Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 8887472, 30 pages
https://doi.org/10.1155/2021/8887472

Review Article

Security Analysis of Out-of-Band Device Pairing Protocols: A Survey

Sameh Khalfaoui,[1,2] **Jean Leneutre**,[1] **Arthur Villard**,[2] **Jingxuan Ma**,[2] **and Pascal Urien**[1]

[1]*LTCI, Télécom Paris, Institut Polytechnique de Paris, France*
[2]*EDF R&D, France*

Correspondence should be addressed to Sameh Khalfaoui; sameh.khalfaoui@edf.fr

Received 5 August 2020; Revised 28 October 2020; Accepted 24 November 2020; Published 30 January 2021

Academic Editor: Qi Jiang

Numerous secure device pairing (SDP) protocols have been proposed to establish a secure communication between unidentified IoT devices that have no preshared security parameters due to the scalability requirements imposed by the ubiquitous nature of the IoT devices. In order to provide the most user-friendly IoT services, the usability assessment has become the main requirement. Thus, the complete security analysis has been replaced by a sketch of a proof to partially validate the robustness of the proposal.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 paper, 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. Furthermore, we refine the adversary capabilities on the out-of-band channel by redefining the replay capability and by introducing a new notion of delay that is dependent on the protocol structure that is more adequate for the ad hoc pairing context. Also, we propose a classification of a number of out-of-band channels based on their security properties and under our refined adversary model. Our work motivates the future SDP protocol designer to conduct a formal or a computational security assessment to allow the comparability between these pairing techniques. Furthermore, it provides a realistic abstraction of the adversary capabilities on the out-of-band channel which improves the modeling of their security characteristics in the protocol verification tools.

1. Introduction

With the growing demand for IoT objects for both the personal and the industrial contexts, the use of a decentralized device-to-device (D2D) communication system has become a necessity for numerous applications in the context of Internet of Things (IoT). This decision is based on the inefficiency of a centralized communication solution to meet the scalability and the interoperability goals. Therefore, the protection of this communication channel requires the use of a secure key establishment protocol between the devices, known as *secure device pairing* (SDP). This process ensures that the communicating nodes agree on the same symmetric encryption key, which represents an initial trust establishment between devices that have no preshared knowledge (a certificate, a shared password, or a symmetric key). The no prior secret condition is motivated by two reasons: the unfeasibility of exploiting a public key infrastructure (PKI) due to the growing numbers of heterogeneous IoT devices, and the *zero-trust* policy that disapproves of trusting the manufacturer with delivering the initial preshared pairing keys to avoid any vulnerabilities or breaches related to a third party.

Numerous secure device pairing solutions have been proposed to securely establish a shared key between a number of

devices that do not share any prior security knowledge. These techniques can be divided into two main categories. The first one ensures the confidentiality and the data authentication of the key through a proof of copresence based on the randomness of the ambient environment and it is better known as context-based pairing or zero-interaction protocols (ZIP) [1, 2]. The second technique relies on an auxiliary channel with specific security properties to send an information that validates what has been exchanged on the main insecure channel, referred to as the in-band channel. However, in this state of art, we will only discuss the security analysis of the out-of-band secure device pairing schemes that rely on an auxiliary channel [3–5].

The use of the secondary channel is due to the unfeasibility of performing the authentication based on a single channel that is controlled by a Dolev-Yao intruder [6], as demonstrated in [7] using BAN logic analysis [8]. 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 the cryptographic functions. As a consequence of adopting this intruder model, the usage of the main insecure channel without having preshared 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. Due to their special nature and their communication properties, they provide an initial level of security that is sufficient to primarily guarantee the integrity, the data origin authenticity, and the demonstrative identification [9], 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. These assumptions on the OoB channel reduce the attacker capabilities in comparison with his abilities on the main insecure channel. On the other hand, there is another variant of secure device pairing schemes that uses the randomness of the ambient environment in order to securely establish a shared key between the intended devices. These protocols might rely on external factors with respect to the human user such as the radio environment [10–12], the acoustic surroundings [13, 14], or other random physical patterns [15–18]. However, numerous context-based pairing research works in the field of wireless body area network (WBAN) rely on specific human-centric biometrics that are extracted by the sensors attached to the user which is more suitable for the implantable medical devices (IMD) [19–22]. These collected random features are used as the secure element in the protocol execution. Nonetheless, the evaluation of these contextual pairing schemes is considered out of the scope of this work. Readers eager to learn more about these protocols and their applications can consult these review articles [1, 2, 23].

In the literature, a variety of surveys [3–5] have addressed the out-of-band pairing problem from the security perspective. In the work of Nguyen and Roscoe [5], the authors conducted a study on the authentication process involving secure device pairing schemes that rely on a manual transfer of a short authentication string (SAS). They discussed the costs related to the cryptographic techniques applied in the protocol design and the required communication between the pairing participants. However, this work proposed a classification of the out-of-band channels according to some assumptions about their threat models which appeared to us to be unrealistic in some cases such as the feasibility of a delay attack. In the work of Mirzadeh et al. [4], the authors extended the work of Nguyen and Roscoe [5] by conducting an extensive survey on a number of pairwise and groupwise device pairing protocols based on a similar classification of the out-of-band channels. Although the work tends to mention the results of the conducted formal or the computational security proofs, it does not describe the evaluated properties nor discuss their associated assumptions, and as a consequence, it does not offer a complete basis to compare the provided security of the different protocol. In addition, a great body of work on SDP tends to investigate the same authentication and confidentiality properties under different definitions that drift away from the commonly known specifications such as the ones given in the work of Lowe [24]. Therefore, these verification results are difficult to interpret. Furthermore, the security analysis using the protocol verification tools has not been discussed even though multiple research works [25–27] have adopted these formal methods to evaluate the security of their proposals based on a predefined set of authentication properties. Also, we have noticed that numerous SDP schemes are based on a threat model, inspired from the Dolev-Yao intruder capabilities [6]. This model allows the adversary to replay messages on the out-of-band channel while guaranteeing the integrity of the exchanged information. On the other hand, the act of forging a message that pleases the attacker is deemed unfeasible and will be, somehow, detected. These two assumptions might be plausible when the two devices have a preshared secret that is used to sign the OoB messages which force the attacker to only replay previous exchanges. Unfortunately, this is not the case for the ad hoc secure device pairing due to the lack of preshared security knowledge between the pairing participants. Thus, it makes these assumptions not valid and it might lead to vulnerabilities when the scheme is deployed. Furthermore, the previously described intruder model assumes that the attacker is able to delay any out-of-band transmission for a desired given time. In the context of a direct communication channel, this specific action is highly dependent on the feasibility of blocking a message and replaying it afterwards. Therefore, if the replay attack is not considered feasible, then the delay assumption is no longer valid. In the work of Fomichev et al. [3], the authors have provided a systematic modeling of the pairwise pairing procedure by describing its three main components: the out-of-band channel, the user involvement, and the pairing context. Also, they outlined the characteristics of the OoB channels by detailing their communication properties, by summarizing some of their known vulnerabilities and by identifying some of their main usability advantages in the IoT context. However, their analysis does not give a detailed security assessment of the SDP schemes. The focus in their

analysis of the protocols is more oriented toward the usability aspects than the security. Even though our main focus is related to the formal or computational security assessment of numerous SDP schemes, we point out the importance of enhancing the usability of these pairing processes in order to facilitate their ease-of-adoption. Readers eager to learn more about the usability and the human-in-the-loop aspect in the secure device pairing procedure can consult these review papers [3, 28–30].

In this work, we focus on providing a comprehensive study on the existing formal and computational security proofs that are conducted on a selection of secure device pairing schemes. This review clearly 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 its own terminologies and its own definitions, we normalize the used taxonomy in order to enhance the understanding of these security validations. Also, we refine the adversary model that has been adopted by multiple pairing proposals by eliminating the replay capability and by introducing a new notion of delay that is based on the protocol structure rather than the out-of-band channel characteristics. These modifications are motivated by the urge to have a security model that is adequate to the ad hoc device pairing context and assumptions in order to facilitate their validation and deployment in a realistic scenario. Based on our security model, we classified a selection of out-of-band channels based on an evaluation of their achieved security goals. In addition, we describe an advanced threat model that consists of violating two security guarantees: *the demonstrative identification* and *the device integrity* (the latter property outlines that one of the pairing participants is under the control of the adversary). This adversary model has yield a recently published attack, called *misbinding* [27], that targets the majority of the device pairing schemes.

This work is aimed 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.

The main contributions of this paper are summarized as follows:

(i) We conduct a comprehensive study on the existing formal and computational security proofs that evaluate a selection of secure device pairing schemes relying on an out-of-band channel

(ii) We enhance the threat model, adopted by numerous SDP proposals to describe the attacker action on the OoB channel, by eliminating the replay assumption and by introducing a new realistic approach to the delay attack based on the structure of the protocol. Then, we derive six categories of the out-of-band channels based on their achieved security goals in our threat model

(iii) We conduct a classification of a commonly used OoB channels based on the security categories derived previously

(iv) We discuss the recently published misbinding attack by explaining its origin, the adopted adversary model, and some of the proposed mitigations

(v) We provide a number of secure pairing design recommendations for future SDP designers and we highlight a number of future challenges, based on identified security weaknesses, where SDP research is demanded

The rest of the paper is organized as follows. Section 2 focuses on the out-of-band channels by describing the limitations of the widely adopted OoB adversary model and it presents our enhancement proposals with respect to the attacker capabilities and the security guarantees that should be evaluated. Also, it discusses the security and the usability properties of a selection of the commonly used out-of-band channels. In addition, it provides a classification based on our refined threat model. Section 3 describes a number of SDP schemes that have been either formally or computationally verified. Thus, other SDP proposals with only a sketch of a security proof are considered out of the scope of this work. Furthermore, it discusses an advanced threat model that assumes that one of the pairing participants is compromised and that the user unintentionally initiates the pairing with a malicious device. These assumptions have been demonstrated feasible and they lead to a misbinding attack that falsely establishes the pairing with a distant malicious object. Also, it focuses on a number of common vulnerabilities and security considerations when designing a pairing protocol that is based on an out-of-band channel. Section 4 highlights four main aspects: the most common design vulnerabilities in the out-of-band pairing protocols, the recommendations of the necessary mitigations, a description of the limitations of the security analysis conducted on the SDP schemes, and the future areas that need to be further studied regarding this matter. Lastly, Section 5 concludes our work.

2. Out-of-Band Channel Overview

2.1. Refined Out-of-Band Threat Model. In this study, we adopt the Dolev-Yao intruder model [6] 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 preestablished security information, the attacker has the same level of knowledge as the legitimate devices which eliminate any possibility of performing a secure key establishment using only the in-band channel, as proved in [7] using BAN logic analysis [8].

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* [4], is demanded by some SDP schemes ([31, 32]). This assumption is hard to obtain and might ultimately lead to vulnerabilities in the protocol design [9]. 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* [4]. 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* [4].

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 preshared secret is ultimately vulnerable. Therefore, while considering the presence of a vigilant user, we will 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 will result in stopping the protocol execution for the same amount of time which, consequently, will lead 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* [9], starts by a bidirectional OoB exchange of the public key hashes which, according to our model, 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:

(i) Confidentiality (C) [33]: the information, sent over the channel, can only be accessed by the authorized pairing parties. Therefore, the attacker cannot overhear the communication

(ii) Data freshness (DF) [33]: the information, sent over the channel, cannot be replayed by a malicious actor. Therefore, the attacker cannot inject any old messages on the channel

(iii) 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

(iv) Liveness (L) [34]: any information, transmitted over the channel, is eventually received by the intended party. Therefore, the attacker cannot block any transmission over the channel

(v) 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 will have six main channel types:

(i) Confidential OoB: all the security goals are guaranteed. Therefore, the adversary has no capabilities

(ii) Delayable-confidential OoB: only the channel availability assumption is not guaranteed. Therefore, the adversary can only delay the transmission

(iii) Protected OoB: only the confidentiality goal does not hold. This means that the attacker is only capable of overhearing the communication

(iv) 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

(v) 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

(vi) 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 1.

