail/CVE-2019-17391`, [online] `https://nvd.nist.gov/vuln/detail/CVE-2019-17391`

[53] Darvish Rouhani, B., Chen, H., Koushanfar, F.: Deepsigns: An end-to-end watermarking framework for ownership protection of deep neural networks. In: Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems. pp. 485–497 (2019)

[54] Delaune, S., Kremer, S., Robin, L.: Formal verification of protocols based on short authenticated strings. In: 2017 IEEE 30th Computer Security Foundations Symposium (CSF). pp. 130–143. IEEE (2017)

[55] Delvaux, J.: Machine-learning attacks on polypufs, ob-pufs, rpufs, lhs-pufs, and puf–fsms. IEEE Transactions on Information Forensics and Security **14**(8), 2043–2058 (2019)

[56] Diffie, W., Hellman, M.: New directions in cryptography. IEEE Trans. Inf. Theor. **22**(6), 644–654 (Sep 2006). https://doi.org/10.1109/TIT.1976.1055638, `http://dx.doi.org/10.1109/TIT.1976.1055638`

[57] Diffie, W., Van Oorschot, P.C., Wiener, M.J.: Authentication and authenticated key exchanges. Designs, Codes and cryptography **2**(2), 107–125 (1992)

[58] Do, Q., Martini, B., Choo, K.K.R.: The role of the adversary model in applied security research. Computers & Security **81**, 156–181 (2019)

[59] Dolev, D., Yao, A.: On the security of public key protocols. IEEE Transactions on information theory **29**(2), 198–208 (1983)

[60] Dziech, A., Bialas, J., Glowacz, A., Korus, P., Leszczuk, M., Matiolalski, A., Baran, R.: Overview of recent advances in cctv processing chain in the indect and insigma projects. In: 2013 International Conference on Availability, Reliability and Security. pp. 836–843. IEEE (2013)

[61] El-hajj, M., Fadlallah, A., Chamoun, M., Serhrouchni, A.: A survey of internet of things (iot) authentication schemes. Sensors **19**(5), 1141 (2019)

[62] Cyber Security for Consumer Internet of Things: Baseline Requirements. Standard ETSI EN 303 645 v2.1.1, European Telecommunications Standards Institute, Sophia Antipolis, FRANCE (2020)

[63] Fábrega, F.J.T., Herzog, J.C., Guttman, J.D.: Strand spaces: Why is a security protocol correct? In: Proceedings. 1998 IEEE Symposium on Security and Privacy (Cat. No. 98CB36186). pp. 160–171. IEEE (1998)

[64] Fagan, M., Fagan, M., Megas, K.N., Scarfone, K., Smith, M.: IoT Device Cybersecurity Capability Core Baseline. US Department of Commerce, National Institute of Standards and Technology (2020)

[65] Fang, Y., Wang, C., Ma, Q., Gu, C., O'Neill, M., Liu, W.: Attacking arbiter pufs using various modeling attack algorithms: A comparative study. In: 2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS). pp. 394–397. IEEE (2018)

[66] Fomichev, M., Álvarez, F., Steinmetzer, D., Gardner-Stephen, P., Hollick, M.: Survey and systematization of secure device pairing. IEEE Communications Surveys & Tutorials **20**(1), 517–550 (2017)

[67] Fomichev, M., Maass, M., Almon, L., Molina, A., Hollick, M.: Perils of zero-interaction security in the internet of things. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies **3**(1), 10 (2019)

[68] Francis, L., Hancke, G., Mayes, K., Markantonakis, K.: Practical nfc peer-to-peer relay attack using mobile phones. In: International Workshop on Radio Frequency Identification: Security and Privacy Issues. pp. 35–49. Springer (2010)

[69] Furtak, J., Zieliński, Z., Chudzikiewicz, J.: Security techniques for the wsn link layer within military iot. In: 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT). pp. 233–238. IEEE (2016)

[70] Gao, Y., Li, G., Ma, H., Al-Sarawi, S.F., Kavehei, O., Abbott, D., Ranasinghe, D.C.: Obfuscated challenge-response: A secure lightweight authentication mechanism for puf-based pervasive devices. In: 2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops). pp. 1–6. IEEE (2016)

[71] Gehrmann, C., Mitchell, C.J., Nyberg, K.: Manual authentication for wireless devices. RSA Cryptobytes **7**(1), 29–37 (2004)

[72] GlobeNewswire, I.: Digital twin industry was valued at $3.6 billion in 2019 and is forecast to reach 73.2 billion by 2030 (2020), https://www.globenewswire.com/news-release/2020/07/21/2065355/0/en/Digital-Twin-Industry-Was-Valued-at-3-6-Billion-in-2019-and-is-Forecast-to-Reach-73.html

[73] Goodrich, M.T., Sirivianos, M., Solis, J., Tsudik, G., Uzun, E.: Loud and clear: Human-verifiable authentication based on audio. In: 26th IEEE International Conference on Distributed Computing Systems (ICDCS'06). pp. 10–10. IEEE (2006)

[74] Gu, C., Chang, C.H., Liu, W., Yu, S., Ma, Q., O'neill, M.: A modeling attack resistant deception technique for securing puf based authentication. In: 2019 Asian Hardware Oriented Security and Trust Symposium (AsianHOST). pp. 1–6. IEEE (2019)

[75] Gu, Z., Liu, Y.: Scalable group audio-based authentication scheme for iot devices. In: 2016 12th International Conference on Computational Intelligence and Security (CIS). pp. 277–281. IEEE (2016)

