HELP: Helper-Enabled In-Band Device Pairing Resistant Against Signal Cancellation

Nirnimesh Ghose, Loukas Lazos, and Ming Li
{*nghose,llazos,lim*}*@email.arizona.edu*

Department of Electrical and Computer Engineering,
University of Arizona, Tucson

Abstract

Bootstrapping trust between wireless devices without entering or preloading secrets is a fundamental security problem in many applications, including home networking, mobile device tethering, and the Internet-of-Things. This is because many new wireless devices lack the necessary interfaces (keyboard, screen, etc.) to manually enter passwords, or are often preloaded with default keys that are easily leaked. Alternatively, two devices can establish a common secret by executing key agreement protocols. However, the latter are vulnerable to Man-in-the-Middle (MitM) attacks. In the wireless domain, MitM attacks can be launched by manipulating the over-the-air transmissions. The strongest form of manipulation is signal cancellation, which completely annihilates the signal at a targeted receiver. Recently, cancellation attacks were shown to be practical under predictable channel conditions, without an effective defense mechanism.

In this paper, we propose HELP, a helper-assisted message integrity verification primitive that detects message manipulation and signal cancellation over the wireless channel (rather than prevent it). By leveraging transmissions from a helper device which has already established trust with one of the devices (e.g., the hub), we enable signal tampering detection with high probability. We then use HELP to build a device pairing protocol, which securely introduces new devices to the network without requiring them to share any secret keys with the existing devices beforehand. We carry out extensive analysis and real-world experiments to validate the security and performance of our proposed protocol.

1 Introduction

In recent years, we have experienced a proliferation of advanced personal wireless devices (APDs) such as smartwatches, wearable sensors, RFID devices, home monitoring sensors for Internet-of-Things applications, etc. [38]. These devices often connect to a gateway/hub (e.g., a Wi-Fi access point) for data collection or for remote actuation. Securing the communication between APDs and the hub is of paramount importance when the former collect sensitive data, or can control critical functions within their environment. The process of establishing trust between the APD and the hub is known as *secure bootstrapping* and is achieved via a two-party mutual authentication and key-agreement mechanism.

The prevailing methods for secure device bootstrapping are either by manually loading the hub's secret to the device or to preload the APDs with some unique secret. The preloaded secret of APDs can be made known to the hub using an out-of-band (OOB) channel, e.g., the user enters the secret manually. However, many APDs such as smart bulbs, motion sensors, smart key locks, etc., lack advanced interfaces for entering or changing passwords. Moreover, it is a common occurrence that manufacturers opt to preload devices with default keys that are easily leaked. In fact, the largest DDoS attack launched to date exploited default passwords preloaded to APDs–IP cameras, digital video recorders, smart TVs–to recruit hundreds of thousands of nodes into the Mirai botnet and attack the DNS infrastructure [57].

On the other hand, a public key infrastructure (PKI) is also impractical for wide deployments. This is because a PKI typically requires a connection to a centralized certification authority. For devices deployed on-the-fly in areas with intermittent Internet connectivity, reachback to central certificate repositories may not be a robust option. Moreover, PKIs face significant scalability, heterogeneity, and interoperability challenges. As an average person or household owns an increasing number of devices, the device association process must happen within a short time and require very little or no human effort. Also, a trust initialization protocol must be lightweight, as APDs typically have low processing capability and are energy constrained.

Several device pairing protocols have been proposed for device pairing without pre-shared secrets [1,8,11,18, 26, 29, 32, 37, 40–42, 44, 54]. Most such protocols require an auxiliary secure out-of-band (OOB) channel, an audio or visual channel for example, that is observable by a user to aid the authentication of messages transmitted over the public wireless channel. However, such OOB channels introduce practical interoperability issues due to the heterogeneity of the devices and are not user-friendly. Recently, in-band pairing protocols [10,17,23] have been proposed as an alternative to OOB pairing. The former protocols only require that devices possess a common wireless interface to communicate. Since the wireless channel is known to be insecure in general, the security of these protocols relies on the assumption that wireless signal cancellation is infeasible, so that message integrity and authentication properties can be derived by encoding the messages in a special way. However, as demonstrated by Popper *et al.* [47], this assumption may not hold in many cases. Thus, it remains an open problem as to whether secure in-band device pairing protocols can still be designed under a strong Dolev-Yao attacker which can annihilate wireless signals.

In this paper, for the first time, we seek an answer to the above question. Instead of trying to *prevent* signal cancellation attacks, we propose an approach to *detect* the presence of an attacker who attempts to nullify the signal at a receiver. Our core idea for verifying the integrity of a message m is to superimpose another signal from a *helper* device (e.g., a smartphone) while m is being transmitted. Any cancellation attack on m is bound to also cancel the superimposed signal from the helper. The helper is assumed to have an existing trust association with one of the devices in the network (e.g., the hub), and is co-present with the primary device that is authenticated by the hub. The superimposed signal is later revealed by the helper via the authenticated channel, to allow for the recovery of m. Our protocol achieves a strong "tamper-evidence" property where there are no restrictions on what kind of signal manipulation the attacker is allowed to do.

Specifically, the device's message m is encoded with ON-OFF keying and Manchester-coding. During the transmission of m, the helper synchronously injects some random signal at randomly selected slots. Any signal nullification attempt will cancel both the legitimate transmitter's and the helper's signal, presuming that the activity periods for the helper are not easily discernible. The helper later reveals its activity periods via an authenticated channel to enable the hub in the detection of signal nullification attempts. Trust between the hub and the helper is established using traditional means (e.g., input a shared random password on the smartphone when it is first paired with the hub), which is a one-time cost. With only one helper in a network, we can securely introduce many new devices at no extra hardware cost, thus ensuring scalability and usability. Essentially, by exploiting the co-presence of the helper with the new device(s), our protocol transfers the trust from the helper to the new device(s).

The main contributions of this paper are four-fold:

- We construct a novel physical layer message integrity verification primitive to detect signal cancellation attacks over the wireless channel. We show that our primitive achieves message integrity protection with only in-band communications.

- We utilize the proposed message integrity verification primitive to construct a secure in-band device pairing protocol named HELP based on the Diffie-Hellman (DH) key agreement [14]. Whereas the primitive provides one-way integrity verification (device-to-hub), we show that HELP achieves two-way authenticated key agreement (counter-intuitively). This is done via a novel way that exploits the helper's superposed random signals to simultaneously protect both the integrity and confidentiality of the DH public parameters, such that an adversary impersonating the hub cannot successfully establish a key with a legitimate device.

- We theoretically analyze the security of the proposed integrity verification primitive and the HELP protocol, and we establish bounds for the adversary's success probability under active attacks (especially Man-in-the-Middle attacks). We show that the adversary's success probability is a negligible function of the protocol parameters and thus can be driven to an arbitrary small value.

- We carry out extensive experiments to evaluate the effectiveness of the signal cancellation detection mechanism and the pairing protocol. Our experiments verify that device co-presence significantly hardens the adversary's ability to distinguish between the helper's and the legitimate device's transmissions. We also implement the proposed protocol in our Universal Software Radio Peripheral (USRP) testbed and evaluate the adversary's successful pairing probability with and without the protection of our integrity verification primitive. The experimental results are in line with our analytical findings.

The paper is organized as follows: we discuss related work in Section 2. We state the system and threat models in Section 3. We present the integrity verification primitive and the HELP pairing protocol in Section 4. The security of the pairing primitive and of HELP are analyzed in Section 5. In Section 6, we study the adversary's capability in inferring the helper's transmissions and injecting modified messages by performing experiments on the USRP platform. We further experimentally evaluate the HELP assisted key-agreement protocol. We conclude the paper in Section 7.

2 Related Work

In this section, we review previous works in trust establishment without prior associations, which involves both message authentication and key-agreement. It is well known that key agreement can be achieved using traditional cryptographic protocols such as a DH key exchange [14]; however, public message exchange over the wireless medium is vulnerable to Man-in-the-Middle (MitM) attacks, which are notoriously difficult to thwart without any prior security associations. To thwart MitM attacks, additional message authentication and integrity protection mechanisms are required. Therefore, next we mainly review works in authentication/integrity protection without pre-shared secrets.

2.1 Out-of-Band Channel based Approaches

Many existing secure device pairing methods rely on some out-of-band (OOB) channel to defend against MitM attacks [1, 8, 11, 18, 26, 29, 32, 37, 40–42, 44, 54]. The OOB channel is assumed to possess certain security properties (e.g., it is only accessible by the user), which helps verify the integrity of messages transmitted over the wireless channel. However, OOB channels usually require non-trivial human support and advanced user interfaces. For example, when a visual channel is used, a user needs to read a string from one device's screen and input it into another [1, 11, 37], or visually compare multiple strings or LED flashing patterns [31, 32, 44]. Other works require specialized hardware such as a Faraday cage to isolate the legitimate communication channel [27, 30]. On the other hand, biometric signals [3,12,21,46,53,61,62,64] have been proposed to create a secure channel through which nodes on the same body can derive a shared secret. However, their applications are restricted to wearable devices, require uniform sensing hardware, and are susceptible to remote biometrics sensing attacks [20]. In addition, others have proposed to exploit the shared physical context for authentication and key agreement. Examples of common modalities include the accelerometer measurements when two devices are shaken together [35, 36], or light and sound for two devices located in the same room [38,52]. Again, these require additional hardware and are not interoperable, whereas in many cases the contextual source has low entropy.