2.2. Out-of-Band Security Classification. The majority of the existing pairing solutions rely on an auxiliary channel with

Table 1: 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	✓	✓	✗	✓	✗	✓	✓	✓	✗	✗

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 [8]. As proven in [7], "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, using only the main insecure channel is not sufficient. Thus, there is a need for an auxiliary channel on which the authentication of the exchanged keys can happen. Known as *out-of-band* (OoB) channel, *location limited channel* (LLC), or *side channels* [9], these communication links can be constructed based on audio, visual, or haptic transmissions ([31, 35, 36]) 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. [3], 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. [30].

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 link itself known as human-computer interaction (HCI) channel [3]. 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 [29].

In this section, we will 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 will be briefly introducing 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 Subsection 2.1, will be used to classify these chosen channels based on the security they offer while taking into account the presence of a vigilant user, as summarized in Table 2.

2.2.1. Near-Field Communication (NFC). 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 1. These devices can be active or passive [37]. NFC chips are widely deployed and they are used in a wide variety of IoT devices.

(1) 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 nonperceptibility 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 nonexpert users. This requirement represents a burden on the user and a drawback when it comes to the user friendliness aspect.

(2) Security Properties. The devices using NFC chips can be active in order to act as a contactless card reader or communicate with another object. It 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 [38].

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 [39] 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 [40] between two NFC-enabled devices separated by a 10 cm distance. The fact that the security is provided based on a proximity assumption, an attacker can always violate such

Table 2: 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

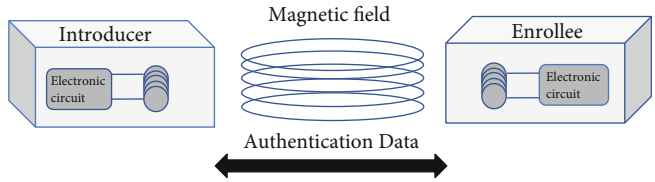

Figure 1: Communication model of a NFC technology.

requirement which does not make this out-of-band channel any better than the in-band channel because of its similar communication properties.

(3) Proposed Schemes.

(1) Push-button configuration (PBC) is part of the standardized Wi-Fi protected setup (WPS) [41] that introduces a pairing scheme using two options:

(i) Password token: the enrollee device will transmit a 32-byte random password to the NFC-enabled registrar. The same password will be used with the in-band registration protocol to provision the enrollee with WLAN configuration data

(ii) 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

(2) Secure Simple Pairing (SSP) is part of the standardized Bluetooth Secure Simple Pairing [42] that introduces a pairing scheme using an out-of-band option:

(i) 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 which has been reported to be resistant against MitM attacks

2.2.2. Radio Frequency Identification Channel (RFID). RFID is a wireless communication technology used for both indoor and outdoor identifications. These systems consist of small *tags* that emit stored identification information when interrogated by an RFID *reader* which makes them a sort of an automatic identification system [43]. 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. We can find *active* tags having on-board their own power supply which makes them able to establish a bidirectional communication channel.

(1) Usability Properties. This technology does not require any human intervention in the case of the high frequencies which make it more user-friendly and more appealing to nonexpert users. On the other hand, for the low frequencies, it has the same requirements as the NFC technology, described in Subsection 2.2.1.

(2) Security Properties. For the low frequencies, RFID has similar security properties to the NFC technology stated in Subsection 2.2.1.

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 [43]. 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 2.1.

(3) Proposed Schemes. Noisy tag [45] is the injection of intentional noise, using an extra RFID tag (noisy tag), into an authentic channel making the eavesdropping process meaningless for the adversary. Only the legitimate reader (owner of the noisy tag) will be able to retrieve the original message from the noise-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

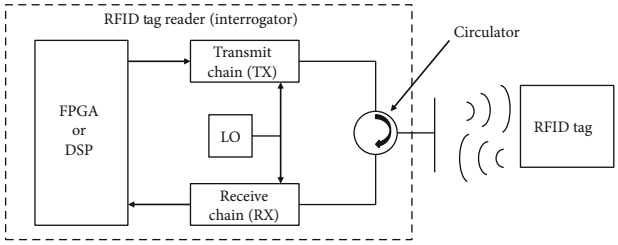

Figure 2: Block diagram of a RFID communication system [44].

barriers, e.g., in private areas (user surveillance, house, office, and building).

2.2.3. Millimeter Waves (MM-Waves). MM-waves is a wireless communication operating 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 [46]. A standard IEEE 802.11ad [47] enables multigigabit wireless communications in the unlicensed 60 GHz band [48], as shown in Figure 3. This band is considered ideal for a variety of indoor applications since it supports data rates up to 7 Gbps [48].

(1) Usability Properties. The short-range requirement, similar to the NFC in Subsection 2.2.1, forces the user to be in close proximity of the two devices and to be vigilant of their surroundings in the covered area. Alongside with 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 set up the devices to face each other for a proper communication.

(2) 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 [50], 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.

(3) Proposed Schemes. There are not many devices that support MM-waves, e.g., [51], but their popularity is on the rise. The previously described pairing scheme PBC from the standardized WPS [41] 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 [52]). 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 force the attacker to be copresent which exposes him even by a benign user.

2.2.4. Visible Communication (VC). VC is a wireless communication technology that relies on modulating the visible spectrum using an illumination source such 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 [53]. 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 an independence of the view angle and the quality of the camera to guarantee a better detection, e.g., Pixnet [54]. However, this option assumes the existence of 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 4 as three main components: a transmitter, a channel, and a receiver.

(1) Usability Properties. Similar to the NFC in Subsection 2.2.1 and the millimeter waves in Subsection 2.2.3, 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 with 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 variety of devices. As for the LoS condition, a user intervention during the pairing is crucial in order to set up 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 photosensor in the case of a unidirectional 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.

(2) 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 [55] 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 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 photosensor. This process is fairly easy to perform in an arbitrary way which means the attacker cannot predict the outcome of

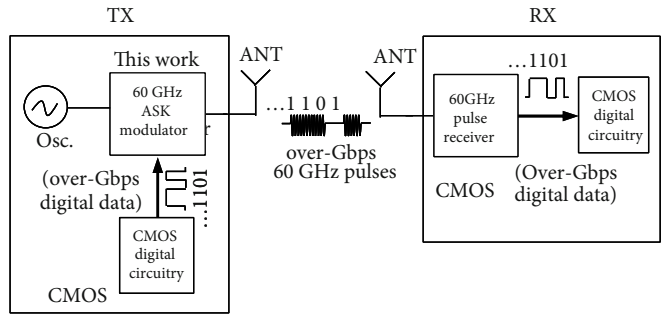

Figure 3: Block diagram of a millimeter wave communication system with a 60 GHz ASK (amplitude shift keying) modulator [49].

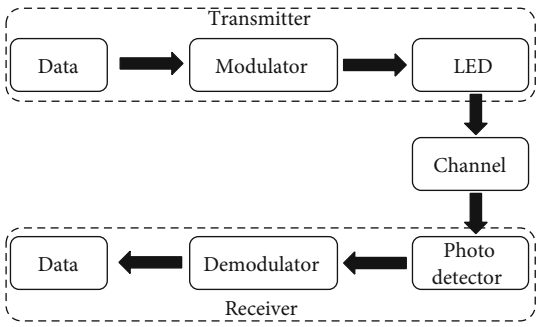

Figure 4: Block diagram of a VLC communication system [53].

the attack. Therefore, it will not be of a great impact on the pairing process and cause a MitM attack but it will lead to a denial of service (DoS). However, this technique might be useful to block the reception of the light pulse by saturating the photodetector 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 [56]. Then, he replays the video to the camera in a way to pair with it. One solution to this issue is the analysis of this property by 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 5 [57].

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.

(3) Proposed Schemes.

(1) Blinking light [58]: after exchanging the key between the devices on the in-band channel, a checksum value is sent from a LED-equipped device to a camera or a photosensor-equipped device using light pulses. The size of the checksum varies 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 [30] where the authors reimplemented the pairing scheme with a 15-bit OoB message and measured an average completion time equal to 28.8 s

(2) KeyLED [59]: two devices use LED photosensor 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

(3) Flashing displays [60]: 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

(4) Secure barcode-based visible light communication (SBVLC) [61]: 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

2.2.5. Audio. 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 6. Numerous modulation techniques have been used such as the dual-tone multifrequency (DMTF) and the on-off keying (OOK) to enhance the reliability of the channel.

(1) 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 the 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

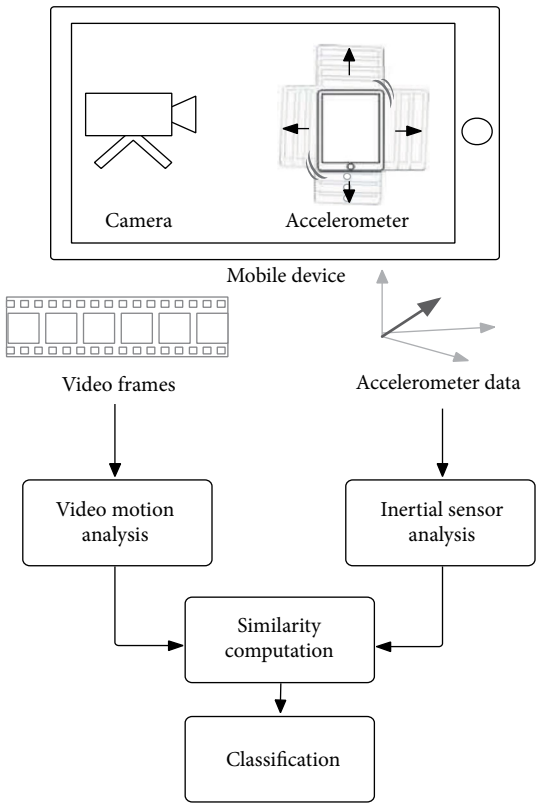

Figure 5: Classification of replay attack using video motion analysis and inertial sensor motion analysis [57].

acoustic transfer against any malicious attempt to interfere with the transmission.

(2) Security Properties. The feasibility of an eavesdropping makes the confidentiality assumption on these channels out of reach, as demonstrated in the work of Halevi and Saxena [63] using off-the-shelf equipment. Furthermore, the high applicability of a relay attack, as demonstrated in [64], 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 that is close to the legitimate devices which prevent any active malicious attempts to interfere with the authentication message transmission.

(3) Proposed Schemes.

(1) Loud and clear [65]: the scheme starts by a Diffie-Hellman key exchange over the main insecure channel and then they send the hashes of the public keys encoded in a Mad Lib sentences that are verifiable by the user. Finally, he 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

(2) HAPADEP [35]: the scheme starts by sending the encoded Diffie-Hellman public keys on the audio channel using fast codec which provides faster transmission rate but it 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 they wait for the user to confirm the match

2.2.6. 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 [68], as illustrated in Figure 7(a), 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 [36]. 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 [67, 69], as shown in Figure 7(b). This out-of-band channel is referred to as body-coupled channel (BCC) [70], 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 [23].

(1) 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 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.

(2) 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 [63]. Due to the necessity of establishing a physical contact between the devices, either by a user intervention or using mechanical waves, the feasibility of 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.

(3) Proposed Schemes.