[76] Guo, J., Potkonjak, M.: Watermarking deep neural networks for embedded systems. In: 2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD). pp. 1–8. IEEE (2018)

[77] Guo, J., Potkonjak, M.: Evolutionary trigger set generation for dnn black-box watermarking. arXiv preprint arXiv:1906.04411 (2019)

[78] Halevi, T., Saxena, N.: Acoustic eavesdropping attacks on constrained wireless device pairing. IEEE Transactions on Information Forensics and Security **8**(3), 563–577 (2013)

[79] Han, J., Chung, A.J., Sinha, M.K., Harishankar, M., Pan, S., Noh, H.Y., Zhang, P., Tague, P.: Do you feel what i hear? enabling autonomous iot device pairing using different sensor types. In: 2018 IEEE Symposium on Security and Privacy (SP). pp. 836–852. IEEE (2018)

[80] Han, J., Lin, Y.H., Perrig, A., Bai, F.: Short paper: Mvsec: secure and easy-to-use pairing of mobile devices with vehicles. In: Proceedings of the 2014 ACM conference on Security and privacy in wireless & mobile networks. pp. 51–56. ACM (2014)

[81] Hancke, G.P., Kuhn, M.G.: Attacks on time-of-flight distance bounding channels. In: Proceedings of the First ACM Conference on Wireless Network Security. p. 194–202. WiSec '08, Association for Computing Machinery, New York, NY, USA (2008). https://doi.org/10.1145/1352533.1352566, `https://doi.org/10.1145/1352533.1352566`

[82] Handschuh, H., Heys, H.M.: A timing attack on rc5. In: International Workshop on Selected Areas in Cryptography. pp. 306–318. Springer (1998)

[83] Hansen, N.: The CMA Evolution Strategy: A Comparing Review, pp. 75–102. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)

[84] Hoepman, J.H.: The ephemeral pairing problem. In: International Conference on Financial Cryptography. pp. 212–226. Springer (2004)

[85] Hossein Motlagh, N., Mohammadrezaei, M., Hunt, J., Zakeri, B.: Internet of things (iot) and the energy sector. Energies **13**(2), 494 (2020)

[86] Huang, K.C., Wang, Z.: Millimeter wave communication systems, vol. 29. John Wiley & Sons (2011)

[87] Hunkeler, U., Truong, H.L., Stanford-Clark, A.: Mqtt-s—a publish/subscribe protocol for wireless sensor networks. In: 2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08). pp. 791–798. IEEE (2008)

[88] Idriss, T., Idriss, H., Bayoumi, M.: A puf-based paradigm for iot security. In: 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT). pp. 700–705 (2016). https://doi.org/10.1109/WF-IoT.2016.7845456

[89] Instrument, N.: Advanced rfid measurements: Basic theory to protocol conformance test (2013)

[90] Jana, S., Premnath, S.N., Clark, M., Kasera, S.K., Patwari, N., Krishnamurthy, S.V.: On the effectiveness of secret key extraction from wireless signal strength in real environments. In: Proceedings of the 15th annual international conference on Mobile computing and networking. pp. 321–332 (2009)

[91] Jennings, C.: Transitive trust enrollment for constrained devices. Web http://www. lix. polytechnique. fr/hipercom/SmartObjectSecurity/papers/CullenJennings. pdf (2012)

[92] Jin, R., Shi, L., Zeng, K., Pande, A., Mohapatra, P.: Magpairing: Pairing smartphones in close proximity using magnetometers. IEEE Transactions on information forensics and security **11**(6), 1306–1320 (2015)

[93] Juels, A., Sudan, M.: A fuzzy vault scheme. Designs, Codes and Cryptography **38**(2), 237–257 (2006)

[94] Jurdak, R., Ruzzelli, A.G., O'hare, G.M., Lopes, C.V.: Mote-based underwater sensor networks: opportunities, challenges, and guidelines. Telecommunication Systems **37**(1), 37–47 (2008)

[95] Kainda, R., Flechais, I., Roscoe, A.: Usability and security of out-of-band channels in secure device pairing protocols. In: Proceedings of the 5th Symposium on Usable Privacy and Security. p. 11. ACM (2009)

[96] Kannimuthu, P., Thangamuthu, J.: Decision tree trust (dttrust)-based authentication mechanism to secure rpl routing protocol on internet of battlefield thing (iobt). International Journal of Business Data Communications and Networking (IJBDCN) **17**(1), 1–23 (2021)

[97] Karapanos, N., Marforio, C., Soriente, C., Capkun, S.: Sound-proof: usable two-factor authentication based on ambient sound. In: 24th {USENIX} Security Symposium ({USENIX} Security 15). pp. 483–498 (2015)

[98] Katzenbeisser, S., Kocabaş, Ü., Van Der Leest, V., Sadeghi, A.R., Schrijen, G.J., Wachsmann, C.: Recyclable pufs: Logically reconfigurable pufs. Journal of Cryptographic Engineering **1**(3), 177 (2011)

[99] Kfir, Z., Wool, A.: Picking virtual pockets using relay attacks on contactless smartcard. In: First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05). pp. 47–58. IEEE (2005)

[100] KhalafAlla, M.: Comprehensive study of physical unclonable functions on fpgas: correlation driven implementation, deep learning modeling attacks, and countermeasures (2020)

[101] Khalafalla, M., Gebotys, C.: Pufs deep attacks: Enhanced modeling attacks using deep learning techniques to break the security of double arbiter pufs. In: 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE). pp. 204–209. IEEE (2019)