2.2 Non-cryptographic Device Authentication

As an alternative, non-cryptographic authentication techniques usually derive trust from *hard-to-forge* physical-layer characteristics unique to each device/link. They usually transmit information "in-band" without requiring an OOB channel. Existing approaches on non-cryptographic device authentication [9, 25, 33, 45, 60, 65] can be classified into three categories: (a) *device proximity*, (b) *location distinction*, and (c) *device identification*. In device proximity methods, the common idea is to exploit the channel reciprocity and its rapid decorrelation (within a few wavelengths) with distance. However, such techniques typically require advanced hardware which is not suitable for constrained wireless devices. For example, [9, 45, 65] require multiple-antennas, and [33] needs a wide-band receiver. Moreover, these techniques only address the common key extraction problem, leaving them vulnerable to MitM attacks. Distance bounding techniques [5, 49, 50] were also proposed to ensure proximity, but they are not so practical yet (either resort to OOB channels or specially designed hardware). Location distinction methods such as temporal link signatures that detect location differences [25, 43, 60] require high bandwidth (> 40MHz), which is not always available to low-cost, resource-constrained devices. Finally, device identification techniques [6,13,16] distinguish devices based on their unique physical-layer or hardware features. Unfortunately, both location distinction and device identification techniques require prior training or frequent retraining, which is not applicable to APDs first introduced to an environment.

2.3 In-Band Approaches for Message Integrity Protection

Whereas the above approaches authenticate a device's presence, they do not necessarily protect the integrity of the messages transmitted by a device, due to the possibility of signal manipulation attacks over the wireless channel [10]. There have been few past attempts to design in-band message integrity protection mechanisms, which assume that signal cancellation over the wireless channel is not possible [10, 23], or occurs with bounded success [22]. For example, Tamper-Evident Pairing (TEP) proposed by Gollakota *et al.* in 2011 [17], and integrity codes (I-codes) proposed by Čapkun *et al.* in 2008 [10] both assumed the infeasibility of signal cancellation. Based on message integrity, message authentication can be achieved by assuming the presence of the legitimate device is known (a.k.a. authentication through presence). However, the infeasibility of signal cancellation assumption does not always hold. Pöpper *et al.*

demonstrated an effective relay signal cancellation attack using a pair of directional antennas, which works regardless of the packet content and modulation [47]. Recently, Hou *et. al.* [22] showed that it is possible to prevent signal cancellation only if the channel itself has enough randomness. A typical indoor environment may not be sufficient because the devices are static and the channel is usually stable.

To remedy the significant shortcomings of existing device pairing schemes, we (for the first time) introduce the core idea of detecting signal manipulation attacks even if signal cancellation is 100% effective. This is achieved through the introduction of a *helper* device which is already securely paired with the hub in an offline fashion (e.g., using conventional pairing methods). With the aid of the helper, trust can be established securely for newly introduced devices without significant human effort or any advanced hardware. Our protocol only uses in-band wireless communication, and thus, it is interoperable.

3 Problem Statement

3.1 System Model

We consider a star network topology, where a wireless base station (*BS*) services multiple personal devices, which is similar to an Internet-of-things (IoTs) scenario. For example, the network can reside inside a home or an office space. Our goal is to securely pair an unauthenticated device with the base station in the presence of an adversary and establish a common key between the device and the *BS*. The adversary can either try to hijack the uplink communication to pair with the *BS*, or spoof a rogue *BS* to pair with a legitimate device. The device and the *BS* do not pre-share any common secrets (*e.g.* secret cryptographic keys). We assume that a user initiates the pairing process by powering the device and setting it to pairing mode. Figure 1 describes the system model. Formally, the following entities are part of the system model.

Base Station (*BS*): The *BS* serves all the legitimate devices and needs to establish a secure communication link with each of them. The *BS* connects with the legitimate devices through a wireless channel. The *BS* verifies and pairs with any legitimate device requesting to join the network.

Helper Device (*H*): The helper is an auxiliary device such as a smartphone, that assists the *BS* in the pairing process. The helper has already established a secure authenticated channel with the *BS*, either by establishing a common key, using a public/private key pair, or through some OOB channel [1, 37]. Using this secure channel, H can apply an authenticated encryption function AE$(\cdot)$ on a message m_H to guarantee the confidentiality and integrity of m_H, and the authenticity of the source. Any such AE$(\cdot)$ can be utilized with the proposed protocol. For example, if H and the *BS* share a public/private key pair, H can encrypt/sign/encrypt (or sign/encrypt/sign) its message to guarantee the necessary security properties. If H and *BS* share a common master symmetric key, an encrypt-then-MAC operation can be followed to implement AE$(\cdot)$, after separate symmetric keys are generated from the master key for the encryption and MAC operations. One of the examples is to use encryption then message authentication code hashing with the shared key. We refer the reader to [2] for more details on authenticated encryption. We leave the exact specification of AE$(\cdot)$ open to allow for both symmetric and/or asymmetric methods.

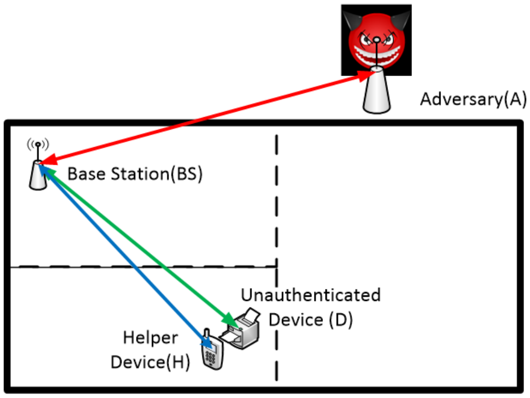


Figure 1: Entities of the system model and basic setup.

Note that pairing H to the *BS* is a one-time effort and need not be repeated with every device join. Moreover, only the helper is required to have an advanced interface to pair with the *BS*.

Legitimate Device (*D*): A legitimate device is a typical APD which does not share any secrets with the *BS* or H. The device is usually small and has simple user interfaces (such as a power button) and hardware capabilities. The legitimate device, H, and the *BS* are assumed to be copresent during the pairing process. H and D are placed in close proximity such that they have a highly correlated wireless channel.

3.2 Threat Model

Adversary: We consider the typical *Dolev-Yao model* [15]. The adversary (A), can fully control the wireless channels of the network. For example, it can eavesdrop, modify, remove, replay or inject messages (frames) transmitted on the wireless channel. The adversary is also powerful enough to annihilate signals transmitted from D and H over the wireless channel, such that they do not reach the BS (and vice versa). This can be accomplished by techniques proposed by Pöpper *et al.* [47]. The pairing protocol itself is known to A, but the adversary does not have physical access to any of the devices. The helper device is assumed to be trusted and its secret key with the BS is kept away from adversaries.

Note that we do not impose any location restriction for the attacker. Although the devices are typically located in a physically bounded area such as a home, we do not assume that this is a secure region. Instead, the attacker can be located inside the physical space, as long as the attacker cannot physically control the device and the BS to be paired. That is, the attacker does not control the helper so that it cannot initiate the pairing with the BS when no legitimate device is present. The user is aware of the presence of both the BS and of the legitimate device (which are powered on) when the pairing is initiated. This is the minimal assumption adopted by the majority of the previous works in device pairing.

The goal of an attacker is to pair successfully with the BS and/or D. Therefore, we mainly consider a MitM attacker in our security analysis. However, in this paper, we do not focus on preventing denial-of-service (DoS) attacks such as jamming, which is orthogonal to our studies. Similarly with all relevant literature, we assume that the adversary is incapable of physically blocking signals (e.g., by adding a Faraday cage) to the device, the helper, or the base station.

In addition, at any point in time, the attacker may try to find out who is transmitting on the wireless channel. There could be several cases: device only, helper only, BS only, or device plus helper together. For example, the attacker can do so via energy detection or use physical layer identification/fingerprinting techniques [7,19,28,39,55,59]. Since we assume that D and H have a highly correlated channel due to their proximity, it is generally difficult for the attacker to differentiate between the cases of device only and helper only. Thus, the attacker can differentiate between the number of transmitters (i.e., $D+H$ or D/H alone), but the attacker cannot perfectly distinguish D and H (i.e., the probability of successful detection is less than 100%). We propose specific power and slot synchronization randomization methods to ensure that D and H are not easily distinguishable. Note that any device distinction method has to operate only to correspond to the online nature of a MitM attack.

4 HELP: Helper-Enabled Pairing

In this section, we present HELP, an in-band Helper-enabled pairing protocol that does not require secret preloading. HELP makes use of a new PHY-layer message integrity protection primitive to detect signal cancellation attacks that are launched to perform a MitM attack against a key agreement protocol. We first describe the PHY-layer protection primitive and then use this primitive to construct HELP.

4.1 Message Integrity Protection Against Signal Cancellation

Consider the simple scenario depicted in Figure 1. A new legitimate device D wants to pair with the BS by transmitting a message m_D over a wireless channel. Message m_D is not protected by any cryptographic message integrity mechanism such as a MAC because D and the BS do not share any prior security association. Let $\mathbf{x}_D$ denote the corresponding signal transmitted from D carrying m_D. Let also an adversary A perform a signal cancellation attack on the received signal $\mathbf{y}_D = \mathbf{h}_{D,BS}\mathbf{x}_D$ at the BS, where $\mathbf{h}_{D,BS}$ denotes the channel between D and the BS. Simultaneously, A injects his own signal $\mathbf{x}_A$ carrying message m_A. The main challenge in providing message integrity is to detect that a cancellation/injection has taken place.

To combat signal cancellations, we employ Manchester-coded (MC) ON-OFF keying modulation to transmit m_D from D to the BS similar to [10,17]. In ON-OFF keying, a zero bit is mapped to (OFF, ON) slots pair, whereas a one bit is mapped to (ON, OFF) slots pair. The receiver demodulates the ON-OFF keying sequence by applying energy detection on every slot. The advantage of ON-OFF keying is that it hardens signal cancellations, as the adversarial device, A has to "erase" the received signal $\mathbf{y}_D$ at the BS by synchronizing its own signal transmission $\mathbf{x}_A$ and taking into account the channels $\mathbf{h}_{D,BS}$ and $\mathbf{h}_{A,BS}$. Different from previous approaches [10, 17, 24], we consider the worst case scenario where signal cancellation is possible due to the stability and predictability of the respective channels, as it was demonstrated in [47].