(1) Vibrate-to-unlock [71]: 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, will be required by the tag to identify the legitimate reader

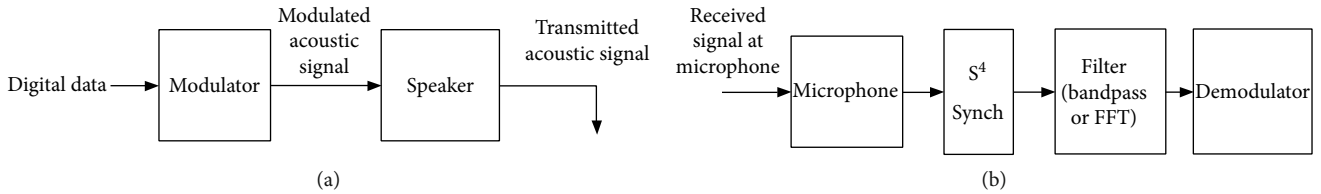

Figure 6: Block diagram of an acoustic communication system: (a) modulator/transmitter and (b) demodulator/receiver [62].

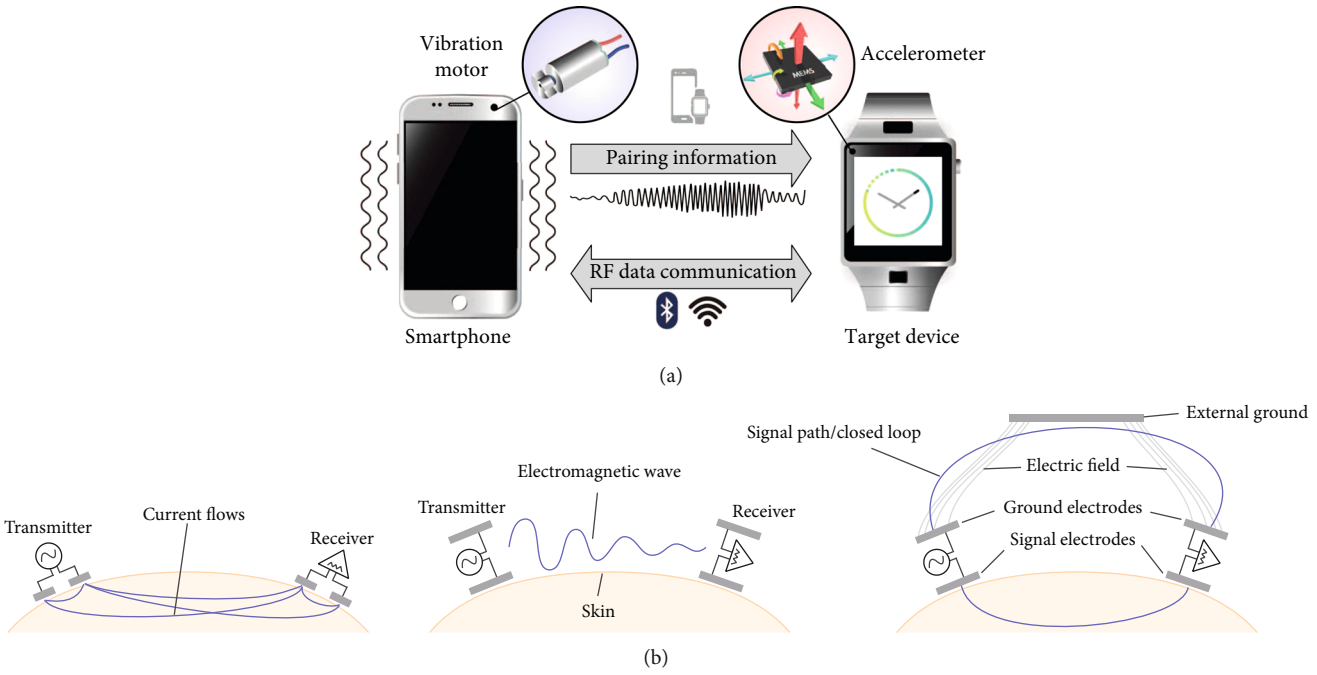

Figure 7: Examples of haptic out-of-band channels. (a) Haptic out-of-band channel based on the physical vibrations [66]. (b) Types of body channel communication: galvanic coupling, surface wave, and capacitive coupling [67].

(2) BEDA [36]: this scheme takes advantage of the user intervention to apply a physical action (button press) on the devices:

(i) 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 will take advantage of the random interevent timing, that is almost equal on each of them, to extract 21 secret bits

(ii) 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 [31] 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

(3) Body channel-based secure device pairing [67]: this protocol is based on the capacitive coupling to establish the body communication channel. It has two main phases:

(i) Key agreement: the two pairing participants establish a secret key K through the Diffie-Hellman key agreement protocol [72]

(ii) 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 8

3. Security Analysis of Out-of-Band Pairing Protocols

3.1. Threat Model Categories. In the secure device pairing context, we identify two categories of threat models based on two security properties: *the demonstrative identification* and *the device integrity*. The first property, *the demonstrative identification*, was first introduced in the work of Balfanz et al. [9] and it guarantees the correctness of the pairing initiation process by making sure that the devices performing the pairing 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

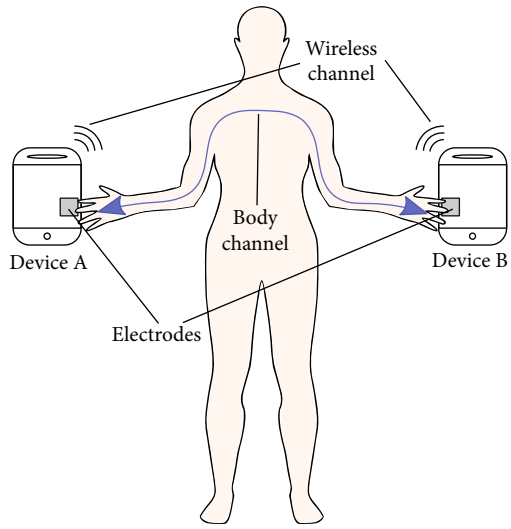

Figure 8: Body channel-based secure device pairing [67].

acquired by the attacker on the victim IoT device. Thus, it outlines the fact that one of the pairing participants is partially or completely under the control of the adversary, as detailed in the work of Do et al. [73]. This property covers both the hardware and the firmware integrity of the object. The adopted intruder models, in the formal or the computational security analysis of numerous research works on SDP, 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. [27] has demonstrated the severity of violating these security requirements by proving the feasibility of an attack that aims at pairing a malicious device instead of a legitimate one. The constraint nature of the target IoT devices is not be taken into account in the threat model since we only consider the attacks that are aiming to compromise the pairing procedure. Thus, the denial of sleep [74] or the denial of service attacks [75] are not in the score of this work. We conclude the existence of two categories of threat models:

3.1.1. Classical 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 pairing between the legitimate participants and that those devices are not under the control of the attacker. To better understand the security analysis of the SDP schemes, outlined in Subsection 3.2, we briefly describe the associated intruder models:

(i) Dolev-Yao model [6]: in this model, the adversary controls the in-band channel but he has limited capabilities on the out-of-band channel. These limitations are specified by the pairing scheme based on the choice of the channel, as described in Subsection 2.2. However, the cryptographic primitives in this model are considered as a black box and out-of-reach of the adversary. Therefore, the computational attacks are not assumed feasible. This intruder is also adopted by the strand space model [76]

(ii) AKISS model [77]: in this model, the capabilities of the adversary are similar to the Dolev-Yao intruder powers. However, the work of Delaune et al. [26] has extended the model to provide the attacker with the capability to guess a low entropy secret

(iii) Bellare-Rogaway [78, 79]: in this model, each participant is modeled as an oracle that can be addressed 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 in-band channel and his powers are limited based on the choice of the out-of-band channel, as detailed in Subsection 2.2

3.1.2. Advanced Threat Model. In comparison with the initial assumptions of the classical threat model, the demonstrative identification and the device integrity properties in the advanced threat model are not guaranteed.

The former property provides the adversary with the ability to lure the user to initiate the pairing with the wrong device which has been demonstrated feasible and easy to accomplish on the Bluetooth Secure Simple Pairing protocol [27]. Therefore, the correctness of the discovery process of the pairing between the intended devices is affected by the human factor error (HFE) and by the lack of authentication due to the absence of preshared security knowledge.

As for the latter violated property, the adversary is able to gain access to the input/output interfaces of one of the pairing participants which makes him able to intercept any message received by that device without the need of eavesdropping on the in-band or the out-of-band channel. Furthermore, he is able to send any message through that compromised devices which simply makes it an external input/output interface for the attacker. This ability can be achieved either by compromising the hardware [80–83] or the software of the object [84–86].

3.2. Security Analysis under the Classical Threat Model

3.2.1. 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 2.1.

In this part, we will presents a selection of OoB-based device pairing protocols that provide a *formal* or *a*

computational security analysis based on the adopted threat model that is described in Subsection 3.1.1. Based on the existing specifications of the chosen research works, we will describe the OoB component using the four main criteria: its nature as stated in Subsection 2.2, its security classification as detailed in our adversary model in Subsection 2.1 and the type of the required user intervention (*relay*, *compare*, *generate* or *set up*) that was first introduced in [3]. 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 dependent 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 [31] and MVSec protocols [32].

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 analyzed 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.

Table 3: 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
$h(.)$	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	Multiplication operator
$(x \times y)$ – matrix	Matrix with x rows and y columns
$\longrightarrow$	In-band channel
$- - \rightarrow$	Exchange out-of-band channel
$- - \rightarrow$	Verification out-of-band channel
$- - \rightarrow$	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

3.2.2. Evaluated Security Properties. In the literature, a number of security properties have been evaluated to investigate the correctness of the proposed pairing 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 properly lay out these results and to present a clear overview of these security assessments, we will match the outlined property with the adequate specification in the work of Lowe [24]. However, we will keep the same property formulation as detailed in the original work to provide the reader with a better understanding of the originally conduced security assessment. Based on the definitions in [24], brief descriptions of the assessed security properties are presented as follows:

(i) 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

(ii) 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

(iii) 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

(iv) 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 noninjective agreement property and, additionally, each protocol run of Alice corresponds to a unique protocol run of Bob

3.2.3. Manual Authentication II (MANA II)

(1) Protocol Steps. This protocol, proposed by Gehrmann et al. [31], is described in Figure 9 and it works as follows:

(i) ① ② The two devices, named Alice and Bob, exchange their Diffie-Hellman public keys g^a and g^b on the in-band

(ii) ③ ④ 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

(iii) ⑤ 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

(iv) ⑥ 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)

(v) ⑦ 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)

(2) Out-of-Band Specifications. The MANA II protocol uses essentially a haptic OoB channel that relies on the physic intervention of the user to compare the displayed messages ③ and ⑥. The purpose of 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 perform a delay attack on the previous in-band exchanges, as explained in Subsection 2.1, which violate the channel availability property.

(3) Security Analysis. The protocol has been formally verified in [25, 26]. The results of the validation are shown in Table 4 and the evaluated security properties are described as follows:

(1) Paper: Delaune et al. [26]

(i) Property description:

(a) 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

(ii) 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

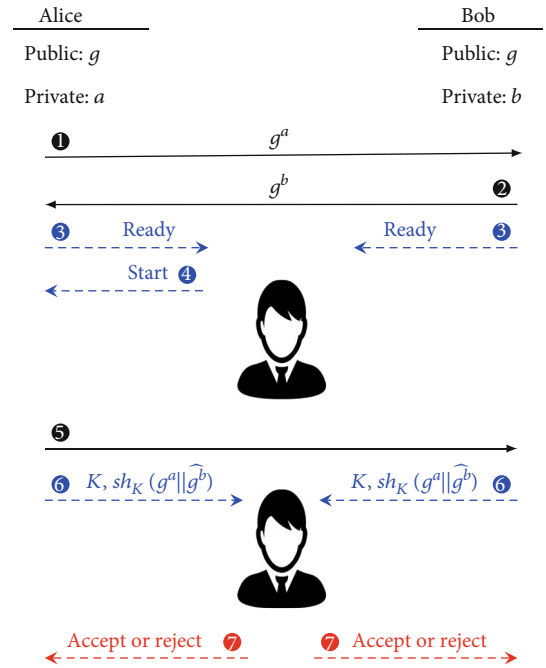

Figure 9: Alice and Bob diagram: MANA II protocol.