[102] Khalfaoui, S., Leneutre, J., Villard, A., Gazeau, I., Ma, J., Danger, J.L., Urien, P.: Water-puf: An insider threat resistant puf enrollment protocol based on machine learning watermarking. In: 2021 IEEE 20th International Symposium on Network Computing and Applications (NCA). pp. 1–10. IEEE (2021)

[103] Khalfaoui, S., Leneutre, J., Villard, A., Ma, J., Urien, P.: Coob: Hybrid secure device pairing scheme in a hostile environment. In: International Conference on Security and Privacy in Communication Systems. pp. 419–438. Springer (2020)

[104] Khalfaoui, S., Leneutre, J., Villard, A., Ma, J., Urien, P.: Coob: Hybrid secure device pairing scheme in a hostile environment. In: International Conference on Security and Privacy in Communication Systems. pp. 419–438. Springer (2020)

[105] Kovačević, T., Perković, T., Čagalj, M.: Flashing displays: user-friendly solution for bootstrapping secure associations between multiple constrained wireless devices. Security and Communication Networks **9**(10), 1050–1071 (2016)

[106] Kroeger, T., Cheng, W., Guilley, S., Danger, J.L., Karimi, N.: Effect of aging on puf modeling attacks based on power side-channel observations. In: 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE). pp. 454–459. IEEE (2020)

[107] Kumar, A., Saxena, N., Tsudik, G., Uzun, E.: A comparative study of secure device pairing methods. Pervasive and Mobile Computing **5**(6), 734–749 (2009)

[108] Kumar, S., Niamat, M.: Machine learning based modeling attacks on a configurable puf. In: NAECON 2018-IEEE National Aerospace and Electronics Conference. pp. 169–173. IEEE (2018)

[109] Kuo, C., Walker, J., Perrig, A.: Low-cost manufacturing, usability, and security: An analysis of bluetooth simple pairing and wi-fi protected setup. In: International Conference on Financial Cryptography and Data Security. pp. 325–340. Springer (2007)

[110] Kusters, L., Willems, F.M.J.: Secret-key capacity regions for multiple enrollments with an sram-puf. IEEE Transactions on Information Forensics and Security **14**(9), 2276–2287 (2019). https://doi.org/10.1109/TIFS.2019.2895552

[111] Laur, S., Nyberg, K.: Efficient mutual data authentication using manually authenticated strings. In: International Conference on Cryptology and Network Security. pp. 90–107. Springer (2006)

[112] Lee, J.W., Lim, D., Gassend, B., Suh, G.E., Van Dijk, M., Devadas, S.: A technique to build a secret key in integrated circuits for identification and authentication applications. In: 2004 Symposium on VLSI Circuits. Digest of Technical Papers (IEEE Cat. No. 04CH37525). pp. 176–179. IEEE (2004)

[113] Lee, K., Raghunathan, V., Raghunathan, A., Kim, Y.: Syncvibe: Fast and secure device pairing through physical vibration on commodity smartphones. In: 2018 IEEE 36th International Conference on Computer Design (ICCD). pp. 234–241. IEEE (2018)

[114] Lenstra, A.K.: Key lengths. Tech. rep. (2006)

[115] Li, M., Yu, S., Lou, W., Ren, K.: Group device pairing based secure sensor association and key management for body area networks. In: 2010 Proceedings IEEE INFOCOM. pp. 1–9. IEEE (2010)

[116] Liang, W., Xie, S., Long, J., Li, K.C., Zhang, D., Li, K.: A double puf-based rfid identity authentication protocol in service-centric internet of things environments. Information Sciences **503**, 129–147 (2019)

[117] Lindgren, A., Ahlgren, B.: OMA Lightweight M2M for Management of Information Centric Networks. Internet-Draft draft-lindgren-icnrg-lwm2m4icn-00 (Nov 2017), https://datatracker.ietf.org/doc/html/draft-lindgren-icnrg-lwm2m4icn-00, work in Progress

[118] Liu, W., Zhang, L., Zhang, Z., Gu, C., Wang, C., O'neill, M., Lombardi, F.: Xor-based low-cost reconfigurable pufs for iot security. ACM Transactions on Embedded Computing Systems (TECS) **18**(3), 1–21 (2019)

[119] Liu, Y., Draper, S.C., Sayeed, A.M.: Exploiting channel diversity in secret key generation from multipath fading randomness. IEEE Transactions on information forensics and security **7**(5), 1484–1497 (2012)

[120] Lowe, G.: A hierarchy of authentication specifications. In: Proceedings 10th Computer Security Foundations Workshop. pp. 31–43. IEEE (1997)

[121] Ltd, M.M.R.P.: Industrial iot (iiot) market by component, application (robotics, maintenance, monitoring, resource optimization, supply chain, management), industry (aerospace, automotive, energy, healthcare, manufacturing, retail), and geography - global forecast to 2027 (2021), https://www.globenewswire.com/news-release/2021/09/30/2306043/0/en/Industrial-IoT-IIoT-Market-Worth-263-4-Billion-by-2027-Market-Size-Share-Forecasts-html

[122] Machida, T., Yamamoto, D., Iwamoto, M., Sakiyama, K.: A new arbiter puf for enhancing unpredictability on fpga. The Scientific World Journal **2015**, 864812 (Sep 2015). https://doi.org/10.1155/2015/864812, https://doi.org/10.1155/2015/864812

[123] Maes, R.: Physically unclonable functions: Properties. In: Physically Unclonable Functions, pp. 49–80. Springer (2013)