The MC facilitates several functions. First, the alteration between ON and OFF slots prevents the zero wandering problem, allowing the receiver to keep a power reference for differentiating between ON and OFF slots, irrespective of the data sequence. More importantly, an MC message contains an equal number of zeros and ones. Our integrity protection mechanism relies on the detection of canceled ON slots and therefore, the guarantee of ON slots irrespective of the data sequence is critical to the protocol security. Finally, the use of MC allows for the recovery of the device's message when the latter has been corrupted from the intentional transmissions of the helper. Revealing the "time locations" of the helper's ON slots enables the message recovery.

In the proposed integrity primitive, the helper is placed in close proximity to the unauthenticated device D and synchronously transmits a message m_H while m_D is being transmitted. A signal cancellation targeted at the *BS* is bound to also cancel the signal from H. With the completion of the m_D transmission, the helper reveals m_H to the *BS*, who verifies if any part of m_H has been canceled.

If the message integrity verification test is passed, the *BS* exploits the knowledge of m_H to recover m_D. A key requirement for the successful detection of signal cancellations is that the adversary A cannot *swiftly* identify the ON slots of the helper. We achieve this requirement by placing the helper in close proximity to D and by randomizing the transmit power and the starting time of each ON-OFF slot at D and H. Placing H close to D makes it difficult to differentiate the two devices using transmission directionality or the uniqueness of the wireless channel. Note that the ON-OFF transmissions contain no preambles, so channel estimation becomes difficult. The randomization of the power and ON slot firing times aim at preventing the device distinction using RSS measurements or the possible time misalignment between the two devices due to inaccurate synchronization or different paths to the adversary. We emphasize that any device distinction mechanism must operate online—the adversary has to decide to cancel an ON slot within the first few samples—which renders existing sophisticated radio fingerprinting techniques inadequate [7,19,28,39,55,59]. We now describe the PHY-layer message integrity verification primitive in detail.

4.2 HELP Integrity Verification

We propose a message integrity verification method called HELP that operates with the assistance of a helper device H. The integrity of a message m_D transmitted from D to the *BS* is verified via the following steps.

1. **Device Placement:** The helper H is placed in close proximity to the unauthenticated device D.

2. **Initialization:** The user presses a button on D or simply switches D on to set it to pairing mode. The user then presses a button on H to initiate the protocol. The helper sends an authenticated *request-to-communicate* message to the *BS* using the AE$(\cdot)$ function. This message attests that the legitimate device D is present and H is placed near D.

3. **Device Synchronization:** The *BS* sends a publicly known synchronization frame SYNC to synchronize the clocks of D, H and itself[1]. The SYNC frame is similar in function to the known preamble that is attached to wireless transmissions for synchronizing the receiver to the transmitter. In our protocol, all three entities synchronize to the same time reference, using the known SYNC message.

4. **Transmission of m_D:** D transmits m_D in the form $[h(m_D)], m_D$, where $[\cdot]$ denotes an MC ON-OFF keyed message and h is a cryptographically-secure hash function. Note that no key input is used with h, as D and the *BS* do not share a common key.

5. **Helper Signal Superposition:** Synchronously with the transmission of $[h(m_D)]$, the helper transmits a signal m_H with ON slots in a random number of slot locations determined by vector $\mathbf{s}$. The ON slots in $\mathbf{s}$ are time-aligned with the slots (ON or OFF) of $[h(m_D)]$. Only one slot of m_H can be ON per MC ON-OFF bit of $[h(m_D)]$. Sequence m_H is not necessarily a proper MC sequence (and hence, is not marked by $[\cdot]$).

6. **Reception at the *BS*:** The *BS* receives $([h(m_D)] + m_H)'$ and m'_D.

7. **Revealing m_H:** The helper reveals AE$(\mathbf{s}, K)$ to the *BS*.

8. **Integrity Verification of s:** The *BS* decrypts $\mathbf{s}$ and verifies its integrity using function VD$(\cdot)$, which is the corresponding decryption/verification function to AE$(\cdot)$. If verification fails, the *BS* aborts m'_D.

9. **Integrity Verification of m_D:** The *BS* verifies that all slot locations indicated by $\mathbf{s}$ are ON on the received $([h(m_D)] + m_H)'$. If not, a signal cancellation attack is detected and m'_D is rejected. Otherwise, the *BS* recovers $h(m_D)'$, by removing m_H from $([h(m_D)] + m_H)'$ using the knowledge of $\mathbf{s}$. For bits where $\mathbf{s}$ was OFF in both corresponding slots, the MC sequence is decoded using typical decoding. For an ON slot in $\mathbf{s}$, a bit b_D is decoded using the truth table in Figure 2(a). Upon recovery of $h(m_D)'$, the *BS* checks if $h(m'_D) \overset{?}{=} h(m_D)'$. If the integrity verification fails at the *BS*, either the *BS* or H display a FAILURE message, and all entities abort the protocol. The user has to restart the pairing process from the initialization step. If the integrity verification passes, then *BS* or H display a SUCCESS message.

The steps for extracting $[h(m_D)']$ from $([h(m_D)] + m_H)'$ at the *BS* are shown in Figure 2(b). After synchronization, D transmits $h(m_D) = 0110110101$ in the form of $[h(m_D)]$ (for illustration purposes, we have restricted the length of the hash function to 10 bits). The helper synchronously transmits during slots $\mathbf{s} =$

[1] The SYNC message doesn't need to be secured since if it is canceled at both device and helper, it becomes a DoS attack. If the device and helper are forced to be out of sync by an attacker, BS will fail to decode which is again a DoS.

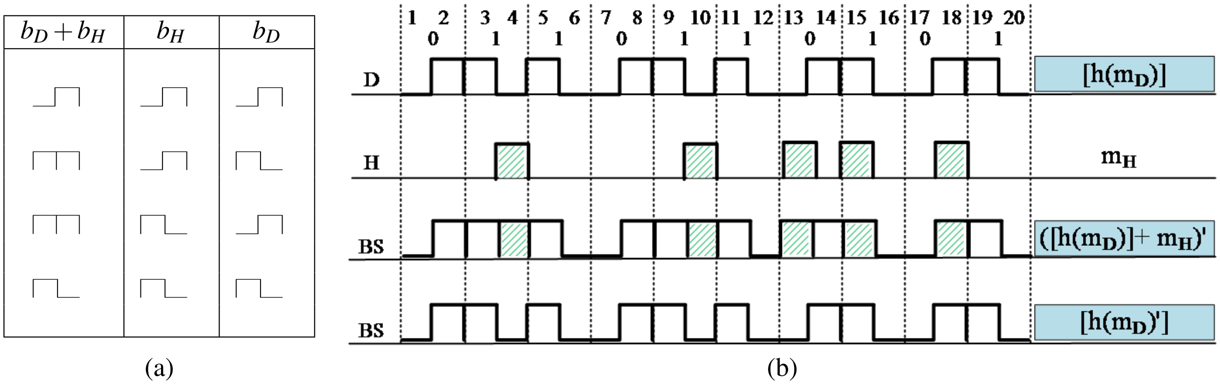


Figure 2: (a) Truth table for recovering $[h(m_D)']$ from $([h(m_D)] + m_H)'$, using $\mathbf{s}$, and (b) an example of recovering $[h(m_D)']$ from $([h(m_D)] + m_H)'$.

$\{4,10,13,15,18\}$. The BS receives the superimposed signal $([h(m_D)] + m_H)'$. Using the truth table in Figure 2(a), the original MC sequence corresponding to $h(m_D)$ is recovered.

4.3 Device Pairing with HELP

In this section, we describe how the BS and D can establish a secret key in the presence of a MitM adversary. We complement the DH key agreement protocol with the HELP integrity verification primitive. The latter is used to detect the cancellation portion of a MitM attack. Moreover, the helper provides the necessary authentication for the DH message exchange. The HELP-enabled DH message exchange is shown in Figure 3.

To fix the ideas, the BS (or D) publishes parameters $(\mathbb{G}, q, g)$ of the DH scheme, where ($\mathbb{G}$ is a cyclic group of order q and g is a generator of G). If $(\mathbb{G}, q, g)$ are already publicly known, they need not be sent by either party. Device D computes $z_D = g^{X_D}$, where X_D is chosen from $\mathbb{Z}_q$ uniformly at random. After the initialization and synchronization steps (omitted from Figure 3), D transmits the integrity-protected form of $m_D : ID_D, z_D$ to the BS, while the helper is injecting m_H on slot positions denoted by $\mathbf{s}$. Here, we opt to protect both $h(m_D)$ and m_D with the PHY-layer primitive to conceal the value of m_D from an adversary A, who cannot learn the helper's sequence m_H. This prevents a rogue BS from recovering m_D, so that it cannot pair with the device successfully. The helper then reveals $\mathbf{s}$ to the BS through the secret channel implemented by AE$(\cdot)$. The BS uses $\mathbf{s}$ to verify the integrity of m_D and recover z_D. BS replies with $z_{BS} = g^{X_{BS}}$, where X_{BS} is chosen in $\mathbb{Z}_q$ uniformly at random. Each party independently calculates $k_{D,BS} = g^{X_D \cdot X_{BS}}$. Immediately following the key-agreement, D and BS engage in a key confirmation phase, initiated by D. This can be done by executing a two-way challenge-response protocol [4], as shown in Figure 4. If any of the verification steps fail, the corresponding party aborts the pairing protocol.

D		BS
Given ID_D,		Given ID_{BS},
$(\mathbb{G}, q, g)$		$(\mathbb{G}, q, g)$
Pick $X_D \in_U \mathbb{Z}_q$		$X_{BS} \in_U \mathbb{Z}_q$
$z_D \leftarrow g^{X_D}$		$z_{BS} \leftarrow g^{X_{BS}}$
$m_D \leftarrow ID_D, z_D$		$m_{BS} \leftarrow ID_{BS}, z_{BS}$
(H active)	$\xrightarrow[+m_H]{[h(m_D)],m_D}$	
(H active)	$\xrightarrow{\text{AE}(\mathbf{s},K)}$	Verify & Extract z_D
	$\xleftarrow{m_{BS}}$	
$k_{D,BS} \leftarrow (z_{BS})^{X_D}$		$k_{D,BS} \leftarrow (z_D)^{X_{BS}}$

Figure 3: Diffie-Hellman key-agreement on $k_{D,BS}$ using the HELP PHY-layer integrity verification method.