(2) Paper: Chang and Shmatikov [25]

(i) 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

Table 4: Summary of the security proofs.

Protocol	Security analysis	Security analysis model	Security analysis tool	Properties	Tested scenario	Results
MANA II [31]	Delaune et al. [26]	Symbolic	AKISS [77]	Noninjective 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 [25]	Symbolic (Dolev-Yao [6])	ProVerif [87]	Weak agreement	Alice to Bob	Verified
					Bob to Alice	Verified
				Injective weak agreement	Alice to Bob	Failed
					Bob to Alice	Failed
				Noninjective agreement	Alice to Bob	Failed
					Bob to Alice	Failed
				Injective agreement	Alice to Bob	Failed
					Bob to Alice	Failed
MANA III [31]	Chang and Shmatikov [25]	Symbolic (Dolev-Yao [6])	ProVerif [87]	Key confidentiality	Low entropy PIN	Failed
						Failed
					Random PIN	Verified
				Noninjective agreement	Low entropy PIN	Failed
					Random PIN	Verified
MANA IV [88] and MA-DH [88]	Laur and Nyberg [88]	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 [89]	Vaudenay [89]	Computational (Bellare-Rogaway [78, 79])	Manual	Upper bound of the successful attack probability	One-shot attack	$2^{-l} + \epsilon$
					Multisession attack	$Q_A \times Q_B \times \left(2^{-l} + \epsilon\right)$
Improved SAS-based cross-authentication [90]	Pasini and Vaudenay [90]	Computational (Bellare-Rogaway [78, 79])	Manual	Upper bound of the successful attack probability	Multisession attack	$\frac{Q(Q-1)}{2}\left(2^{-l} + \epsilon + \epsilon_u\right)$
Ephemeral pairing [91]	Hoepman [91]	Computational (Bellare-Rogaway [78, 79])	Manual	Upper bound of the successful attack probability	Multisession attack	$1 - e^{-Q/2^t} + 2^{-\lvert g^a \rvert}$
Wong-Stajano asymmetric pairing protocol [92]	Nguyen and Leneutre [93]	Symbolic (strand space model [76])	Manual	Noninjective agreement	Alice to Bob	Failed
					Bob to Alice	Failed
2-round authenticated key agreement protocol [94]	Nguyen and Leneutre [94]	Symbolic (strand space model [76])	Manual	Noninjective agreement	Alice to Bob	Verified
					Bob to Alice	Verified

(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

(ii) 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 by ProVerif [87] of session which is not the case for the AKISS tool [77]

3.2.4. Manual Authentication III (MANA III)

(1) Protocol Steps. This protocol, proposed by Gehrmann et al. [31], is described in Figure 10 and it works as follows:

(i) ①② The two devices, named Alice and Bob, exchange their Diffie-Hellman public keys g^a and g^b on the in-band

(ii) ③ The user enters a four- to six-digit random number on both devices their input interfaces (e.g., a keypad)

(iii) ④ Alice computes a long secret K_A that is used to generate an authentication string $h_{K_A}(g^a \| \widehat{g^b}, R)$. $h_K(\bullet)$ which 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

(iv) ⑤ Bob computes a long secret K_B that is used to generate an authentication string $h_{K_B}(\widehat{g^a} \| g^b, R)$. $h_K(\bullet)$ which 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, he sends it to Alice on the in-band channel

(v) ⑥⑦ Alice and Bob exchange the long keys, K_A and K_B, on the in-band channel

(vi) ⑧ Each device notifies the user of the verification outcome (e.g., using an LED signal)

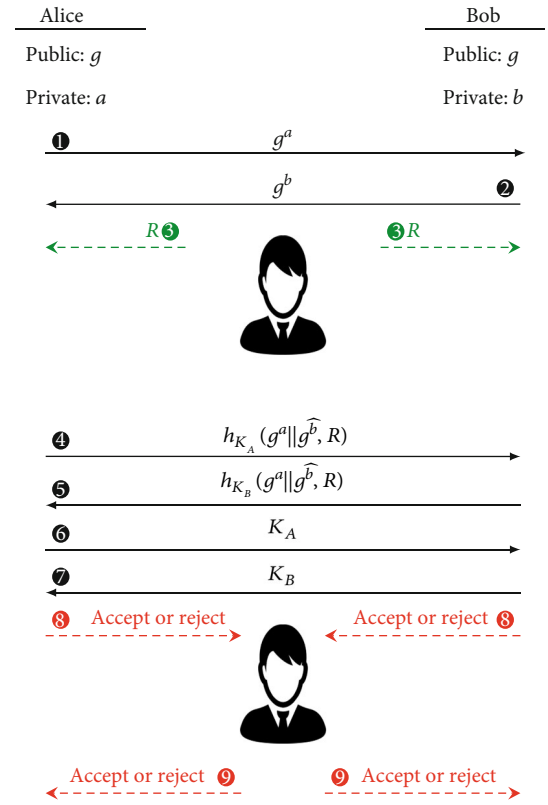

Figure 10: Alice and Bob diagram: MANA III protocol.

(vii) 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)

(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 in 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 to perform a delay attack on the previous in-band exchanges, as explained in Subsection 2.1, which violate the channel availability property for both OoB communication links.

(3) Security Analysis. The protocol has been formally verified as follows:

(i) Paper: Chang and Shmatikov [25]

(ii) 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

(iii) 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

3.2.5. Manual Authentication IV (MANA IV) and Manual Authentication Diffie-Hellman (MA-DH)

(1) Protocol Steps. This protocol MANA IV, proposed by Laur and Nyberg [88], is described in Figure 11 and it works as follows:

(i) The two devices, Alice and Bob, generate, respectively, an l-bit keys, k_A and k_B, and their DH private keys, a and b

(ii) ① 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

(iii) ② Bob sends both his DH public key g^b and the authentication key k_B to Alice

(iv) ③ Alice sends her open value d_A to Bob on the in-band channel

(v) ④ Alice computes her short authentication string (SAS) $\mathrm{SAS}_A = h_{k_A \| \widehat{k_B}}(g^a \| \widehat{g^b})$ and sends it to Bob on the out-of-band channel

(vi) ⑤ Bob verifies the correctness of the SAS sent by Alice and notifies the user to confirm the pairing

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 Figure 12 and it works as follows:

(i) The two devices, Alice and Bob, generate, respectively, a unique session identifiers, ID_A and ID_B, and their DH private keys, a and b, on the in-band channel

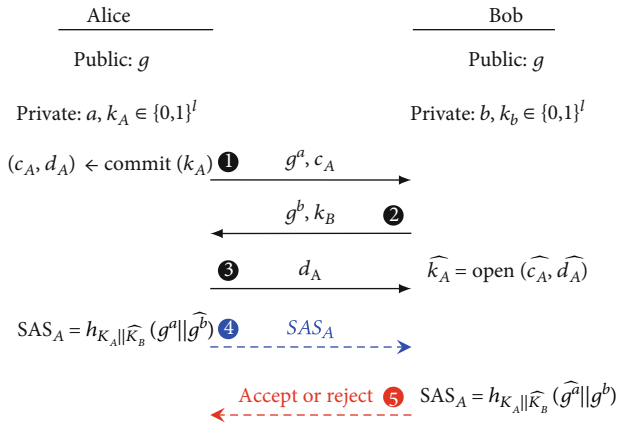

FIGURE 11: Alice and Bob diagram: MANA IV protocol.

(ii) 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

(iii) Bob sends both his DH public key g^b and his identifier to Alice on the in-band channel

(iv) Alice sends her open value d_A to Bob on the in-band channel

(v) Alice computes her short authentication string (SAS) $\mathrm{SAS}_A = h_{g^a \| \widehat{g^b}}(ID_A \| ID_B)$ and sends it to Bob on the out-of-band channel

(vi) Bob verifies the correctness of the SAS sent by Alice and notifies the user to confirm the pairing

(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 Subsection 2.1. 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 protocol allows the attacker to perform a delay attack based on the previous in-band exchanges which violate the channel availability property.

(3) Security Analysis. The two protocols have been computationally verified as follows:

(i) Paper: Laur and Nyberg [88]

(ii) Verification terminology: Appendix A

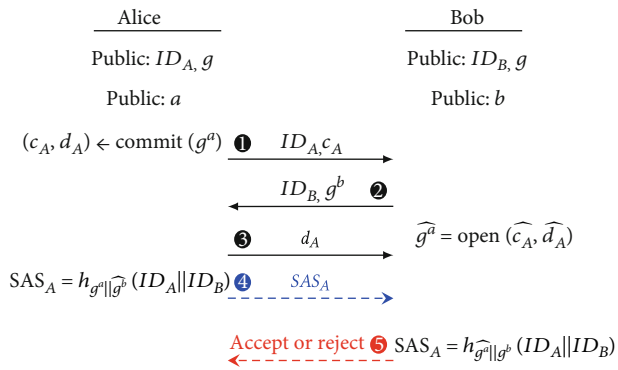

Figure 12: Alice and Bob diagram: MA-DH protocol.

(iii) Evaluated properties:

(1) Property: *upper bound of the successful attack probability*

(i) Property description: an adversary succeeds in deception if at the end of the protocol Alice and Bob reach the accepting state but $(g^a, g^{b\wedge}) \neq (g^{a\wedge}, g^a)$. As stated in [88], let A be the attacker algorithm. A protocol is considered (t, ϵ) −secure if for any t-time attacker A, the attack success probability is formulated as follows:

$$Adv^{\text{attack}}(A) = \max_{g^a, g^b} \Pr\left[\text{successful pairing}\left(g^a, g^{b\wedge}\right) \neq \left(g^{a\wedge}, g^a\right)\right] \leq \varepsilon. \quad (1)$$

(ii) Tested scenarios:

(a) Statistically binding commitment scheme: for any t, there exists $\tau = t + O(1)$ such that if the commit function Commit $(.)$ is (τ, ε_1)−hiding, ε_2-binding and (τ, ε_3)−nonmalleable and the hash function $h(.)$ is $(\varepsilon_a, \varepsilon_b)$-almost regular and ε_u-almost universal then the protocol is $(2\varepsilon_1 + \varepsilon_2 + \sqrt{\varepsilon_2} + \varepsilon_3 + \max\{\varepsilon_a, \varepsilon_b\varepsilon_u\})$−secure

(b) Computationally binding commitment scheme: for any t, there exists $\tau = 2t + O(1)$ such that if the commit function Commit $(.)$ is (τ, ε_1)−hiding, ε_2-binding and (τ, ε_3)−nonmalleable and the hash function $h(.)$ is $(\varepsilon_a, \varepsilon_b)$-almost regular and ε_u-almost universal, then the protocol is $(2\varepsilon_1 + \varepsilon_2 + \sqrt{\varepsilon_2} + \varepsilon_3 + \max\{\varepsilon_a, \varepsilon_b\varepsilon_u\})$−secure

(iv) 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 $\{\varepsilon_a, \varepsilon_b, \varepsilon_u\} = 2^{-l}$, 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 terms of security

3.2.6. SAS-Based Cross-Authentication Protocol

(4) Protocol Steps. This protocol, proposed by Vaudenay [89], is described in Figure 13 and it works as follows:

(i) The two devices, Alice and Bob, generate, respectively, nonces, R_A and R_B, and their DH private keys, a and b

(ii) ① 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

(iii) ② Bob uses a commitment scheme to commit on her DH public key g^b and her nonce R_B.Then, he sends the commit value c_B and his public key to Alice on the in-band channel

(iv) ③ Alice sends her open value d_A to Bob on the in-band channel

(v) ④ Bob sends his open value d_A to Bob on the in-band channel

(vi) ⑤ Alice retrieves the values hidden in the commitment $\widehat{c_B}$ sent by Bob using the open value $\widehat{d_B}$. He verifies b both the public key committed and the fact that the first bit is equal to one to avoid reflection attacks. Then, she computes her short authentication string (SAS) $\text{SAS}_A = R_A \oplus \widehat{R_B}$ and sends it to Bob on the out-of-band channel

(vii) ⑥ Bob verifies the correctness of the SAS sent by Alice and replies with his SAS as a confirmation of the pairing

(5) Out-of-Band Specifications. Similar to the MANA IV and MA-DH protocols 3.2.5, 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 Subsection 2.1 which can compromise the security of the scheme in a practical scenario. Finally, the structure of protocol allows the attacker to perform a delay attack based on the previous in-band exchanges which violate the channel availability property.