[124] Maiti, A., Gunreddy, V., Schaumont, P.: A systematic method to evaluate and compare the performance of physical unclonable functions. In: Embedded systems design with FPGAs, pp. 245–267. Springer (2013)

[125] Majzoobi, M., Elnably, A., Koushanfar, F.: Fpga time-bounded unclonable authentication. In: International Workshop on Information Hiding. pp. 1–16. Springer (2010)

[126] Majzoobi, M., Koushanfar, F.: Time-bounded authentication of fpgas. IEEE Transactions on Information Forensics and Security **6**(3), 1123–1135 (2011)

[127] Majzoobi, M., Koushanfar, F., Potkonjak, M.: Techniques for design and implementation of secure reconfigurable pufs. ACM Transactions on Reconfigurable Technology and Systems (TRETS) **2**(1), 1–33 (2009)

[128] Majzoobi, M., Rostami, M., Koushanfar, F., Wallach, D.S., Devadas, S.: Slender puf protocol: A lightweight, robust, and secure authentication by substring matching. In: 2012 IEEE Symposium on Security and Privacy Workshops. pp. 33–44. IEEE (2012)

[129] Maroli, A., Narwane, V.S., Gardas, B.B.: Applications of iot for achieving sustainability in agricultural sector: A comprehensive review. Journal of Environmental Management **298**, 113488 (2021)

[130] Mathur, S., Miller, R., Varshavsky, A., Trappe, W., Mandayam, N.: Proximate: proximity-based secure pairing using ambient wireless signals. In: Proceedings of the 9th international conference on Mobile systems, applications, and services. pp. 211–224 (2011)

[131] Mathur, S., Trappe, W., Mandayam, N., Ye, C., Reznik, A.: Radio-telepathy: extracting a secret key from an unauthenticated wireless channel. In: Proceedings of the 14th ACM international conference on Mobile computing and networking. pp. 128–139 (2008)

[132] McGrath, T., Bagci, I.E., Wang, Z.M., Roedig, U., Young, R.: A puf taxonomy. Applied physics reviews **6**, 011303 (2019)

[133] Meier, S., Schmidt, B., Cremers, C., Basin, D.: The tamarin prover for the symbolic analysis of security protocols. In: International Conference on Computer Aided Verification. pp. 696–701. Springer (2013)

[134] Mirzadeh, S., Cruickshank, H., Tafazolli, R.: Secure device pairing: A survey. IEEE Communications Surveys & Tutorials **16**(1), 17–40 (2013)

[135] Miyaji, A., Nonaka, M., Takii, Y.: Known plaintext correlation attack against rc5. In: Cryptographers' Track at the RSA Conference. pp. 131–148. Springer (2002)

[136] Mukhopadhyay, S.C., Suryadevara, N.K., Nag, A.: Wearable sensors and systems in the iot (2021)

[137] Mursi, K.T., Thapaliya, B., Zhuang, Y., Aseeri, A.O., Alkatheiri, M.S.: A fast deep learning method for security vulnerability study of xor pufs. Electronics **9**(10), 1715 (2020)

[138] Nayak, S., Patgiri, R.: Robustbf: A high accuracy and memory efficient 2d bloom filter. arXiv preprint arXiv:2106.04365 (2021)

[139] Ndjiongue, A.R., Ferreira, H.C., Ngatched, T.M.: Visible light communications (vlc) technology. Wiley Encyclopedia of Electrical and Electronics Engineering pp. 1–15 (1999)

[140] nearfieldcommunication.org: How nfc works. http://nearfieldcommunication.org/how-it-works.html (Dec 2015)

[141] News, T.H.: Casino gets hacked through its internet-connected fish tank thermometer (2018), `https://thehackernews.com/2018/04/iot-hacking-thermometer.html`

[142] Nguyen, L.H., Roscoe, A.W.: Authentication protocols based on low-bandwidth unspoofable channels: a comparative survey. Journal of Computer Security **19**(1), 139–201 (2011)

[143] Nguyen, P.H., Sahoo, D.P., Jin, C., Mahmood, K., Rührmair, U., van Dijk, M.: The interpose puf: Secure puf design against state-of-the-art machine learning attacks. IACR Transactions on Cryptographic Hardware and Embedded Systems **2019**(4), 243–290 (Aug 2019). https://doi.org/10.13154/tches.v2019.i4.243-290, `https://tches.iacr.org/index.php/TCHES/article/view/8351`

[144] Nguyen, T., Leneutre, J.: Formal analysis of secure device pairing protocols. In: 2014 IEEE 13th International Symposium on Network Computing and Applications. pp. 291–295. IEEE (2014)

[145] Nguyen, T., Leneutre, J.: A secure and effective device pairing protocol. In: 2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC). pp. 507–512. IEEE (2015)

[146] Oncu, A., Fujishima, M.: Millimeter-wave cmos impulse radio. Advances in Solid State Circuit Technologies pp. 255–288 (2010)

[147] Open Connectivity Foundation, I.: Ocf security specification (2017), `https://openconnectivity.org/specs/OCF_Security_Specification_v1.0.0.pdf`

[148] Pal, K., et al.: Internet of things and blockchain technology in apparel manufacturing supply chain data management. Procedia Computer Science **170**, 450–457 (2020)

[149] Pasini, S., Vaudenay, S.: Sas-based authenticated key agreement. In: International Workshop on Public Key Cryptography. pp. 395–409. Springer (2006)