5 Security Analysis

In this section, we analyze the security of the HELP integrity verification primitive and evaluate the security of the DH-based pairing protocol presented in Section 4.3.

5.1 Security of the HELP primitive

Consider the transmission of $[h(m_D)], m_D$ from D to the BS, superimposed with the transmission of m_H. The goal of the adversary A is to replace m_D with some desired m'_D and pass the verification at the BS. In the absence of the helper, a straightforward strategy for A is to annihilate $[h(m_D)], m_D$ and inject $[h(m'_D)], m'_D$. However, when m_H is superimposed on $[h(m_D)]$, a cancellation of $[h(m_D)] + m_H$ leads to the likely detection of the cancellation attack due to the "erasure" of the helper's ON slots.

Rather than blindly canceling the composite signal $[h(m_D)] + m_H$ transmitted by D and H, the adversary can attempt to detect the ON slots of the helper and leave those intact. He can then target only the OFF symbols of m_H and modify those to desired values so that the BS decodes m'_D. To pass the integrity verification performed by

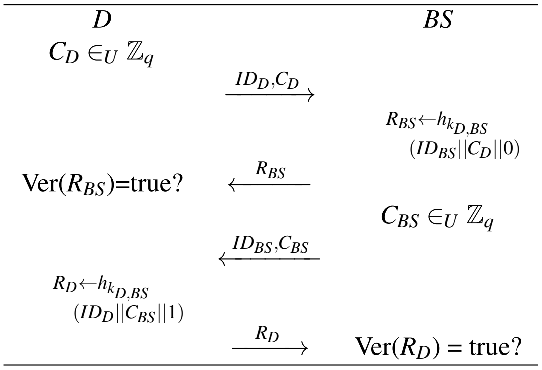


Figure 4: Key confirmation of $k_{D,BS}$ using a challenge-response protocol.

the *BS*, it must hold that (a) all the ON slots indicated in $\mathbf{s}$ are also ON slots in $[h(m'_D)] + m_H$, and (b) the removal of m_H during step 8 of HELP (see Section 4.2), leads to the decoding of $[h(m'_D)]$. As m_D follows in plaintext, the adversary can then replace m_D with m'_D.

We first show that if the adversary can identify the ON slots of the helper (this is equivalent to knowing m_H), then it can modify the transmitted signal such that the desired value m'_D is decoded at the *BS*. Consider the transmission of one MC ON-OFF bit b_D and the superposition of an ON slot by H either during the ON or the OFF slot of the coded b_D. The possible outcomes of this superposition are shown in the third column of Table 1. Moreover, we show the signal b_A that must be injected by A to cause the decoding of the desired value b'_D at the *BS*. For illustration purposes, we show the signal cancellation as a negation of the ON value.

From Table 1, we observe that if b_H is known, the adversary can always make the *BS* decode the desired bit b'_D, irrespective of the value of b_D. Moreover, since the ON bits of m_H stay intact, the modified signal will pass the PHY-layer integrity verification at the *BS*. However, identifying the ON slots of the helper is difficult due to the location proximity between D and H and also the strict reaction time necessary to perform the cancellation attack in an online fashion. In the next proposition, we prove the security of the integrity verification mechanism under the realistic assumption that an ON slot for the helper is timely identified by A with some probability. We experimentally evaluate this probability in Section 6. The security of the integrity verification of HELP is given by Proposition 1.

Proposition 1. *The HELP integrity verification primitive is δ–secure with*

$$\delta = \left(1 - \frac{1-p_I}{4}\right)^{|\mathbf{s}|}. \tag{1}$$

Here δ is the probability that the BS accepts a message forgery by A, $|\mathbf{s}|$ is the length of the vector indicating the number of the helper's ON slots, and p_I is the probability of inferring the helper's activity during one MC ON-OFF bit when D and H do not co-transmit. Here, δ is a negligible function of $|\mathbf{s}|$. In eq. (1)*, it is assumed that a strongly universal hash function is used as part of the HELP primitive.*

Table 1: Injection of desired bit b'_D, when the ON slots of the helper can be detected.

	b_D	b_H	$b_D + b_H$	b_A	$b_D + b_H$ $+ b_A$	b'_D
1	_⎍	_⎍	_⎍	_⎍	_⎍	_⎍
2	_⎍	_⎍	_⎍	⎍_	⎍⎍	⎍_
3	_⎍	⎍_	⎍⎍	_⎍	⎍⎍	_⎍
4	_⎍	⎍_	⎍⎍	⎍⊔	⎍_	⎍_
5	⎍_	_⎍	⎍⎍	⊔⎍	_⎍	_⎍
6	⎍_	_⎍	⎍⎍	⎍_	⎍⎍	⎍_
7	⎍_	⎍_	⎍_	_⎍	⎍⎍	_⎍
8	⎍_	⎍_	⎍_	⎍_	⎍_	⎍_

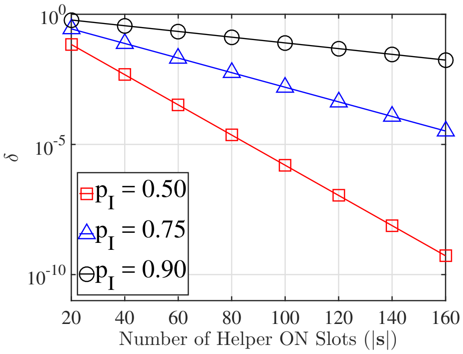


Figure 5: Probability of accepting a forged message m'_D at the *BS* as a function of $|\mathbf{s}|$, for varying inference capabilities of helper activity.

Proof. The proof is provided in Appendix A. □

In our analysis, we set the inference probability of H's activity to one when either D and H co-transmit or none transmits. In the former case, the presence of high power can be used to detect the superposition of D and H ON slots, and hence infer H's ON slot. In the latter case, the absence of power can be used to detect a helper's OFF slot. When either D or H are active, the inference probability is set to $p_I < 1$ due to the ambiguity in deciding which of the two devices is active. Summarizing,

$$\Pr[\text{Inference}] = \begin{cases} 1, & D \text{ \& } H \text{ transmit} \\ 1, & D \text{ \& } H \text{ do not transmit} \\ p_I, & \text{D or } H \text{ transmits.} \end{cases} \quad (2)$$

In Proposition 1, δ depends on two variables; the cardinality of $\mathbf{s}$ and p_I. From (1), it is evident that δ is a negligible function of $|\mathbf{s}|$, and a monotonically increasing function of p_I. In Figure 5, we show δ as a function of $|\mathbf{s}|$ for various values of p_I. As expected, a higher p_I yields a higher δ value for the adversary. For instance, when $p_I = 0.9$, $\delta = 0.0174$, when $|\mathbf{s}| = 160$, which may not be acceptable. However, doubling the size of $\mathbf{s}$ lowers δ to 0.0003. Note that in a single use of the HELP primitive, the attacker has only one chance to guess $\mathbf{s}$ and modify the value of m_D in an online fashion. Hence, a higher probability of forgery is acceptable here relative to standard cryptographic security (similar security values are sought in previous pairing protocols, which use short authentication strings [40]).

5.2 Security of the Device Pairing Protocol

We now analyze the security of the device pairing protocol proposed in Section 4.3. Since the security of the DH key-agreement protocol under a passive adversary is standard [56], we focus on the security under active attacks. We divide our analysis into two parts. In the first part, we examine if the adversary can pair a rogue device to a legitimate *BS*. In the second part, we examine if a legitimate device can be deceived to pair with a rogue base station. These two steps are part of a MitM attack.

5.2.1 Pairing a Rogue Device with a legitimate *BS*

The pairing of a rogue device D' with the *BS* can occur under two different scenarios: (a) D' pairs in the absence of a legitimate device D, and (b) D' pairs while D and the *BS* execute a pairing session.

Pairing in the absence of a legitimate device: The pairing protocol described in Section 4.3 is initiated with the placement of H in close proximity to the legitimate device and the press of a button on H and D, respectively. The button pressing sends a pairing initialization message to the *BS* which is authenticated using the secure $\text{AE}(\cdot)$ function. Without access to the helper device, the adversary cannot initiate the pairing process from a remote location.

Hijacking a legitimate pairing session: Since A cannot initiate the pairing process with the *BS*, he can only attempt to pair a rogue device with the *BS* by hijacking a pairing session involving a legitimate device D. To establish a secret key with the *BS*, the adversary must modify the DH public number z_D of D into its own DH public number z'_D, where z_D is contained in the first message m_D sent from D to the *BS* (similar to a typical MitM attack against a DH key exchange).

However, m_D is protected by our integrity verification primitive. Note that in the HELP primitive, only $h(m_D)$ is encoded using MC ON-OFF keying while m_H is being superimposed. The actual value of m_D follows in plaintext. In our proposed modified DH protocol, both $h(m_D)$ and m_D are encoded using HELP. According to Proposition 1, the adversary's success probability in forging m_D in the HELP primitive is δ. When both $h(m_D)$ and m_D are encoded using HELP, we claim that the adversary's success probability in replacing m_D is upper bounded by δ. This is because in the primitive, the adversary can change m_D into any m'_D with probability 1, but his advantage is limited by the probability of changing $h(m_D)$ into $h(m'_D)$, which is δ. In the pairing protocol, the adversary's success probability of changing m_D into m'_D is less or equal to 1. Thus overall, its success probability is less or equal to δ, which is a negligible function of $|\mathbf{s}|$ (number of ON slots injected by helper during $[h(m'_D)]$). Therefore, the adversary will be unable to pair D' with the legitimate *BS*.

5.2.2 Pairing D with a Rogue Base Station

We now examine whether the adversary acting as a rogue *BS* can pair with a legitimate device D. To do so, the adversary can perform a similar MitM attack as in the uplink direction, by replacing the *BS*'s DH public parameter z_{BS} with its own number $z_{BS'}$. This step of the MitM attack corresponding to the message sent by A to D after the reception of m_D is shown in Figure 6.