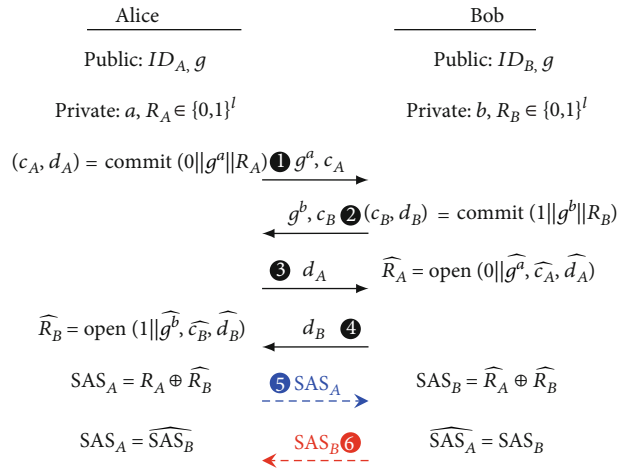

FIGURE 13: Alice and Bob diagram: SAS-based Cross-Authentication protocol.

(6) Security Analysis. The protocol has been computationally verified as follows:

(i) Paper: Vaudenay [89]

(ii) Verification terminology: Appendix A

(iii) Evaluated properties:

(1) Property: *upper bound of the successful attack probability*

(i) Property description: an attack is considered successful if there exists an instance of the protocol, between Alice and Bob, which terminates by reaching an accepting state $(ID_A \widehat{ID}_B, g^a, \widehat{g^b}) \neq (\widehat{ID}_A, ID_B, \widehat{g^a}, g^b)$

(ii) Tested scenarios:

(a) One-shot attack: assuming that the commitment scheme is either (t_c, ϵ)−extractable or (t_c, ϵ)−equivocable, there exists a small constant μ (overall time complexity of the protocol) such that for any t-time adversary, $P_{\text{one-shot}} \leq 2^{-l} + \epsilon$ or $t \geq tc - \mu$, where ϵ is negligible

(b) 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 $P_{\text{multisession}}$ can be formulated using the t-time one-shot adversary scenario to have $P_{\text{multisession}} \leq P_{\text{one-shot}} \times Q_A \times Q_B$ with a complexity $t \leq t_0 + \mu_A \times Q_A + \mu_B \times Q_B$

(iv) Assessment: the first tested scenario provides the upper bound of the one-shot attack success probability. This bound is dependent 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 $P_{\text{multisession}} \leq P_{\text{one-shot}} \times Q_A \times Q_B$. For a negligible ϵ, the probability can be $Q_A \times Q_B \times 2^{-l}$

3.2.7. Improved SAS-Based Cross-Authentication Protocol

(1) Protocol Steps. This protocol, proposed by Pasini and Vaudenay [90], is described in Figure 14 and it works as follows:

(i) The two devices, Alice and Bob, generate, respectively, a hashing key K_A and a nonce R_B. Then they generate their DH private keys, a and b

(ii) ① 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

(iii) ② Bob sends his public key g^b and his nonce R_B to Alice on the in-band channel

(iv) ③ Alice sends her open value d_A to Bob on the in-band channel

(v) ④ Alice computes her short authentication string (SAS) $\text{SAS}_A = R_A \oplus h_{K_A}(g^{b\wedge})$ and sends it to Bob on the out-of-band channel

(vi) ⑤ Bob retrieves the hashing key value from Alice's commitment. Then, he verifies the correctness of the received on the out-of-band channel and replies with his SAS as a confirmation of the pairing

(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 Subsection 2.1 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 preshared secret to construct a signature-based mechanism.

Finally, the structure of protocol allows the attacker to perform a delay attack based on the previous in-band exchanges which violate the channel availability property.

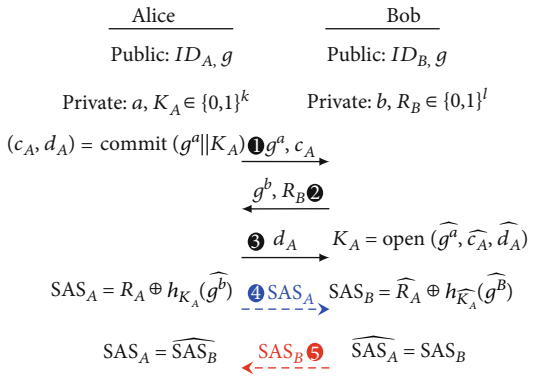

FIGURE 14: Alice and Bob diagram: improved SAS-based cross-authentication protocol.

(3) Security Analysis. The protocol has been computationally verified as follows:

(i) Paper: Pasini and Vaudenay [90]

(ii) Verification terminology: Appendix A

(iii) Evaluated properties:

(1) Property: *upper bound of the successful attack probability*

(i) Property description: an attack is considered successful if there exists an instance of the protocol, between Alice and Bob, which terminates by reaching an accepting state $(ID_A, \widehat{ID_B}, g^a, g^{b\wedge}) \neq (\widehat{ID_A}, ID_B, g^{a\wedge}, g^b)$

(ii) Tested scenario:

(a) Multisession 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 $(Q(Q-1)/2)(2^{-l}\epsilon + \epsilon_u)$

(iv) Assessment: the case of a one-shot success probability attack can be found when assuming $Q = 2$. Also, in the work of Laur and Nyberg [88], the extractability and the equivocability notions have been put into question. Furthermore, the use of the Bellare-Rogaway adversary model has been deemed complex and unsuitable for evaluating the security of authentication schemes that run statistically independent consecutive protocol executions (ad hoc device pairing protocols)

3.2.8. Ephemeral Key Exchange Protocol. (1) Protocol steps: This protocol, proposed by Hoepman [91], is described in Figure 15 and it works as follows:

(i) The two devices, Alice and Bob, generate, respectively, their DH private keys, a and b

(ii) ① 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

(iii) ② 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

(iv) ③ Alice sends a short hash of her public key $sh(g^a)$ to Bob on the out-of-band channel

(v) ④ Bob sends a short hash of his public key $sh(g^b)$ to Alice on the out-of-band channel

(vi) ⑤ Alice sends the real value of her DH public key to Bob on the in-band channel

(vii) ⑥ Bob verifies the two hashes sent in 1① and ③ using the received public key of Alice. Then, he sends the real value of his DH public key on the in-band channel

(viii) ⑦ Alice verifies the two hashes sent in ② and ④ using the received public key of Bob. Then, she sends a confirmation of the shared DH secret key $g^{b\wedge a}$ using the long hash function on the in-band channel

(ix) ⑧ Bob verifies the key confirmation of Alice and confirms the pairing by sending the hash of his DH secret key $g^{a\wedge b}$ on the in-band channel

(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 violate the security provided by the scheme when applied to a realistic use-case as detailed in Subsection 2.1. Also, the channel availability property is not guaranteed based on the structure of the protocol

(3) Security analysis: the protocol has been computationally verified as follows:

(i) Paper: Hoepman [91]

(ii) Verification terminology: Appendix A

(iii) Evaluated properties:

(1) Property: *upper bound of the successful attack probability*

(i) Property description: an attack is considered successful if there exists an instance of the protocol, between Alice and Bob, which

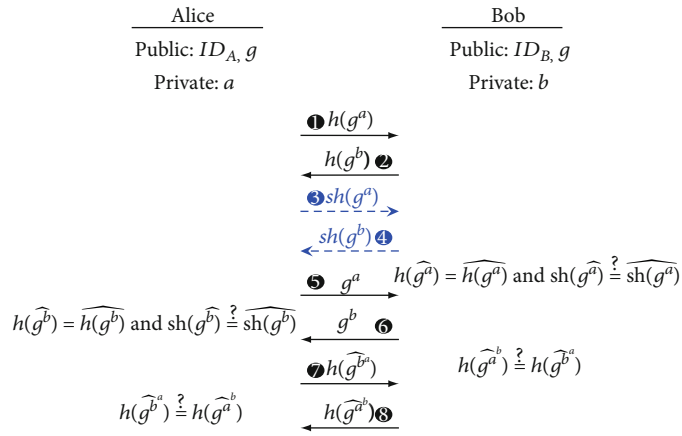

Figure 15: Alice and Bob diagram: ephemeral key exchange protocol based on a bidirectional out-of-band channel.

terminates by reaching an accepting state $(g^a, g^{b\wedge}) \neq (g^{a\wedge}, g^b)$

(ii) Tested scenario:

(a) 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 $1 - e^{Q/2^l} + 2 - |g^a|$

(iv) 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 [88] 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

3.2.9. Wong-Stajano Asymmetric Pairing Protocol

(1) Protocol Steps. This protocol, proposed by Wong and Stajano [92], is described in Figure 16 and it works as follows:

(i) The two devices generate, respectively, their DH private keys, a and b. Then, Bob generates a short nonce R_B and long hashing key K_B

(ii) ① Alice sends her identifier ID_A and her DH public key $h(g^a)$ to Bob on the in-band channel

(iii) ② Bob computes the keyed hash $h_{K_B}(ID_B, R_B, g^b, g^{a\wedge})$. Then, he sends it along with his identifier and his DH public key g^b to Alice on the in-band channel

(iv) ③ Alice replies by an acknowledgement Ack on the out-of-band channel to confirm the reception of the message ②

(v) ④ Bob sends the short nonce R_B to Alice on the out-of-band channel

(vi) ⑤ Bob sends the value of the hashing key K_B to Alice on the in-band channel

(vii) ⑥ Alice verifies the hash sent in using the hashing key and the public key of Bob. Then, she confirms or rejects the pairing on the out-of-band channel

(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 Subsection 2.2.6, 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 Subsection 2.2.4, the vigilant user is required to set up the devices in a way to create a direct line of sight (LoS). Finally, the protocol structure allows the attacker to delay messages on the out-of-band channel by applying this action on the previous

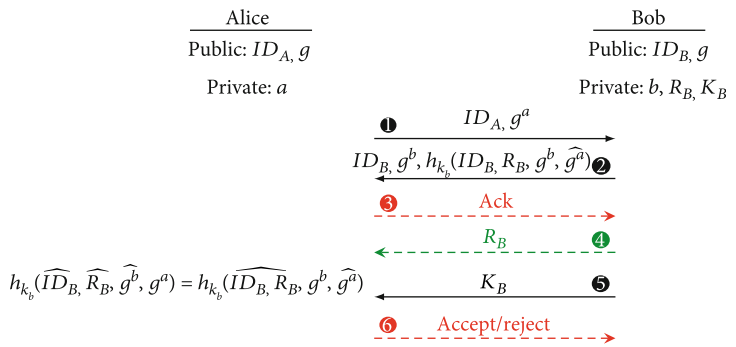

Figure 16: Alice and Bob diagram: asymmetric pairing protocol based on a unidirectional out-of-band channel.

in-band exchanges which violate the channel availability property. Therefore, the channels used in this scheme are considered *delayable*.