[150] Patwari, N., Croft, J., Jana, S., Kasera, S.K.: High-rate uncorrelated bit extraction for shared secret key generation from channel measurements. IEEE Transactions on Mobile Computing **9**(1), 17–30 (2009)

[151] Peltonen, A., Sethi, M., Aura, T.: Formal verification of misbinding attacks on secure device pairing and bootstrapping. Journal of Information Security and Applications **51**, 102461 (2020)

[152] Perli, S.D., Ahmed, N., Katabi, D.: Pixnet: Interference-free wireless links using lcd-camera pairs. In: Proceedings of the sixteenth annual international conference on Mobile computing and networking. pp. 137–148. ACM (2010)

[153] Pour, A.A., Beroulle, V., Cambou, B., Danger, J.L., Di Natale, G., Hely, D., Guilley, S., Karimi, N.: Puf enrollment and life cycle management: Solutions and perspectives for the test community. In: 2020 IEEE European Test Symposium (ETS). pp. 1–10. IEEE (2020)

[154] von elektrischen PUF, K.: Cryptanalysis of electrical pufs via machine learning algorithms

[155] Rabcan, J., Levashenko, V., Zaitseva, E., Kvassay, M., Subbotin, S.: Application of fuzzy decision tree for signal classification. IEEE Transactions on Industrial Informatics **15**(10), 5425–5434 (2019)

[156] Rahim, M.A., Rahman, M.A., Rahman, M.M., Asyhari, A.T., Bhuiyan, M.Z.A., Ramasamy, D.: Evolution of iot-enabled connectivity and applications in automotive industry: A review. Vehicular Communications **27**, 100285 (2021)

[157] Rahman, M., Topkara, U., Carbunar, B.: Seeing is not believing: Visual verifications through liveness analysis using mobile devices. In: Proceedings of the 29th Annual Computer Security Applications Conference. pp. 239–248. ACM (2013)

[158] Riedmiller, M., Braun, H.: A direct adaptive method for faster backpropagation learning: The rprop algorithm. In: IEEE international conference on neural networks. pp. 586–591. IEEE (1993)

[159] Rivest, R.L.: The rc5 encryption algorithm. In: International Workshop on Fast Software Encryption. pp. 86–96. Springer (1994)

[160] Roeschlin, M., Martinovic, I., Rasmussen, K.B.: Device pairing at the touch of an electrode. In: NDSS. vol. 18, pp. 18–21 (2018)

[161] Roman, R., Lopez, J.: Keyled-transmitting sensitive data over out-of-band channels in wireless sensor networks. In: 2008 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems. pp. 796–801. IEEE (2008)

[162] Rostami, M., Majzoobi, M., Koushanfar, F., Wallach, D.S., Devadas, S.: Robust and reverse-engineering resilient puf authentication and key-exchange by substring matching. IEEE Transactions on Emerging Topics in Computing **2**(1), 37–49 (2014)

[163] Roy, N., Gowda, M., Choudhury, R.R.: Ripple: Communicating through physical vibration. In: 12th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 15). pp. 265–278 (2015)

[164] Ruaro, A., Thaysen, J., Jakobseny, K.B.: Head-centric body-channel propagation paths characterization. In: 2015 9th European Conference on Antennas and Propagation (EuCAP). pp. 1–4. IEEE (2015)

[165] Rührmair, U., Sehnke, F., Sölter, J., Dror, G., Devadas, S., Schmidhuber, J.: Modeling attacks on physical unclonable functions. In: Proceedings of the 17th ACM conference on Computer and communications security. pp. 237–249 (2010)

[166] Ruhrmair, U., Solter, J.: Puf modeling attacks: An introduction and overview. In: 2014 Design, Automation & Test in Europe Conference & Exhibition (DATE) (2014)

[167] Rührmair, U., Sölter, J., Sehnke, F., Xu, X., Mahmoud, A., Stoyanova, V., Dror, G., Schmidhuber, J., Burleson, W., Devadas, S.: Puf modeling attacks on simulated and silicon data. IEEE transactions on information forensics and security **8**(11), 1876–1891 (2013)

[168] Salman, O., Abdallah, S., Elhajj, I.H., Chehab, A., Kayssi, A.: Identity-based authentication scheme for the internet of things. In: 2016 IEEE Symposium on Computers and Communication (ISCC). pp. 1109–1111. IEEE (2016)

[169] Santikellur, P., Bhattacharyay, A., Chakraborty, R.S.: Deep learning based model building attacks on arbiter puf compositions. Cryptology ePrint Archive, Report 2019/566 (2019), `https://ia.cr/2019/566`

[170] Sarikaya, B., Sethi, M., Garcia-Carillo, D.: Secure IoT Bootstrapping: A Survey. Internet-Draft draft-sarikaya-t2trg-sbootstrapping-05, `https://datatracker.ietf.org/doc/html/draft-sarikaya-t2trg-sbootstrapping-05`, work in Progress

[171] Saxena, N., Ekberg, J.E., Kostiainen, K., Asokan, N.: Secure device pairing based on a visual channel. In: 2006 IEEE Symposium on Security and Privacy (S&P'06). pp. 6–pp. IEEE (2006)

[172] Saxena, N., Uddin, M.B., Voris, J., Asokan, N.: Vibrate-to-unlock: Mobile phone assisted user authentication to multiple personal rfid tags. In: 2011 IEEE International Conference on Pervasive Computing and Communications (PerCom). pp. 181–188. IEEE (2011)

[173] Scannell, A., Varshavsky, A., LaMarca, A., De Lara, E.: Proximity-based authentication of mobile devices. International Journal of Security and Networks **4**(1-2), 4–16 (2009)