For this attack to be successful, the adversary must extract the DH public value z_D so that it can compute $k_{D,BS'} = (z_D)^{X_{BS'}}$. The value of z_D is carried in $[h(m_D), m_D] + m_H$, using the HELP primitive. To recover m_D, the adversary must be able to determine the location vector $\mathbf{s}$ that is used to generate m_H for the portion that corresponds to the transmission of m_D. However, $\mathbf{s}$ is transmitted from H to *BS* using the authenticated encryption function $\text{AE}(\cdot)$, so A cannot obtain $\mathbf{s}$ directly from the encrypted version of it.

Alternatively, A can collect and analyze the transmitted signal of $[h(m_D), m_D] + m_H$ after receiving it and attempt to identify all the ON slots in m_H using radio fingerprinting methods [7,19,28,39,55,59]. However, none of the fingerprinting methods can achieve 100% accuracy. As long as A infers H's ON slots with some probability smaller than one, we can drive the probability of successfully extracting m_D arbitrarily low by increasing

D		A		BS
Given ID_D, $(\mathbb{G}, q, g)$		Given $ID_{D'}$, $(\mathbb{G}, q, g)$		Given ID_{BS}, $(\mathbb{G}, q, g)$
Pick $X_D \in_U \mathbb{Z}_q$		$X_{D'} \in_U \mathbb{Z}_q$		$X_{BS} \in_U \mathbb{Z}_q$
$z_D \leftarrow g^{X_D}$		$z_{D'} \leftarrow g^{X_{D'}}$		$z_{BS} \leftarrow g^{X_{BS}}$
$m_D \leftarrow ID_D, z_D$		$m'_D \leftarrow ID_{D'}, z_{D'}$		$m_{BS} \leftarrow ID_{BS}, z_{BS}$
(H active)	$\xrightarrow{[h(m_D), m_D] + m_H}$	Cancel and inject	$\xrightarrow{[h(m'_D), m'_D]}$	
(H active)	$\xrightarrow{\mathrm{AE}(\mathbf{s}, K)}$		$\xrightarrow{\mathrm{AE}(\mathbf{s}, K)}$	Verify & Extract $z_{D'}$
			$\xleftarrow{m_{BS}}$	
		$k'_{D',BS} \leftarrow (z_{BS})^{X_{D'}}$		$k'_{D',BS} \leftarrow (z_{D'})^{X_{BS}}$
		$X_{BS'} \in_U \mathbb{Z}_q$		
		$z_{BS'} \rightarrow g^{X_{BS'}}$		
		$m_{BS'} \rightarrow ID_{D'}, z_{BS'}$		
	$\xleftarrow{m_{BS'}}$			
		Recover z'_D		
$k_{D,BS'} \leftarrow (z_{BS'})^{X_D}$		$k'_{D,BS'} \leftarrow (z'_D)^{X_{BS'}}$		

Figure 6: MitM attack against the key-agreement phase of HELP-enabled pairing protocol.

the number of slots carrying m_D.

In the following proposition, we derive the probability of D successfully pairing with a rogue BS, when the ON slots of the helper are inferred with probability p'_I. Note that in general p'_I is different than the p_I of Proposition 1. The inference of the helper's ON slots in Proposition 1 must occur based on very few samples because the adversary must quickly decide whether to perform signal cancellation. In the rogue BS case, the adversary can analyze $[h(m_D), m_D] + m_H$ based on all the samples, so it is expected that $p'_I > p_I$.

Proposition 2. *A legitimate device D pairs with a rogue BS with probability $\delta + \varepsilon$, where*

$$\delta = (p'_I)^{|\mathbf{s}'|}, \tag{3}$$

and ε is a negligible function of the hash length. Here $|\mathbf{s}'| < |\mathbf{s}|$ corresponds to the number of helper's ON slots only during the transmission of m_D in the $[h(m_D), m_D]$, p'_I is the probability of inferring the helper's activity during one MC ON-OFF bit when D and H do not co-transmit, and δ is a negligible function of $|\mathbf{s}'|$ when $p'_I < 1$.

Proof. The proof is provided in Appendix B. □

In Proposition 2, δ depends on two variables; the cardinality of set $\mathbf{s}'$ which is a subset of $\mathbf{s}$ corresponding to H's ON signal only during the transmission of m_D in $[h(m_D), m_D]$, and the inference probability of the helper's activity during the transmission of $[h(m_D), m_D] + m_H$, which is p'_I. From eq. (3), it is evident that δ is a negligible function of $|m_D|$, and a monotonically increasing function of p'_I. In Figure 7, we show δ as a function of $|\mathbf{s}'|$ for various values of p'_I and fixed hash length of $\ell = 160$. As expected, a higher p'_I yields a higher δ value for the adversary. For instance, when $p'_I = 0.9$, $\delta = 0.0018$, when $|\mathbf{s}'| = 80$, which may not be acceptable. However, doubling the size of $\mathbf{s}'$ lowers δ to 5×10^{-8}. Note that, such an attack has to happen in an online manner. This is because the rogue BS must pass the challenge-response phase from the device in the key confirmation phase, so the attacker only has one chance to guess $\mathbf{s}$ and derive a probable DH key from the guessed z_D, which is only successful with small probability δ (similar to limited-guess online password attacks).

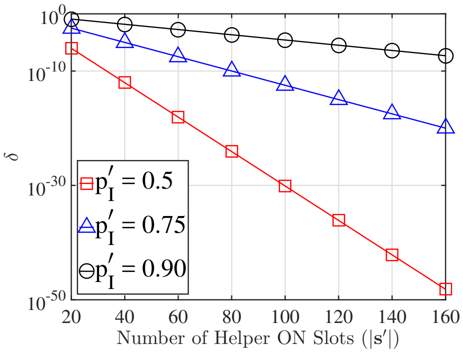


Figure 7: Probability of pairing with a rogue BS as a function of $|\mathbf{s}|$, for varying inference capabilities of helper activity.

6 Evaluation

6.1 Helper Activity Inference

In this section, we first analyze A's capability in timely identifying the helper's ON slot when the helper is transmitting the ON-OFF message m_H. For this purpose, the adversary could employ several PHY-layer characteristics of the helper's transmission to pinpoint when H is active. These include (a) the received signal strength [55], (b) the frequency offset [59], (c) the channel impulse response $\mathbf{h}_{H,A}$ [39], (d) the I/Q origin offset [7], (e) the transient radio state [19], and (f) the angle of arrival for the incoming signal [28].

We first examine A's attempt to perform the signal cancellation and injection required by the MitM attack of Figure 6. To avoid rejection of m'_D by the *BS*, the adversary has to *swiftly detect* a helper's ON slot and decide whether to perform signal cancellation. Most existing radio fingerprinting methods are not suitable for such quick online detection. The frequency offset and channel impulse response are estimated using known preambles that are typically included in headers. Such preambles do not precede the helper's ON slots. The I/Q origin offset is not a suitable method because we employ ON-OFF modulation for message transmission. The methods that detect the transient state of a radio when it turns on can only be used to identify the start of a transmission (although an ON-OFF modulation implies a transition from an OFF to an ON state, the radio transmitter is powered through the entire transmission of an ON-OFF signal and a transient state is not observed with every slot). Differentiating between D and H using an AOA requires a very narrow directional beam due to the proximity between H and D. Such narrow beamwidths can be achieved by using an antenna array [48] or a parabolic antenna [63]. However, the hardware cost is prohibitive and the antenna would be quite visible. For example, an adversary at 50ft from D and H requires two 50-element antenna arrays pointed to D and H respectively via the LoS path, to differentiate between D and H when their distance is set to 4ft. This calculation assumes a 2.4GHz operating frequency.

6.1.1 Fast Helper Detection based on RSS

The simplest and most timely method for detecting the presence of the helper is to measure the received signal strength over some small number of samples at the beginning of every slot. Let b_D and b_H represent the bit simultaneously transmitted by D and H respectively over two slots t_i and t_{i+1}. There are four possible bit combinations that yield two candidate power profiles for $b_D + b_H$, as measured by the adversary. When $b_D = b_H$, the helper and D overlap in one of the two slots (either t_i or t_{i+1}), depending on the value of b_D, b_H. In this case, one of the slots is OFF whereas the other slot is ON with a significantly higher power because the two ON slots of H and D are superimposed (here, we have considered the worst-case scenario and ignored the possibility of destructive interference). We expect that A will be able to infer the ON slot of the helper with probability $p_I = 1$, due to the higher RSS value of the first few samples of the ON slot.

When $b_D \neq b_H$, both t_i and t_{i+1} are ON and have similar power profiles if H and D transmit with the same power and are placed in close proximity. In this case, the adversary is expected to be unable to differentiate a helper's ON slot from a device's ON slot with the probability much higher than a random guess. The four possible cases for one slot observed by the adversary are: (a) P_1 : both H and D are ON, (b) P_2 : H is ON and D is OFF, (c) P_3 : D is ON and H is OFF, and (d) P_4 : both H and D are OFF. For each case, the adversary determines four threshold values $E[P_1], E[P_2], E[P_3]$, and $E[P_4]$, that represent the average expected power, as measured by the first few samples of a slot.