(3) Security Analysis. The protocol has been formally verified as follows:

(i) Paper: Nguyen and Leneutre [93]

(ii) Evaluated properties:

(a) Property: non-injective agreement [24]

(b) Property description: the initiator Alice completes a run of the protocol, apparently with Bob, and 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

(iii) 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 [5] by eliminating the acknowledgement message which reduces the user intervention. Furthermore, they improved the protocol design by removing the use of two successive unidirectional messages that eliminate the vulnerability noticed by Nguyen and Leneutre [93] 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

3.2.10. 2-Round Authenticated Key Agreement Protocol

(1) Protocol Steps. This protocol, proposed by Nguyen and Leneutre [94], is described in Figure 17 and it works as follows:

(i) The two devices, Alice and Bob, generate, respectively, their DH private keys, a and b, and their nonces, r_a and r_b

(ii) ① Alice sends her DH public key g^a and the hash value $h(g^a, r_a)$ to Bob on the in-band channel

(iii) ② Bob sends his DH public key g^b and his nonce r_b to Alice on the in-band channel

(iv) ③ 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

(v) ④ Bob retrieves the value of r_a from the message ③, verifies the hash sent in message ①, and confirms or rejects the pairing on the out-of-band channel

(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 in Subsection 2.1, 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 violate the channel availability property. Therefore, the channels used in this scheme are considered *delayable*

(3) Security Analysis. The protocol has been formally verified as follows:

(i) Paper: Nguyen and Leneutre [94]

(ii) Evaluated properties:

(1) Property: non-injective agreement [24]

(a) Property description: the initiator Alice completes a run of the protocol, apparently with

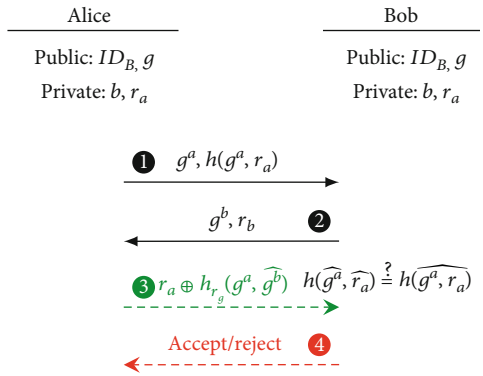

Figure 17: Alice and Bob diagram: 2-round authenticated key agreement protocol with unidirectional out-of-band channel.

Bob, and 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

(iii)Assessment: based on the manual formal analysis conducted by the authors, the scheme achieves the non-injective agreement property while minimizing 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-constrained devices

3.2.11. Summary. In this subsection, we summarize the highlighted results shown in Table 4. The MANA II protocol [31] has been formally verified in [25, 26] using two automated verification tools: ProVerif [87] and AKISS [77]. The work of Delaune et al. [26] focused on evaluating the non-injective agreement property, described in Subsection 3.2.2, 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 behavior or a one-digit mismatch might compromise the security of the pairing process as detailed in the work of Fomichev et al. [3]. However, a similar formulation of this property has been verified in the work of Chang and Shmatikov [25] 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 previous session. In addition, three other similar formulations of the properties, described in Subsection 3.2.2, have been evaluated: weak agreement, injective weak agreement, and injective agreement. On the first, one holds since it provides the weakest definition authentication by guaranteeing the agreement on the identities of the two intended devices that are 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 [31] 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 a memorable four to six-digit values. On the other hand, the Wong-Stajano asymmetric pairing protocol [92] does not guarantee the non-injective agreement that has been formally evaluated, in the work of Nguyen and Leneutre [93], based on the strand space model [76]. 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 have been corrected in the design proposed in the work of Nguyen and Roscoe [5]. A lightweight pairing scheme has been introduced in another work of Nguyen and Leneutre [94] 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-force attack that is aimed 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. The two variants of the MANA suite protocols, MANA IV and MA-DH [88], 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 protocols are asymptotically optimal in terms 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 [89, 90], has been evaluated under the Bellare-Rogaway model [78, 79]. Nonetheless, in the work of Laur and Nyberg [88], the extractability and the equivocability notions, described in Appendix A, 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 [91], has two outcomes. It describes the advantage of an active adversary searching for a collision between the two hashes to a bypass the verification. The second part describes the advantage of a passive attacker that tries to guess an $|g|$-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 $Q \times 2^{-l}$ which has been improved in the work of Laur and Nyberg [88] where they reduced the number of OoB exchanges by using a single unidirectional channel.

3.3. Security Analysis under the Advanced Threat Model

3.3.1. Identity Misbinding Attack. The identity misbinding attack, also known as *unknown key-share* attack, was first

identified on the station-to-station (STS) protocol [95] in the work of Blake-Wilson and Menezes [96] in 1999. To simplify the attack's applicability on secure device pairing schemes, brought to light in the work of Sethi et al. [27], 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 preshared information between the devices. Second, 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.

In Figure 18, 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 by her own keyed hash to confirm to Bob that she has the same key which was not 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 device performing the pairing is 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. Things explain the attack assumption of having at least a compromised device. At this moment, the SDP assumption of having two unidentified devices without any preshared 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.

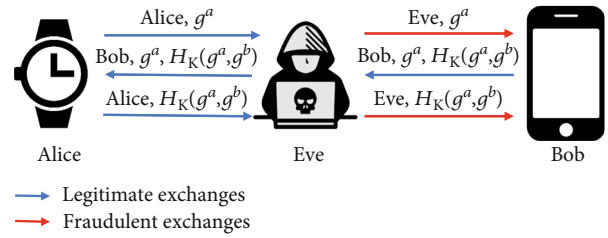

Figure 18: Misbinding attack against a Diffie-Hellman key exchange.

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 maximize 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.

3.3.2. Case Study: Bluetooth Secure Simple Pairing (SSP) Protocol. This attack has been demonstrated on the *numerical comparison* variant of the Bluetooth Secure Simple Pairing (SSP) protocol [42], as shown in Figure 19.

The attack on the SSP protocol can occur as follows:

(1) The user makes Bob discoverable and starts discovering the neighboring objects enabling Bluetooth

(2) Eve copies the Bluetooth identifier of Bob and then makes it nondiscoverable

(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

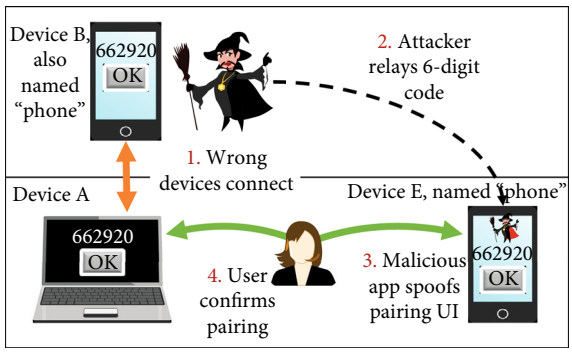

Figure 19: Misbinding attack against Bluetooth SSP numeric comparison [97].

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.

3.3.3. 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 [27] demonstrated this attack on the bootstrapping scheme Extensible Authentication Protocol-Nimble Out-of-Band (EAP-NOOB) [98] that pairs and registers the IoT devices to an online server. This scheme is an authentication method for the Extensible Authentication Protocol [99] 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 preshared 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:

(i) In-band key exchange: the IoT object perform an ECDH key exchange with the server

(ii) Object selection: the user selects the IoT object from a list, provided by the server, on his personal device

(iii) 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

(iv) 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, is aimed 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.

3.3.4. Mitigation. The misbinding attack can be mitigated by cryptographically binding the device identifiers to the protocol session. Unfortunately, this solution is not possible for the secure device pairing schemes since the objects do not share any prior information, including preshared symmetric keys or certificates. Another potential solution is the use of copresence verification techniques that are based on variables from the ambient environment. However, numerous samples of these methods have been proven vulnerable against active attacks in the work of Shrestha et al. [64] 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.

4. Secure Pairing Design Recommendations and 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 (SSP) protocol represents one of the most widely used security pairing schemes with its three variants: *PIN entry* inspired from the MANA III protocol, described in Subsection 3.2.4, *numerical comparison* inspired from the MANA II protocol, described in Subsection 3.2.3, and the *out-of-band* which uses the NFC technology 2.2.1. 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, [100, 101], pointed out numerous vulnerabilities related to the human factor error resulted from the previously described user action, 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. [32], 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 [102] 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 property 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 4 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 [24]. Furthermore, the automated formal and computation verification tools should consider the derived categories of the out-of-band channels, highlighted in Section 2, in order to better model their offered security guarantees and to enhance their applicability to the ad hoc secure device pairing protocols. Also, we have noticed that the automated computational analysis using tools such as CryptoVerif [103] does not support the use of out-of-band channels which eliminate 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 vulnerabilities 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, permit 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 neglect by future work in the secure device pairing field, is the consideration of the advanced threat model, described in Subsection 3.1.2, 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 protocol, we noticed the interest in using context-based schemes, also referred to as zero-interaction protocols [2]. However, the security analysis of these techniques is only 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. [104] by disclosing a brute-force attack against the interlock protocol applied in the MagPairing protocol [15] 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.

5. Conclusion

In this survey, we have addressed the secure device pairing problem from the security perspective by providing a refined adversary model on the out-of-band channel that is suitable to the ad hoc pairing context. This threat model eliminates the replay capability of the attacker and it introduces a new notion of delay that is based on the protocol structure rather than the out-of-band channel characteristics. Based on these refinements, we proposed a new out-of-band 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. Furthermore, we surveyed the formal and the computational security analysis 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 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 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. Our work motivates the use of a formal or a computational security analysis to validate the correctness of the SDP scheme that will be proposed in the future. Our description framework can be extended to all the SDP proposals in the literature in order to create an official secure device pairing repository that clearly describes the security aspects and the discovered attacks on a specific pairing scheme. Finally, we think that the modeling of the out-of-band channels by the security verification tools should be extended in order to better abstract all the security properties guaranteed by these channels that are considered the only source of security in the secure pairing context.

Appendix

A. Cryptographic Primitives

In this part, we introduce the properties of the cryptographic primitives used in these security proofs [88–91].