[174] Schertler, M.J., Boyen, X.: Identity-Based Cryptography Standard (IBCS) #1: Supersingular Curve Implementations of the BF and BB1 Cryptosystems. RFC 5091 (Dec 2007). https://doi.org/10.17487/RFC5091, `https://www.rfc-editor.org/info/rfc5091`

[175] Schürmann, D., Sigg, S.: Secure communication based on ambient audio. IEEE Transactions on mobile computing **12**(2), 358–370 (2011)

[176] Sen, J.: Security in wireless sensor networks. Wireless Sensor Networks: Current Status and Future Trends **407**, 408 (2012)

[177] Sethi, M., Peltonen, A., Aura, T.: Misbinding attacks on secure device pairing and bootstrapping. In: Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security. pp. 453–464. ACM (2019)

[178] Sethi, M., Sarikaya, B., Garcia-Carrillo, D.: Secure IoT Bootstrapping: A Survey. Internet-Draft draft-sarikaya-t2trg-sbootstrapping-11 (Feb 2021), `https://datatracker.ietf.org/doc/html/draft-sarikaya-t2trg-sbootstrapping-11`, work in Progress

[179] Shamir, A.: How to share a secret. Commun. ACM **22**(11), 612–613 (nov 1979). https://doi.org/10.1145/359168.359176, `https://doi.org/10.1145/359168.359176`

[180] Shelby, Z., Hartke, K., Bormann, C.: The Constrained Application Protocol (CoAP). RFC 7252 (Jun 2014). https://doi.org/10.17487/RFC7252, `https://rfc-editor.org/rfc/rfc7252.txt`

[181] Shrestha, B., Saxena, N., Truong, H.T.T., Asokan, N.: Sensor-based proximity detection in the face of active adversaries. IEEE Transactions on Mobile Computing **18**(2), 444–457 (2018)

[182] Soriente, C., Tsudik, G., Uzun, E.: Hapadep: human-assisted pure audio device pairing. In: International Conference on Information Security. pp. 385–400. Springer (2008)

[183] Soriente, C., Uzun, E.: Beda : Button-enabled device association (2007)

[184] Steinmetzer, D., Chen, J., Classen, J., Knightly, E., Hollick, M.: Eavesdropping with periscopes: Experimental security analysis of highly directional millimeter waves. In: 2015 IEEE Conference on Communications and Network Security (CNS). pp. 335–343. IEEE (2015)

[185] Suh, G.E., Devadas, S.: Physical unclonable functions for device authentication and secret key generation. In: 2007 44th ACM/IEEE Design Automation Conference. pp. 9–14. IEEE (2007)

[186] Sun, D.Z., Mu, Y., Susilo, W.: Man-in-the-middle attacks on secure simple pairing in bluetooth standard v5. 0 and its countermeasure. Personal and Ubiquitous Computing **22**(1), 55–67 (2018)

[187] Tanaka, H.: Information leakage via electromagnetic emanations and evaluation of tempest countermeasures. In: International Conference on Information Systems Security. pp. 167–179. Springer (2007)

[188] Tebelmann, L., Danger, J.L., Pehl, M.: Self-secured puf: protecting the loop puf by masking. In: International Workshop on Constructive Side-Channel Analysis and Secure Design. pp. 293–314. Springer (2020)

[189] Tobisch, J., Becker, G.T.: On the scaling of machine learning attacks on pufs with application to noise bifurcation. In: International Workshop on Radio Frequency Identification: Security and Privacy Issues. pp. 17–31. Springer (2015)

[190] Torrey, L., Shavlik, J.: Transfer learning. In: Handbook of research on machine learning applications and trends: algorithms, methods, and techniques, pp. 242–264. IGI global (2010)

[191] Varshavsky, A., Scannell, A., LaMarca, A., De Lara, E.: Amigo: Proximity-based authentication of mobile devices. In: International Conference on Ubiquitous Computing. pp. 253–270. Springer (2007)

[192] Vaudenay, S.: Secure communications over insecure channels based on short authenticated strings. In: Annual International Cryptology Conference. pp. 309–326. Springer (2005)

[193] Wang, S.C.: Artificial neural network. In: Interdisciplinary computing in java programming, pp. 81–100. Springer (2003)

[194] Wang, T., Kerschbaum, F.: Robust and undetectable white-box watermarks for deep neural networks. arXiv preprint arXiv:1910.14268 (2019)

[195] Wang, X., Haynes, R.D., He, Y., Feng, Q.: Well control optimization using derivative-free algorithms and a multiscale approach. Computers & Chemical Engineering **123**, 12–33 (2019)

[196] Wankhede, K., Wukkadada, B., Nadar, V.: Just walk-out technology and its challenges: A case of amazon go. In: 2018 International Conference on Inventive Research in Computing Applications (ICIRCA). pp. 254–257. IEEE (2018)

[197] Weimerskirch, A., Paar, C., Wolf, M.: Cryptographic component identification: Enabler for secure vehicles. In: IEEE Vehicular Technology Conference. vol. 62, p. 1227. IEEE; 1999 (2005)

[198] Weis, S.A.: Rfid (radio frequency identification): Principles and applications. System **2**(3), 1–23 (2007)

[199] Wisiol, N., Gräbnitz, C., Mühl, C., Zengin, B., Soroceanu, T., Pirnay, N.: pypuf: Cryptanalysis of Physically Unclonable Functions (2021). https://doi.org/10.5281/zenodo.3901410, `https://doi.org/10.5281/zenodo.3901410`