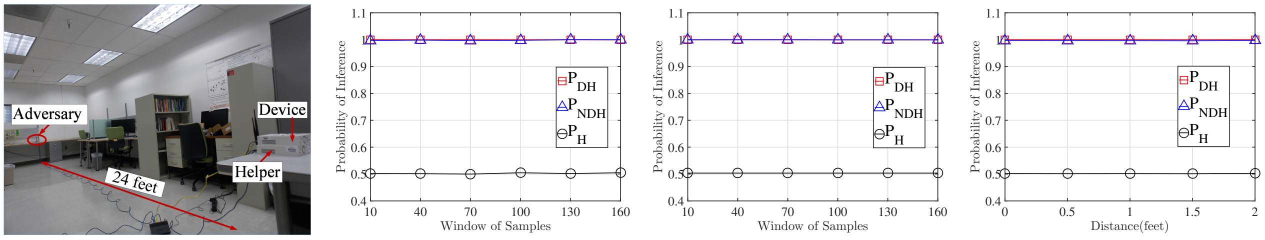
Without loss of generality, let $E[P_1] > E[P_2] > E[P_3] > E[P_4]$.[2] Let also $E[P(t_i)]$ denote the average power measured over slot t_i using the first few samples. The adversary classifies t_i to one of four cases by mapping $E[P(t_i)]$ to the closest threshold. That is, case P_1 is inferred if $E[P(t_i)] > \frac{E[P_1]+E[P_2]}{2}$, case P_2 is inferred if $\frac{E[P_1]+E[P_2]}{2} \leq E[P(t_i)] < \frac{E[P_2]+E[P_3]}{2}$, etc. A wrong inference is made when $E[P(t_i)]$ that belongs to case P_i is mapped to a case P_j with $P_i \neq P_j$. In Proposition 1, we have assumed that the probability p_I for correctly inferring cases P_1 and P_4 is equal to one. In P_1, the RSS is expected to be relatively high due to the co-transmission from D and H. In P_4, the RSS is expected to be low because neither D nor H are transmitting. However, the thresholds for cases P_2 and P_3 are expected to be very close, thus leading to frequent wrong inferences. We experimentally verify this claim.

Experimental Evaluation of p_I: *Experimental setup:* To evaluate p_I, we setup three NI-USRP 2921 devices in an indoor laboratory environment. Two USRP devices represented D and H, whereas a third USRP device is placed at 24 feet away acting as an adversary. The transmit power for an ON slot was set to 20dBm for both D and H with a symbol duration of 1ms. The devices were set to work at 2.4GHz and were synchronized. The sampling frequency was set to 2MHz. We tested two scenarios: (1) H is stacked on top of D, and (2) H is moved away from the legitimate device. The experiment setup is shown in Figure 8(a).

We implemented amplitude shift keying (ASK) to transmit MC ON-OFF coded messages and repeatedly transmitted message $\{1,0,1,0\}$ from D and message $\{1,1,0,0\}$ from H simultaneously. The signals from H are MC-coded only when the bit value is one. The superposition of the two signals implemented all four cases P_1-P_4.

Results: Let P_{DH} denote the probability of detecting that D and H transmit simultaneously, P_{NDH} denote the prob-

[2] $E[P_2]$ and $E[P_3]$ can be similar but not exactly the same, so we can assume some ordering to make a classification rule.



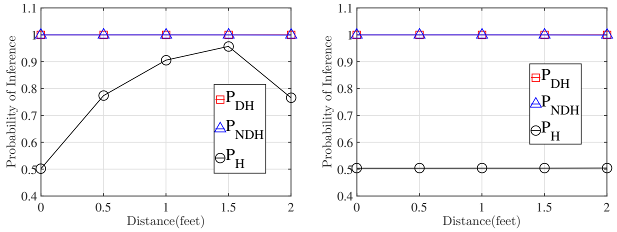
(a) (b) (c) (d)

Figure 8: (a) Experimental setup, (b) detection probability as a function of the window of samples when the power at H and D is fixed, (c) detection probability as a function of the window of samples when the power at H and D varies, and (d) detection probability as a function of the distance between D and H, when H and D remain equidistant from A.

ability of detecting that neither D nor H transmit, and P_H denote the probability of detecting that H is transmitting alone. These correspond to p_I for any of the candidate scenarios. In the first experiment, we measured the detection probability as a function of the sampling window size used for computing the average RSS value for a given slot. Intuitively, a longer sampling window would lead to better inference but will delay the cancellation operation. Figure 8(b) shows the resulting detection probabilities as a function of the sample window. We observe that the detection probabilities P_{DH} and P_{NDH} are relatively low and are further reduced with the increase of the sample window. However, the detection probability P_H is close to 0.5 irrespective of the sample window size. This indicates that differentiating between the ON slots of the helper and of the legitimate device, when only one of the two transmits, is practically equivalent to a random guess. Our results justify the selection of $p_I = 1$ when the H and D are simultaneously absent or present, and $p_I = 0.5$ otherwise.

In the second experiment, we repeated the first experiments but configured H and D to vary their transmission power on a per-slot basis. The power was varied to reduce the inference capability of A. Specifically, H and D oscillated their power at random between 10dBm and 20dBm. Figure 8(c) shows the detection probabilities as a function of the window of samples used for inference.

Effect of proximity on p_I: We further performed experiments to evaluate the effect of the proximity between D and H on their distinguishability. We repeated the first experiment and varied the distance between H and D. In the first part of the experiment, H was moved away from D while keeping the D-A and H-A distances similar (the helper's motion was perpendicular to the D-A line. Figure 8(d) shows that the detection probability for each case is similar to the case where H is stacked on top of D. In the second part of the experiment, H was moved towards A, and therefore, the distance between H and A was gradually reduced. Figure 9(a) shows the respective detection probabilities. As expected, decreasing the distance between A and H improves the adversary's inference capability, but the inference remains imperfect when D and H remain relatively close.



(a) (b)

Figure 9: (a) Detection probability as a function of the distance between D and H when H is moved towards A, and (b) detection probability as a function of the distance between D and H when H is moved towards A, when D and H are transmitting random powers.

In the fourth experiment, we repeated the second part of the third experiment but configured H and D to vary their transmission power on a per-slot basis. The power was varied to reduce the inference capability of A. Specifically, H and D oscillated their power at random between 10dBm and 20dBm. Figure 9(b) shows the same results when the distance between D and H was also varied, with H moving towards A. We observe that P_H remains a random guess even when H is moved away from D (comparison of P_H in Figures 9(a) and 9(b)), indicating that a power variation approach can account for situations where H is not placed exactly on top of D. Distinguishing signals from D and H using RSS remains a random guess even when H is 2ft away from D.

6.1.2 Fast Helper Detection Based on Time

In this section, we discuss an inference technique that exploits the possible time misalignment between the transmissions of H and D due to clock drift and different path delays to the receiver. There have been extensive studies on synchronization of independent wireless nodes, but practically it is impossible to reach perfect synchronization [51]. The adversary can exploit the synchronization offset between H and D to infer the presence of helper's ON signals. If H is faster (slower) than D, the ON slots of H will appear slightly earlier (later) than the ON slots of D. An example of a fast H is shown in Figure 10,

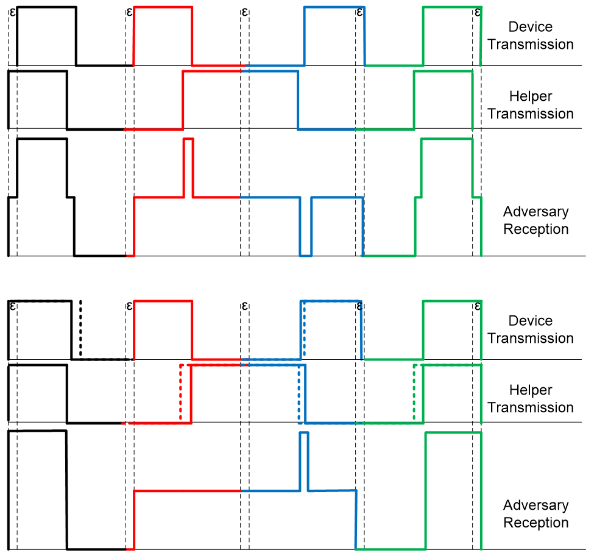


Figure 10: Synchronization offset without and with randomized start time of each bit.

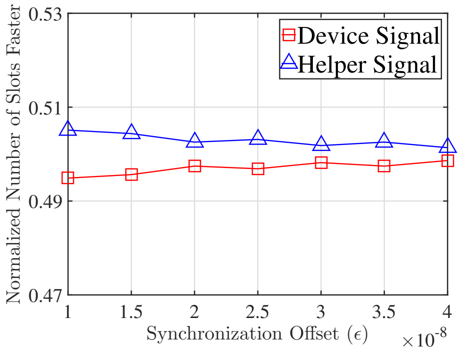


Figure 11: Fraction of slots that one device is faster than the other as a function of the delay offset ε.

where there is a synchronization offset ε between D and H. If A fixes his clock to H, it can infer the presence of helper's ON slots without having to resort to RSS estimation. It should be noted here, the BS performs detection of ON slots by taking an average value of the power of all the samples. Therefore, a perfect synchronization between D and H is not required for the correctness of the proposed protocol.

To prevent the inference of the helper's ON slots based on time misalignment, we randomize the start times of each bit (first slot of the MC ON-OFF bit) both at H and D. Specifically, a random time offset ε, positive or negative, is selected from a uniform distribution $\mathscr{U}(\varepsilon_l, \varepsilon_h)$. The lower bound ε_l is selected to be the maximum synchronization error between D and H. This can be calculated as the expected clock drift over the transmission time of H plus a maximum time difference between path delays. The upper bound τ_h can be some reasonable value (e.g., $2\varepsilon_l$). Moreover $\tau << t$, where t is the slot duration. This will ensure the correct sequence decoding at the BS. The lower part of Figure 10 shows an example of applying the randomized start time for each bit. We observe that no device is always faster (slower), thus preventing A from fixing its clock to H.

Experimental Evaluation of p_I: To verify the validity of our time randomization approach and its impact on the inference probability p_I, we setup three NI-USRP 2921 devices in an indoor laboratory environment as D, H, and A, respectively. As in previous experiments, H was stacked on top of D, whereas A was placed 24 feet away from D, H. The transmit power for an ON slot was set to 20dBm with a symbol duration of 1ms. An artificial clock misalignment $\tau = 0.1$msec was set between H and D to emulate the maximum synchronization error. We then varied the random time offset ε selected by H and D. The experiment lasted for the transmission of 10^6 sequences of 40 bits each.