A.1. Keyed Hash Function. The keyed hash function $h : M \times K \longrightarrow T$ 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:

(i) ϵ_u -almost universal: for any two inputs $x_0, x_1 \in M$ such that $x_0 \neq x_1$, the probability $\Pr[k \longleftarrow K : h(x_0, k) \oplus h(x_1, k)] \leq \epsilon_u$

(ii) ϵ_u -almost XOR universal: for any $x_0, x_1 \in M$ and $y \in T$ such that $x_0 \neq x_1$, the probability $\Pr[k \longleftarrow K : h(x_0, k) \oplus h(x_1, k) = y] \leq \epsilon u$, where $\epsilon_u \geq 1/|T|$

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

(i) (ϵ_a, ϵ_b)–almost regular with respect to the subkeys: for each input $x \in M$, $y \in T$ and subkeys $\widehat{K_a} \in K_a$, $\widehat{K_b} \in K_b$, the probabilities $\Pr[k_a \longleftarrow K_a : h(x, k_a, \widehat{k_b})] \leq \epsilon_a$ and $\Pr[k_a \longleftarrow K_b : h(x, \widehat{k_a}, k_b)] \leq \epsilon_u$, where $\epsilon_a, \epsilon_b \geq 1/|T|$

(ii) ϵ_u–almost universal with respect to the subkey 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 $\Pr[k_a \longleftarrow K_a : h(x_0, k_a, k_b) = h(x_1, k_a, \widehat{k_b})] \leq \epsilon_u$, where $\epsilon_u \geq 1/|T|$

(iii) Strongly ϵ_u–almost universal with respect to the subkey 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 $\Pr[k_a \longleftarrow K_a : h(x_0, k_a, k_b) = h(x_1, k_a, \widehat{k_b})] \leq \epsilon_u$, where $\epsilon_u \geq 1/|T|$

(iv) 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 functions h_1, h_2 are assumed independent if they satisfy $\Pr[h_2(x) = a | h_1(x) = b] = \Pr[h_2(x) = a] = 2^{-l}$

A.2. Commitment Scheme. The commitment scheme is constructed using three algorithms:

(i) The generation function Gen: generates the public parameters pk used by the commitment function

(ii) The commitment function $\mathrm{Com}_{pk} : M \times R \longleftarrow C \times D$: transforms the input $m \in M$ and a random value $r \in R$ into a commitment string $c \in C$ and an open value $d \in D$

(iii) The decommitment function $\mathrm{Open}_{pk} : C \times D \longleftarrow M$: reveals the value of the commitment string $m = \mathrm{Open}_{pk}(c, d)$ for all $(c, d) = \mathrm{Com}_{pk}(m, r)$. 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, ϵ_1)–hiding if any t time adversary achieves the following attack success probability:

$$2.\left|\Pr\left[pk \longleftarrow \mathrm{Gen}, s \longleftarrow \{0,1\}, \{x_1, x_0\} \longleftarrow A(pk), (c_s, d_s) \longleftarrow \mathrm{Com}_{pk}(x_s) : A(c_s) = s\right] - \frac{1}{2}\right| \leq \epsilon_1. \quad (A.1)$$

The commitment scheme is (t, ϵ_2)−binding if any t time adversary achieves the following attack success probability:

$$\Pr\left[pk \longleftarrow \text{Gen}, (c, d_0, d_1) \longleftarrow A(pk): \text{Open}_{pk}(c, d_0) \neq \perp \text{and Open}_{pk}(c, d_1) \neq \perp\right] \leq \epsilon_2. \quad (A.2)$$

In addition, a commitment scheme is nonmalleable; 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 [89, 90], there are two extra commitment properties introduced as follows:

(i) 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 $c = \text{Com}_{pk}(m, r)$ when there exists a decommitment d such that $(r, m) = \text{Open}_{pk}(c, d)$

(ii) Equivocability: there are two deterministic algorithms simcommit (m) and equivocate(m, c, r, ϕ). 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 $(m, r) = \text{Open}_{pk}(c, d)$ from the information (c, ϕ) provided by simcommit

Furthermore, they use, in [89, 90], the notion of a random oracle commitment scheme where the function Co $\text{m}_{pk}(m, r)$ 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 $\text{Open}_{pk}(m, c, d)$ simply verifies the hash $H(d, m) = c$ and uses d to retrieves r when the condition holds.

Data Availability

No data were used to support this study.

Additional Points

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the submission of this paper.

Acknowledgments

This work was supported by the SEIDO LAB (the joint research laboratory for Security and Internet of Things between EDF R&D and Télécom Paris). This research was also funded in part by the National Association for Research and Technology under grant No. 2018/1810.

References

[1] M. Conti and C. Lal, "Context-based co-presence detection techniques: a survey," *Computers & Security*, vol. 88, p. 101652, 2019.

[2] M. Fomichev, M. Maass, L. Almon, A. Molina, and M. Hollick, "Perils of zero-interaction security in the Internet of things," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 1, pp. 1–38, 2019.

[3] M. Fomichev, F. Álvarez, D. Steinmetzer, P. Gardner-Stephen, and M. Hollick, "Survey and systematization of secure device pairing," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 517–550, 2017.

[4] S. Mirzadeh, H. Cruickshank, and R. Tafazolli, "Secure device pairing: a survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 17–40, 2014.

[5] L. H. Nguyen and A. W. Roscoe, "Authentication protocols based on low-bandwidth unspoofable channels: a comparative survey," *Journal of Computer Security*, vol. 19, no. 1, pp. 139–201, 2011.

[6] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[7] W. R. Claycomb and D. Shin, "Extending formal analysis of mobile device authentication," *Journal of Internet Services and Information Security*, vol. 1, no. 1, pp. 86–102, 2011.

[8] M. Burrows, M. Abadi, and R. M. Needham, "A logic of authentication," *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, vol. 426, no. 1871, pp. 233–271, 1989.

[9] D. Balfanz, D. K. Smetters, P. Stewart, and H. C. Wong, "Talking to strangers: authentication in ad-hoc wireless networks," in *Proceedings of Network and Distributed System Security Symposium 2002 (NDSS'02)*, San Diego, CA, USA, February 2002.

[10] S. Mathur, R. Miller, A. Varshavsky, W. Trappe, and N. Mandayam, "Proximate: proximity-based secure pairing using ambient wireless signals," in *Proceedings of the 9th*

international conference on Mobile systems, applications, and services - MobiSys '11, pp. 211–224, Bethesda, Maryland, USA, 2011.

[11] A. Scannell, A. Varshavsky, A. LaMarca, and E. D. Lara, "Proximity-based authentication of mobile devices," *International Journal of Security and Networks*, vol. 4, no. 1/2, pp. 4–16, 2009.

[12] W. Xi, C. Qian, J. Han et al., "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, Vienna, Austria, October 2016.

[13] N. Karapanos, C. Marforio, C. Soriente, and S. Capkun, "Sound-proof: usable two-factor authentication based on ambient sound," in *24th {USENIX} Security Symposium ({USENIX} Security 15)*, pp. 483–498, Washington D.C, USA, 2015.

[14] D. Schürmann and S. Sigg, "Secure communication based on ambient audio," *IEEE Transactions on Mobile Computing*, vol. 12, no. 2, pp. 358–370, 2013.

[15] R. Jin, L. Shi, K. Zeng, A. Pande, and P. Mohapatra, "MagPairing: pairing smartphones in close proximity using magnetometers," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1306–1320, 2016.

[16] M. Miettinen, N. Asokan, T. D. Nguyen, A.-R. Sadeghi, and M. Sobhani, "Context-based zero-interaction pairing and key evolution for advanced personal devices," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security - CCS '14*, Scottsdale, Arizona, USA, 2014.

[17] B. Shrestha, N. Saxena, H. T. T. Truong, and N. Asokan, "Drone to the rescue: relay-resilient authentication using ambient multi-sensing," in *Financial Cryptography and Data Security*, Springer, 2014.

[18] H. T. T. Truong, X. Gao, B. Shrestha, N. Saxena, N. Asokan, and P. Nurmi, "Comparing and fusing different sensor modalities for relay attack resistance in zero-interaction authentication," in *2014 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, Budapest, Hungary, March 2014.

[19] C. Hu, X. Cheng, F. Zhang, D. Wu, X. Liao, and D. Chen, "OPFKA: secure and efficient ordered-physiological-feature-based key agreement for wireless body area networks," in *2013 Proceedings IEEE INFOCOM*, pp. 2274–2282, Turin, Italy, April 2013.

[20] Q. Jiang, Z. Chen, J. Ma, X. Ma, J. Shen, and D. Wu, "Optimized fuzzy commitment based key agreement protocol for wireless body area network," *IEEE Transactions on Emerging Topics in Computing*, p. 1, 2019.

[21] D. Oberoi, W. Y. Sou, Y. Y. Lui, R. Fisher, L. Dinca, and G. P. Hancke, "Wearable security: key derivation for body area sensor networks based on host movement," in *2016 IEEE 25th International Symposium on Industrial Electronics (ISIE)*, pp. 1116–1121, Santa Clara, CA, USA, June 2016.

[22] D. Schürmann, A. Brüsch, S. Sigg, and L. Wolf, "Bandana—-body area network device-to-device authentication using natural gait," in *2017 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 190–196, Kona, HI, USA, 2017.

[23] A. Ali and F. A. Khan, "Key agreement schemes in wireless body area networks: taxonomy and state-of-the-art," *Journal of Medical Systems*, vol. 39, no. 10, p. 115, 2015.

[24] G. Lowe, "A hierarchy of authentication specifications," in *Proceedings 10th Computer Security Foundations Workshop*, pp. 31–43, Rockport, MA, USA, 1997.

[25] R. Chang and V. Shmatikov, "Formal analysis of authentication in Bluetooth device pairing," *FCS-ARSPA07*, p. 45, 2007.

[26] S. Delaune, S. Kremer, and L. Robin, "Formal verification of protocols based on short authenticated strings," in *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pp. 130–143, Santa Barbara, CA, USA, August 2017.

[27] M. Sethi, A. Peltonen, and T. Aura, "Misbinding attacks on secure device pairing and bootstrapping," in *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, pp. 453–464, Auckland, New Zealand, July 2019.

[28] L. F. Cranor, "A framework for reasoning about the human in the loop," in *UPSEC'08: Proceedings of the 1st Conference on Usability, Psychology, and Security*, pp. 1:1–1:15, Berkeley, CA, USA, April 2008.

[29] R. Kainda, I. Flechais, and A. W. Roscoe, "Usability and security of out-of-band channels in secure device pairing protocols," in *Proceedings of the 5th Symposium on Usable Privacy and Security - SOUPS '09*, p. 11, Mountain View, California, USA, 2009.

[30] A. Kumar, N. Saxena, G. Tsudik, and E. Uzun, "A comparative study of secure device pairing methods," *Pervasive and Mobile Computing*, vol. 5, no. 6, pp. 734–749, 2009.

[31] C. Gehrmann, C. J. Mitchell, and K. Nyberg, "Manual authentication for wireless devices," *RSA Cryptobytes*, vol. 7, no. 1, pp. 29–37, 2004.

[32] J. Han, Y.-H. Lin, A. Perrig, and F. Bai, "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 - WiSec '14*, pp. 51–56, Oxford, United Kingdom, 2014.

[33] J. Sen, "Security in wireless sensor networks," *Wireless Sensor Networks: Current Status and Future Trends*, vol. 407, p. 408, 2012.

[34] B. Alpern and F. B. Schneider, "Recognizing safety and liveness," *Distributed Computing*, vol. 2, no. 3, pp. 117–126, 1987.

[35] C. Soriente, G. Tsudik, and E. Uzun, "HAPADEP: human-assisted pure audio device pairing," in *Information Security*, pp. 385–400, Springer, 2008.

[36] C. Soriente and E. Uzun, "*BEDA: button-enabled device association*".

[37] "How NFC works," nearfieldcommunication.org, 2015, http://nearfieldcommunication.org/how-it-works.html.

[38] L. Francis, G. Hancke, K. Mayes, and K. Markantonakis, "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.

[39] R. Zhou and G. Xing, "nShield: a noninvasive NFC security system for mobiledevices," in *Proceedings of the 12th annual international conference on Mobile systems, applications, and services - MobiSys '14*, pp. 95–108, 2014.

[40] S. Akter, T. Chakraborty, T. A. Khan, S. Chellappan, and I. Al, "Can you get into the middle of near field communication?," in *2017 IEEE 42nd Conference on Local Computer Networks (LCN)*, pp. 365–373, Singapore, Singapore, October 2017.

[41] W.-F. Alliance, *Wi-Fi Simple Configuration Technical Specification, version 2.0. 5*, Wi-Fi Alliance, 2014.