[200] Wisiol, N., Mühl, C., Pirnay, N., Nguyen, P.H., Margraf, M., Seifert, J.P., van Dijk, M., Rührmair, U.: Splitting the interpose puf: A novel modeling attack strategy. IACR Transactions on Cryptographic Hardware and Embedded Systems **2020**(3), 97–120 (Jun 2020). https://doi.org/10.13154/tches.v2020.i3.97-120, `https://tches.iacr.org/index.php/TCHES/article/view/8584`

[201] Wong, F.L., Stajano, F.: Multichannel security protocols. IEEE Pervasive Computing **6**(4), 31–39 (2007)

[202] Wu, Y., Chen, B., Zhao, Z., Cheng, Y.: Attack and countermeasure on interlock-based device pairing schemes. IEEE Transactions on Information Forensics and Security **13**(3), 745–757 (2017)

[203] Xi, W., Li, X.Y., Qian, C., Han, J., Tang, S., Zhao, J., Zhao, K.: Keep: Fast secret key extraction protocol for d2d communication. In: 2014 IEEE 22nd International Symposium of Quality of Service (IWQoS). pp. 350–359. IEEE (2014)

[204] Xi, W., Qian, C., Han, J., Zhao, K., Zhong, S., Li, X.Y., Zhao, J.: Instant and robust authentication and key agreement among mobile devices. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. pp. 616–627. ACM (2016)

[205] Yamaguchi, S., Gupta, B.: Malware threat in internet of things and its mitigation analysis. In: Research Anthology on Combating Denial-of-Service Attacks, pp. 371–387. IGI Global (2021)

[206] Yilmaz, Y., Gunn, S.R., Halak, B.: Lightweight puf-based authentication protocol for iot devices. In: 2018 IEEE 3rd International Verification and Security Workshop (IVSW). pp. 38–43. IEEE (2018)

[207] Yu, M.D., M'Raïhi, D., Verbauwhede, I., Devadas, S.: A noise bifurcation architecture for linear additive physical functions. In: 2014 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST). pp. 124–129. IEEE (2014)

[208] Zafer, M., Agrawal, D., Srivatsa, M.: Limitations of generating a secret key using wireless fading under active adversary. IEEE/ACM Transactions on Networking **20**(5), 1440–1451 (2012)

[209] Zhang, B., Ren, K., Xing, G., Fu, X., Wang, C.: Sbvlc: Secure barcode-based visible light communication for smartphones. IEEE Transactions on Mobile Computing **15**(2), 432–446 (2015)

[210] Zhang, J., Gu, Z., Jang, J., Wu, H., Stoecklin, M.P., Huang, H., Molloy, I.: Protecting intellectual property of deep neural networks with watermarking. In: Proceedings of the 2018 on Asia Conference on Computer and Communications Security. pp. 159–172 (2018)

[211] Zhang, J., Shen, C.: Set-based obfuscation for strong pufs against machine learning attacks. IEEE Transactions on Circuits and Systems I: Regular Papers **68**, 288–300 (2021). https://doi.org/10.1109/TCSI.2020.3028508

[212] Zhou, R., Xing, G.: nshield: a noninvasive nfc security system for mobiledevices. In: Proceedings of the 12th annual international conference on Mobile systems, applications, and services. pp. 95–108. ACM (2014)

[213] Zimmer, V., Krau, M.: Establishing the root of trust (2016)

ECOLE DOCTORALE

Titre : Amorçage de la sécurité dans les réseaux IoT

Mots clés : Internet des objets, sécurité, appairage sécurisé, enrolement sécurisé, mécanisme de mise en accord sur une clé, fonctions physiques non clonables

Résumé : La demande de services qui se basent sur l'Internet des objets (IoT) augmente de manière exponentielle. Ces objets connectés sont considérés comme une partie essentielle des processus industriels de nombreux secteurs d'activités. Cependant, ces dispositifs peuvent représenter une menace pour la sécurité du réseau de déploiement et un point d'entrée potentiel pour des adversaires. Il existe donc un besoin imminent de réaliser une approche d'association sécurisée des objets connectés avant qu'ils ne soient rendus opérationnels sur le réseau de l'utilisateur. Cette procédure, appelée "amorçage de la sécurité", garantit en premier lieu la confidentialité et l'intégrité des échanges de données entre l'utilisateur et les dispositifs. Ensuite, ce processus fournit une assurance sur l'identité et l'origine de ces objets.

En raison des limites d'évolutivité, la première phase du processus d'amorçage ne peut pas être menée efficacement en utilisant des connaissances de sécurité pré-partagées telles que des certificats numériques. Cette étape d'appairage assure l'établissement d'un canal de communication sécurisé entre l'utilisation et l'objet. La phase d'appairage utilise un protocole d'accord de clé symétrique qui est adapté à la nature de ces dispositifs à ressources limitées. L'utilisation de canaux auxiliaires a été proposée comme moyen d'authentifier l'échange de clés, mais elle nécessite un temps relativement long et une participation importante de l'utilisateur pour transférer les bits d'authentification. Cependant, les systèmes basés sur le contexte utilisent l'environnement ambiant pour extraire un secret commun sans intervention importante de l'utilisateur, à condition d'avoir un périmètre sécurisé pendant la phase d'extraction, ce qui est considéré comme une hypothèse de sécurité forte.