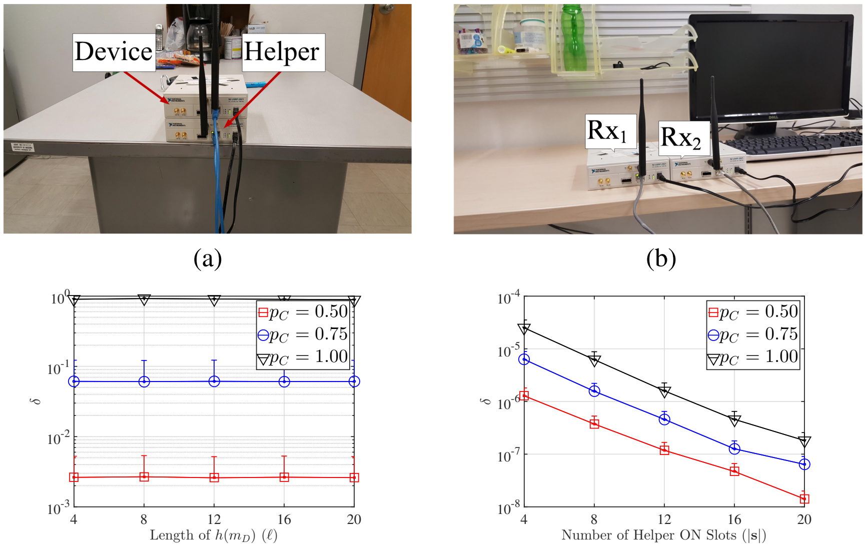
Figure 11 shows the fraction of slots for which each device was detected to be faster as a function of the maximum synchronization error ε. We observe that for sufficiently high values of ε, H is almost 50% of the time faster than D. Practically, using time misalignment to distinguish the helper becomes a random guess.

6.2 Protocol Evaluation

In the final set of experiments, we evaluated the integrity protection offered by HELP against an adversary capable of canceling and injecting signals. We setup two USRP devices stacked over each other as D and H, one device (Rx_1) at 24ft from D, H acting as the BS and a second device RX_2 set by RX_1 that performed cancellation on RX_1. The transmitters and the receivers are shown in Figure 12(a) and Figure 12(b), respectively. The distance between the two receivers was set to approximately one wavelength λ to cause signal inversion at RX_1. After receiving the transmissions of D and H at Rx_1 and Rx_2, cancellation was performed via signal processing in MATLAB [34]. The signal of RX_2 was added to RX_1 to cancel the transmission of D and H, whereas a random signal was added to emulate A's signal injection.

In the first scenario, we transmitted MC ON-OFF sequences of length $\ell = \{4, 8, 12, 20\}$, while the helper was inactive. We measured the probability δ of accepting A's random sequence at the BS (RX_1). We also varied the probability of successful cancellation p_C by suppressing cancellation for a corresponding fraction of bits. Figure 12(c), shows δ as a function of ℓ for various p_C. We observe that for high cancellation probability values p_C, a message cancellation/injection has a high success probability (close to one).

We repeated the experiment of the first scenario in the presence of H who transmitted at random slot locations simultaneously with D. In the experiment, the adversary



(a) (b)

(c) (d)

Figure 12: (a) Placement of D and H, (b) placement of the BS (RX_1) and RX_2. (c) probability of acceptance of a modified message at the BS in the absence of H, and (d) probability of acceptance of a modified message at the BS in the presence of H.

attempted to distinguish between D and H using the RSS sampling method discussed in Section 6.1.1. Also, the adversary canceled slots on which D or H's signals were indistinguishable. Figure 12(d) shows the probability δ of accepting the adversary's modified message as a function of the number of active helper slots $|\mathbf{s}|$ when the message length is $\ell = 20$. We observe that δ decreases drastically compared to Figure 12(c). Moreover, imperfect cancellation ($p_C < 1$) leads to further deterioration of the adversary's performance. The results obtained support the analytical results provided in Section 5, which are computed assuming $p_C = 1$.

Timing performance: The upper bound on the execution time of the DH protocol with HELP primarily depends on the communication time of the ON-OFF keyed message, since the rest of the messages are exchanged in the normal communication mode. Public key parameters for an EC-DH key-agreement [58] can have values from 160–512 bits, depending on the security requirement. Assuming a hash length of 160 bits and a slot duration of 1ms, the time required to transmit the HELP protected DH public primitive varies between 0.6–1.4s, which is acceptable.

7 Conclusion

We considered the problem of pairing two devices using in-band communications in the absence of prior shared secrets. We proposed a new PHY-layer integrity protection scheme called HELP that is resistant to signal cancellation attacks. Our scheme operates with the assistance of a helper device that has an authenticated channel to the BS. The helper is placed in close proximity to the legitimate device and simultaneously transmits at random times to allow the detection of cancellation attacks at the BS. We showed that a pairing protocol such as the DH key agreement protocol using HELP as an integrity protection primitive can resist MitM attacks without requiring an authenticated channel between D and the BS. This was not previously feasible by any of the pairing methods if signal cancellation is possible. We studied various implementation details of HELP and analyzed its security. Our protocol is aimed at alleviating the device pairing problem for IoT devices that may not have the appropriate interfaces for entering or pre-loading cryptographic primitives.

Acknowledgments

We thank our shepherd Manos Antonakakis and the anonymous reviewers for their insightful comments. This research was supported in part by the NSF under grant CNS-1409172 and CNS-1410000. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of the NSF.

References

[1] BALFANZ, D., SMETTERS, D. K., STEWART, P., AND WONG, H. C. Talking to strangers: authentication in ad-hoc wireless networks. In *Proc. of NDSS'02* (2002).

[2] BELLARE, M., AND NAMPREMPRE, C. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In *Proc. of International Conference on the Theory and Application of Cryptology and Information Security* (2000), Springer, pp. 531–545.

[3] Bichler, D., Stromberg, G., Huemer, M., and Löw, M. Key generation based on acceleration data of shaking processes. In *Proc. of 9th international conference on Ubiquitous computing* (Berlin, Heidelberg, 2007), UbiComp'07, Springer-Verlag.

[4] Boyko, V., MacKenzie, P., and Patel, S. Provably secure password-authenticated key exchange using diffie-hellman. In *Proc. of International Conference on the Theory and Applications of Cryptographic Techniques* (2000), Springer, pp. 156–171.

[5] Brands, S., and Chaum, D. Distance-bounding protocols. In *Proc. of Advances in Cryptology EUROCRYPT 93* (1994), Springer, pp. 344–359.

[6] Brik, V., Banerjee, S., Gruteser, M., and Oh, S. Wireless device identification with radiometric signatures. In *Proc. of 14th ACM international conference on Mobile computing and networking* (2008), ACM, pp. 116–127.

[7] Brik, V., Banerjee, S., Gruteser, M., and Oh, S. Wireless device identification with radiometric signatures. In *Proc. of 14th ACM international conference on Mobile computing and networking* (2008), ACM, pp. 116–127.

[8] Cagalj, M., Capkun, S., and Hubaux, J.-P. Key agreement in peer-to-peer wireless networks. In *Proc. of IEEE* (Feb. 2006), vol. 94, pp. 467–478.

[9] Cai, L., Zeng, K., Chen, H., and Mohapatra, P. Good neighbor: Ad hoc pairing of nearby wireless devices by multiple antennas. In *Proc. of Network and Distributed System Security Symposium* (2011).

[10] Capkun, S., Cagalj, M., Rengaswamy, R., Tsigkogiannis, I., Hubaux, J.-P., and Srivastava, M. Integrity codes: Message integrity protection and authentication over insecure channels. *IEEE Transactions on Dependable and Secure Computing 5*, 4 (2008), 208–223.

[11] Chen, C.-H. O., Chen, C.-W., Kuo, C., Lai, Y.-H., McCune, J. M., Studer, A., Perrig, A., Yang, B.-Y., and Wu, T.-C. Gangs: gather, authenticate 'n group securely. In *Proc. of MobiCom'08* (2008), pp. 92–103.

[12] Cornelius, C., and Kotz, D. Recognizing whether sensors are on the same body. In *Proc. of 9th international conference on Pervasive computing* (Berlin, Heidelberg, 2011), Pervasive'11, Springer-Verlag.

[13] Danev, B., Heydt-Benjamin, T., and Čapkun, S. Physical-layer identification of rfid devices. In *Proc. of the 18th conference on USENIX security symposium* (2009), USENIX Association, pp. 199–214.

[14] Diffie, W., and Hellman, M. E. New directions in cryptography. *IEEE Transactions on Information Theory 22*, 6 (1976), 644–654.

[15] Dolev, D., and Yao, A. C. On the security of public key protocols. *Information Theory, IEEE Transactions on 29*, 2 (1983), 198–208.

[16] Franklin, J., McCoy, D., Tabriz, P., Neagoe, V., Randwyk, J., and Sicker, D. Passive data link layer 802.11 wireless device driver fingerprinting. In *Proc. 15th USENIX Security Symposium* (2006), pp. 167–178.

[17] Gollakota, S., Ahmed, N., Zeldovich, N., and Katabi, D. Secure in-band wireless pairing. In *Proc. of USENIX security symposium* (2011), San Francisco, CA, USA, pp. 1–16.

[18] Goodrich, M. T., Sirivianos, M., Solis, J., Tsudik, G., and Uzun, E. Loud and clear: Human-verifiable authentication based on audio. In *Proc. of IEEE ICDCS 2006* (2006), p. 10.

[19] Hall, J., Barbeau, M., and Kranakis, E. Enhancing intrusion detection in wireless networks using radio frequency fingerprinting. In *Proc. of Communications, Internet, and Information Technology* (2004), pp. 201–206.

[20] Harland, C. J., Clark, T. D., and Prance, R. J. Electric potential probes - new directions in the remote sensing of the human body. *Measurement Science and Technology 13*, 2 (2002), 163.

[21] Hei, X., and Du, X. Biometric-based two-level secure access control for implantable medical devices during emergencies. In *Proc. of 30th IEEE International Conference on Computer Communications* (Shanghai, P.R.China, April 2011), pp. 346 – 350.