[42] S. Bluetooth, *Bluetooth Core Specification v5. 0*, Bluetooth Special Interest Group, Kirkland, WA, USA, 2016.

[43] S. A. Weis, "RFID (radio frequency identification): principles and applications," *System*, vol. 2, no. 3, pp. 1–23, 2007.

[44] N. Instrument, *Advanced RFID Measurements: Basic Theory to Protocol Conformance Test*, 2013.

[45] C. Castelluccia and G. Avoine, "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.

[46] K.-C. Huang and Z. Wang, *Millimeter Wave Communication Systems*, vol. 29, John Wiley & Sons, 2011.

[47] IEEE Standards Association 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.

[48] W.-F. Alliance, *WiGig® and the Future of Seamless Connectivity*, Wi-Fi Alliance, 2013.

[49] A. Oncu and M. Fujishim, "Millimeter-wave CMOS impulse radio," in *Advances in Solid State Circuit Technologies*, pp. 255–288, 2010.

[50] D. Steinmetzer, J. Chen, J. Classen, E. Knightly, and M. Hollick, "Eavesdropping with periscopes: experimental security analysis of highly directional millimeter waves," in *2015 IEEE Conference on Communications and Network Security (CNS)*, pp. 335–343, Florence, Italy, September 2015.

[51] *SiBEAM captures world's first 60 GHz millimeter-wave smartphone design win in leTV's flagship smartphone, Le Max*, 2015.

[52] *HP Elite x2 1011 G2 - connecting to the wireless dock*, 2019.

[53] A. R. Ndjiongue, H. C. Ferreira, and T. M. Ngatched, *Visible Light Communications (VLC) Technology*, Wiley Encyclopedia of Electrical and Electronics Engineering, 1999.

[54] S. D. Perli, N. Ahmed, and D. Katabi, "Pixnet: interference-free wireless links using LCD-camera pairs," in *Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking - MobiCom '10*, pp. 137–148, Chicago, Illinois, USA, 2010.

[55] J. Classen, J. Chen, D. Steinmetzer, M. Hollick, and E. Knightly, "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, Paris, France, 2015.

[56] A. Dziech, J. Bialas, A. Glowacz et al., "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, Regensburg, Germany, September 2013.

[57] M. Rahman, U. Topkara, and B. Carbunar, "Seeing is not believing: visual verifications through liveness analysis using mobile devices," in *Proceedings of the 29th Annual Computer Security Applications Conference on - ACSAC '13*, pp. 239–248, New Orleans, Louisiana, USA, 2013.

[58] N. Saxena, J.-E. Ekberg, K. Kostiainen, and N. Asokan, "Secure device pairing based on a visual channel," in *2006 IEEE Symposium on Security and Privacy (S&P'06)*, p. 6, Berkeley/Oakland, CA, USA, 2006.

[59] R. Roman and J. Lopez, "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, Atlanta, GA, USA, 2008.

[60] T. Kovačević, T. Perković, and M. Čagalj, "Flashing displays: user-friendly solution for bootstrapping secure associations between multiple constrained wireless devices," *Security and Communication Networks*, vol. 9, no. 10, 2016.

[61] B. Zhang, K. Ren, G. Xing, X. Fu, and C. Wang, "SBVLC: secure barcode-based visible light communication for smartphones," *IEEE Transactions on Mobile Computing*, vol. 15, no. 2, pp. 432–446, 2016.

[62] R. Jurdak, A. G. Ruzzelli, G. M. P. O'Hare, and C. V. Lopes, "Mote-based underwater sensor networks: opportunities, challenges, and guidelines," *Telecommunication Systems*, vol. 37, no. 1-3, pp. 37–47, 2008.

[63] T. Halevi and N. Saxena, "Acoustic eavesdropping attacks on constrained wireless device pairing," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 3, pp. 563–577, 2013.

[64] B. Shrestha, N. Saxena, H. T. T. Truong, and N. Asokan, "Sensor-based proximity detection in the face of active adversaries," *IEEE Transactions on Mobile Computing*, vol. 18, no. 2, pp. 444–457, 2019.

[65] M. T. Goodrich, M. Sirivianos, J. Solis, G. Tsudik, and E. Uzun, "Loud and clear: human-verifiable authentication based on audio," in *26th IEEE International Conference on Distributed Computing Systems (ICDCS'06)*, p. 10, Lisboa, Portugal, 2006.

[66] K. Lee, V. Raghunathan, A. Raghunathan, and Y. Kim, "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, Orlando, FL, USA, October 2018.

[67] M. Roeschlin, I. Martinovic, and K. B. Rasmussen, "Device pairing at the touch of an electrode," *NDSS*, vol. 18, pp. 18–21, 2018.

[68] N. Roy, M. Gowda, and R. R. Choudhury, "Ripple: communicating through physical vibration," in *12th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 15)*, pp. 265–278, Oakland, CA, USA, 2015.

[69] M. Li, S. Yu, W. Lou, and K. Ren, "Group device pairing based secure sensor association and key management for body area networks," in *2010 Proceedings IEEE INFOCOM*, pp. 1–9, San Diego, CA, USA, March 2010.

[70] A. Ruaro, J. Thaysen, and K. B. Jakobseny, "Head-centric body-channel propagation paths characterization," in *2015 9th European Conference on Antennas and Propagation (EuCAP)*, pp. 1–4, Berlin, Heidelberg, Germany, 2015.

[71] N. Saxena, M. B. Uddin, J. Voris, and N. Asokan, "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, Seattle, WA, USA, March 2011.

[72] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 2006.

[73] Q. Do, B. Martini, and K.-K. R. Choo, "The role of the adversary model in applied security research," *Computers & Security*, vol. 81, pp. 156–181, 2019.

[74] A. Gallais, T.-H. Hedli, V. Loscri, and N. Mitton, "Denial-of-sleep attacks against IoT networks," in *2019 6th International*

Conference on Control, Decision and Information Technologies (CoDIT), pp. 1025–1030, Paris, France, April 2019.

[75] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[76] F. J. T. Fábrega, J. C. Herzog, and J. D. Guttman, "Strand spaces: why is a security protocol correct?," in *Proceedings. 1998 IEEE Symposium on Security and Privacy (Cat. No.98CB36186)*, pp. 160–171, Oakland, CA, USA, 1998.

[77] R. Chadha, V. Cheval, Ş. Ciobâcă, and S. Kremer, "Automated verification of equivalence properties of cryptographic protocols," *ACM Transactions on Computational Logic (TOCL)*, vol. 17, no. 4, pp. 1–32, 2016.

[78] M. Bellare and P. Rogaway, "Entity authentication and key distribution," in *Advances in Cryptology — CRYPTO' 93. CRYPTO 1993. Lecture Notes in Computer Science, vol 773*, D. R. Stinson, Ed., pp. 232–249, Springer, Berlin, Heidelberg, 1994.

[79] M. Bellare and P. Rogaway, "Provably secure session key distribution: the three party case," in *STOC '95: Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing*, pp. 57–66, Las Vegas, NV, USA, May 1995.

[80] J. Clark, S. Leblanc, and S. Knight, "Compromise through USB-based hardware Trojan horse device," *Future Generation Computer Systems*, vol. 27, no. 5, pp. 555–563, 2011.

[81] J. Dofe, J. Frey, and Q. Yu, "Hardware security assurance in emerging IoT applications," in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2050–2053, Montreal, QC, Canada, May 2016.

[82] S. Moein, F. Gebali, and I. Traore, "Analysis of covert hardware attacks," *Journal of Convergence*, vol. 5, 2014.

[83] N. Patwari, J. Croft, S. Jana, and S. K. Kasera, "High-rate uncorrelated bit extraction for shared secret key generation from channel measurements," *IEEE Transactions on Mobile Computing*, vol. 9, no. 1, pp. 17–30, 2010.

[84] H. A. Abdul-Ghani, D. Konstantas, and M. Mahyoub, "A comprehensive IoT attacks survey based on a building-blocked reference model," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 3, 2018.

[85] M. M. Ahemd, M. A. Shah, and A. Wahid, "IoT security: a layered approach for attacks defenses," in *2017 International Conference on Communication Technologies (ComTech)*, pp. 104–110, Rawalpindi, Pakistan, April 2017.

[86] J. Deogirikar and A. Vidhate, *Security Attacks in IoT: A Survey*32–37.

[87] B. Blanchet, B. Smyth, V. Cheval, and M. Sylvestre, *ProVerif 2.00: Automatic Cryptographic Protocol Verifier, User Manual and Tutorial*, pp. 05–16, 2018.

[88] S. Laur and K. Nyberg, "Efficient mutual data authentication using manually authenticated strings," in *Cryptology and Network Security*, pp. 90–107, Springer, 2006.

[89] S. Vaudenay, "Secure communications over insecure channels based on short authenticated strings," in *Advances in Cryptology – CRYPTO 2005*, pp. 309–326, Springer, 2005.

[90] S. Pasini and S. Vaudenay, "SAS-based authenticated key agreement," in *Public Key Cryptography - PKC 2006*, pp. 395–409, Springer, 2006.

[91] J.-H. Hoepman, "The ephemeral pairing problem," in *Financial Cryptography*, pp. 212–226, Springer, 2004.

[92] F. L. Wong and F. Stajano, "Multichannel security protocols," *IEEE Pervasive Computing*, vol. 6, no. 4, pp. 31–39, 2007.

[93] T. Nguyen and J. Leneutre, "Formal analysis of secure device pairing protocols," in *2014 IEEE 13th International Symposium on Network Computing and Applications*, pp. 291–295, Cambridge, MA, USA, August 2014.

[94] T. Nguyen and J. Leneutre, "A secure and effective device pairing protocol," in *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, pp. 507–512, Las Vegas, NV, USA, 2015.

[95] W. Diffie, P. C. Van Oorschot, and M. J. Wiener, "Authentication and authenticated key exchanges," *Designs, Codes and Cryptography*, vol. 2, no. 2, pp. 107–125, 1992.

[96] S. Blake-Wilson and A. Menezes, "Unknown key-share attacks on the station-to-station (STS) protocol," in *Public Key Cryptography. PKC 1999. Lecture Notes in Computer Science, vol 1560*pp. 154–170, Springer, Berlin, Heidelberg.

[97] A. Peltonen, M. Sethi, and T. Aura, "Formal verification of misbinding attacks on secure device pairing and bootstrapping," *Journal of Information Security and Applications*, vol. 51, article 102461, 2020.

[98] T. Aura and M. Sethi, *Nimble Out-of-Band Authentication for EAP (EAP-NOOB). draft-aura-eap-noob-03 (Work in Progress)*, 2018.

[99] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz, "et al," *Extensible authentication protocol (EAP)*, 2004.

[100] C. Kuo, J. Walker, and A. Perrig, "Low-cost manufacturing, usability, and security: an analysis of Bluetooth simple pairing and Wi-Fi protected setup," in *Financial Cryptography and Data Security*, pp. 325–340, Springer, 2007.

[101] D.-Z. Sun, Y. Mu, and W. Susilo, "Man-in-the-middle attacks on secure simple pairing in Bluetooth standard v5. 0 and its countermeasure," *Personal and Ubiquitous Computing*, vol. 22, no. 1, pp. 55–67, 2018.

[102] H. Tanaka, "Information leakage via electromagnetic emanations and evaluation of tempest countermeasures," in *Information Systems Security*, pp. 167–179, Springer, 2007.

[103] B. Blanchet, "CryptoVerif: computationally sound mechanized prover for cryptographic protocols," *Dagstuhl Seminar Formal Protocol Verification Applied*, vol. 117, p. 156, 2007.

[104] Y. Wu, B. Chen, Z. Zhao, and Y. Cheng, "Attack and countermeasure on interlock-based device pairing schemes," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 3, pp. 745–757, 2018.