La deuxième phase du processus d'amorçage est appelée "enrôlement sécurisée" et vise à éviter l'association d'un objet IoT malveillant en authentifiant son identité et son origine. L'utilisation d'éléments de sécurité matériels, tels que les fonctions physiques non clonables (PUF), a été présentée comme une solution prometteuse adaptée à la nature limitée des ressources de ces dispositifs. Un nombre croissant d'architectures PUF ont été démontrées mathématiquement clonables grâce à des techniques de modélisation par apprentissage automatique. L'utilisation de modèles de PUF a été récemment proposée pour authentifier les objets IoT. Cette procédure facilite l'évolutivité du processus d'authentification en réduisant l'espace de stockage requis pour chaque dispositif. Néanmoins, le scénario de fuite du modèle PUF vers un adversaire en raison d'une menace interne au sein de l'organisation n'est pas pris en charge par les solutions existantes. Par conséquent, la sécurité de ces propositions d'inscription basées sur le modèle PUF peut être compromise. Dans cette thèse, nous étudions le processus d'amorçage de la sécurité des objets connectés à ressources limitées et nous introduisons deux protocoles :

I. Un protocole hybride d'appairage, appelé COOB, qui combine d'une manière efficace un schéma d'appairage contextuel avec l'utilisation d'un canal auxiliaire. Ce protocole exploite une technique d'exponentiation spécifiques des clés publiques Diffie-Hellman en utilisant des nonces pour atteindre l'objectif de secret temporaire nécessaire à l'accord de clé. Notre méthode assure la sécurité même contre un attaquant qui peut contrôler la zone de sécurité (un environnement hostile), ce qui n'est pas pris en charge par les schémas contextuels existants. Cette amélioration de la sécurité a été formellement validée dans le modèle symbolique en utilisant l'outil de vérification formelle TAMARIN.

II. Une solution d'enrôlement qui exploite un modèle de PUF dans le processus d'authentification, appelé Water-PUF. Notre protocole est basé sur une technique de tatouage numérique spécialement conçue pour les modèles PUF. Cette procédure empêche un adversaire de s'appuyer sur le modèle tatoué ou sur un autre modèle dérivé pour contourner l'authentification. Par conséquent, toute fuite du modèle PUF filigrané utilisé pour l'enrôlement n'affecte pas l'exactitude du protocole. La conception du Water-PUF est validée par un certain nombre de simulations contre de nombreuses attaques de suppression de tatouage numérique afin d'évaluer la robustesse de notre proposition.

Title : Secure bootstrapping for Internet of Things

Keywords : Internet of Things, security, secure device pairing, secure device enrollment, key agreement techniques, physical unclonable functions

Abstract : The demand for IoT services is increasing exponentially. These connected objects are considered as an essential part of the business processes of numerous industry sectors. However, these devices can represent a serious threat to the security of the deployment network and a potential entry-point when exploited by the adversaries. Thus, there is an imminent need to perform a secure association approach of the IoT objects before being rendered operational on the network of the user. This procedure is referred to as secure bootstrapping and it primarily guarantees the confidentiality and the integrity of the data exchanges between the user and the devices. Secondly, this process provides an assurance on the identity and the origin of these objects.

Due to scalability limitations, the first phase of the bootstrapping process cannot be efficiently conducted using pre-shared security knowledge such as digital certificates. This step is referred to as secure device pairing and it ensures the establishment of a secure communication channel between the use and the object. The pairing phase uses an ad-hoc symmetric key agreement protocol that is suitable to the resource-constrained nature of these devices. The use of auxiliary channels has been proposed as a way to authenticate the key exchange but they require a relatively long time and an extensive user involvement to transfer the authentication bits. However, the context-based schemes use the ambient environment to extract a common secret without an extensive user intervention under the requirement of having a secure perimeter during the extraction phase, which is considered a strong security assumption.

The second phase of the bootstrapping process is referred to as secure device enrollment and it aims at avoiding the associating of a malicious IoT object by authenticating its identity. The use of hardware security elements, such as the PUF, has been introduced as a promising solution that is suitable for the resource-constraint nature of these devices. A growing number of PUF architectures has been demonstrated mathematically clonable through ML modeling techniques. The use of ML PUF models has been recently proposed to authenticate the IoT objects. This procedure facilitates the scalability of the authentication process by reducing the storage space required for each device. Nonetheless, the leakage scenario of the PUF model to an adversary due to an insider threat within the organization is not supported by the existing solutions. Hence, the security of these PUF model-based enrollment proposals can be compromised.

In this thesis, we study the secure bootstrapping process of resource-constrained devices and we introduce two security schemes:

I. A hybrid ad-hoc pairing protocol, called COOB, that efficiently combines a state-of-the-art fast context-based scheme with the use of an auxiliary channel. This protocol exploits a nonce exponentiation of the Diffie-Hellman public keys to achieve the temporary secrecy goal needed for the key agreement. Our method provides security even against an attacker that can violate the safe zone requirement, which is not supported by the existing contextual schemes. This security improvement has been formally validated in the symbolic model using the TAMARIN prover.

II. An enrollment solution that exploits a ML PUF model in the authentication process, called Water-PUF. Our enrollment scheme is based on a specifically designed black-box watermarking technique for PUF models with a binary output response. This procedure prevents an adversary from relying on the watermarked model in question or another derivative model to bypass the authentication. Therefore, any leakage of the watermarked PUF model that is used for the enrollment does not affect the correctness of the protocol. The Water-PUF design is validated by a number of simulations against numerous watermark suppression attacks to assess the robustness of our proposal.

Institut Polytechnique de Paris
91120 Palaiseau, France