[22] Hou, Y., Li, M., Chauhan, R., Gerdes, R. M., and Zeng, K. Message integrity protection over wireless channel by countering signal cancellation: Theory and practice. In *Proc. of AsiaCCS Symposium* (2015), pp. 261–272.

[23] Hou, Y., Li, M., and Guttman, J. D. Chorus: Scalable inband trust establishment for multiple constrained devices over the insecure wireless channel. In *Proc. of WiSec Conference* (2013), pp. 167–178.

[24] Hu, B., Zhang, Y., and Lazos, L. PHYVOS: Physical layer voting for secure and fast cooperation. In *Proc. of IEEE Conference on Communications and Networks Security* (2015).

[25] Kalamandeen, A., Scannell, A., de Lara, E., Sheth, A., and LaMarca, A. Ensemble: cooperative proximity-based authentication. In *Proc. of 8th international conference on Mobile systems, applications, and services* (New York, NY, USA, 2010), MobiSys '10, ACM, pp. 331–344.

[26] Kumar, A., Saxena, N., Tsudik, G., and Uzun, E. Caveat eptor: A comparative study of secure device pairing methods. In *Proc. of IEEE PerCom '09* (2009), pp. 1–10.

[27] Kuo, C., Luk, M., Negi, R., and Perrig, A. Message-in-a-bottle: user-friendly and secure key deployment for sensor nodes. In *Proc. of SenSys'07* (2007), pp. 233–246.

[28] Laitinen, H., Lahteenmaki, J., and Nordstrom, T. Database correlation method for gsm location. In *Proc. of 53rd IEEE Vehicular Technology Conference* (2001), vol. 4, IEEE, pp. 2504–2508.

[29] Laur, S., and Pasini, S. SAS-Based Group Authentication and Key Agreement Protocols. In *Proc. of Public Key Cryptography - PKC'08* (2008), LNCS, pp. 197–213.

[30] Law, Y., Moniava, G., Gong, Z., Hartel, P., and Palaniswami, M. Kalwen: A new practical and interoperable key management scheme for body sensor networks. *Security and Communication Networks* (2010).

[31] Li, M., Yu, S., Guttman, J. D., Lou, W., and Ren, K. Secure ad hoc trust initialization and key management in wireless body area networks. *ACM Trans. Sen. Netw. 9*, 2 (Apr. 2013), 18:1–18:35.

[32] Lin, Y.-H., Studer, A., Hsiao, H.-C., McCune, J. M., Wang, K.-H., Krohn, M., Lin, P.-L., Perrig, A., Sun, H.-M., and Yang, B.-Y. Spate: small-group pki-less authenticated trust establishment. In *Proc. of Mobisys'09* (2009), pp. 1–14.

[33] Mathur, S., Miller, R., Varshavsky, A., Trappe, W., and Mandayam, N. Proximate: proximity-based secure pairing using ambient wireless signals. In *Proc. of 9th international conference on Mobile systems, applications, and services* (New York, NY, USA, 2011), MobiSys '11, ACM, pp. 211–224.

[34] MATLAB. *version 9.0.0.341360 (R2016a)*. The MathWorks Inc., Natick, Massachusetts, 2016.

[35] Mayrhofer, R., and Gellersen, H. Shake well before use: Authentication based on accelerometer data. In *Proc. of International Conference on Pervasive Computing* (2007), Springer, pp. 144–161.

[36] Mayrhofer, R., and Gellersen, H. Shake well before use: Intuitive and secure pairing of mobile devices. *IEEE Transactions on Mobile Computing 8* (2009), 792–806.

[37] McCune, J. M., Perrig, A., and Reiter, M. K. Seeing-is-believing: Using camera phones for human-verifiable authentication. In *Proc. of IEEE S & P* (2005), pp. 110–124.

[38] Miettinen, M., Asokan, N., Nguyen, T. D., Sadeghi, A.-R., and Sobhani, M. Context-based zero-interaction pairing and key evolution for advanced personal devices. In *Proc. of the CCS Conference* (2014), pp. 880–891.

[39] Nerguizian, C., Despins, C., and Affès, S. Geolocation in mines with an impulse response fingerprinting technique and neural networks. *IEEE Transactions on Wireless Communications 5*, 3 (2006), 603–611.

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

[41] Nithyanand, R., Saxena, N., Tsudik, G., and Uzun, E. Groupthink: Usability of secure group association for wireless devices. In *Proc. of 12th ACM international conference on Ubiquitous computing* (2010), ACM, pp. 331–340.

[42] Pasini, S., and Vaudenay, S. SAS-based Authenticated Key Agreement. In *Proc. of Public Key Cryptography - PKC'06* (2006), vol. 3958 of *LNCS*, pp. 395 – 409.

[43] Patwari, N., and Kasera, S. Robust location distinction using temporal link signatures. In *Proc. of 13th annual ACM international conference on Mobile computing and networking* (2007), ACM, pp. 111–122.

[44] Perković, T., Čagalj, M., Mastelić, T., Saxena, N., and Begušić, D. Secure Initialization of Multiple Constrained Wireless Devices for an Unaided User. *IEEE transactions on mobile computing* (2011).

[45] Pierson, T. J., Liang, X., Peterson, R., and Kotz, D. Wanda: Securely introducing mobile devices. In *Proc. of IEEE INFOCOM–2016* (April 2016), pp. 1–9.

[46] Poon, C., Zhang, Y.-T., and Bao, S.-D. A novel biometrics method to secure wireless body area sensor networks for telemedicine and m-health. *IEEE Communications Magazine 44*, 4 (April 2006), 73–81.

[47] Pöpper, C., Tippenhauer, N. O., Danev, B., and Capkun, S. Investigation of signal and message manipulations on the wireless channel. In *Proc. of 16th European conference on Research in computer security* (2011), ESORICS'11, pp. 40–59.

[48] Rabinovich, V., and Alexandrov, N. Typical array geometries and basic beam steering methods. In *Antenna Arrays and Automotive Applications*. Springer, 2013, pp. 23–54.

[49] Rasmussen, K., Castelluccia, C., Heydt-Benjamin, T., and Capkun, S. Proximity-based access control for implantable medical devices. In *Proc. of 16th ACM conference on Computer and communications security* (2009), ACM, pp. 410–419.

[50] Rasmussen, K. B., and Čapkun, S. Realization of rf distance bounding. In *Proc. of 19th USENIX conference on Security* (2010), USENIX Security'10, pp. 25–25.

[51] Sampath, A., and Tripti, C. Synchronization in distributed systems. In *Advances in Computing and Information Technology*. Springer, 2012, pp. 417–424.

[52] Schürmann, D., and Sigg, S. Secure communication based on ambient audio. *IEEE Transactions on mobile computing 12*, 2 (2013), 358–370.

[53] Singh, K., and Muthukkumarasamy, V. Authenticated key establishment protocols for a home health care system. In *Proc. of ISSNIP'07* (Dec. 2007), pp. 353–358.

[54] Stajano, F., and Anderson, R. J. The resurrecting duckling: Security issues for ad-hoc wireless networks. In *Proc. of IWSP'00* (2000), pp. 172–194.

[55] Stella, M., Russo, M., and Begusic, D. Location determination in indoor environment based on rss fingerprinting and artificial neural network. In *Proc. of 9th International Conference on Telecommunications* (2007), IEEE, pp. 301–306.

[56] Stinson, D. R. *Cryptography: theory and practice*. CRC press, 2005.

[57] The Guardian. DDoS attack that disrupted internet was largest of its kind in history, experts say, 2016.

[58] Turner, S., Brown, D., Yiu, K., Housley, R., and Polk, T. Rfc 5480: Elliptic curve cryptography subject public key information. *Requests for Comments, Network Working Group, Tech. Rep* (2009).

[59] Ureten, O., and Serinken, N. Wireless security through rf fingerprinting. *Canadian Journal of Electrical and Computer Engineering 32*, 1 (2007), 27–33.

[60] Varshavsky, A., Scannell, A., LaMarca, A., and De Lara, E. Amigo: Proximity-based authentication of mobile devices. In *Proc. of 9th International Conference on Ubiquitous Computing* (2007), pp. 253–270.

[61] Venkatasubramanian, K., Banerjee, A., and Gupta, S. Pska: Usable and secure key agreement scheme for body area networks. *Information Technology in Biomedicine, IEEE Transactions on 14*, 1 (2010), 60–68.

[62] Venkatasubramanian, K., and Gupta, S. Physiological value-based efficient usable security solutions for body sensor networks. *ACM Transactions on Sensor Networks (TOSN) 6*, 4 (2010), 1–36.

[63] Visser, H. J. *Array and phased array antenna basics*. John Wiley & Sons, 2006.

[64] Xu, F., Qin, Z., Tan, C., Wang, B., and Li, Q. Imdguard: Securing implantable medical devices with the external wearable guardian. In *Proc. of IEEE INFOCOM–2011* (april 2011), pp. 1862 –1870.

[65] Zeng, K., Govindan, K., and Mohapatra, P. Non-cryptographic authentication and identification in wireless networks. *Wireless Commun. 17* (October 2010), 56–62.

Appendix A

Proposition. *The PHY-layer integrity verification of D by mechanism in Section 4.2 is δ–secure, where*

$$\delta = \left(1 - \frac{1-p_I}{4}\right)^{|\mathbf{s}|}. \quad (4)$$

Here δ is the probability that the BS accepts a message forgery by A, $|\mathbf{s}|$ is the length of the vector indicating the number of the helper's ON slots, and p_I is the probability of inferring the helper's activity during one MC ON-OFF bit when D and H do not co-transmit. Here, δ is a negligible function of $|\mathbf{s}|$. In eq. (4)*, it is assumed that a strongly universal hash function is used as part of the HELP primitive.*

Proof. Assume that the adversary A wants to modify the message m_D sent from D to the BS to a message $m'_D \neq m_D$. To accept m'_D, the BS must correctly receive $[h(m'_D)], m'_D$ and all the slots indicated in $\mathbf{s}$ must be ON slots. The modification of m_D to m'_D can be made by canceling m_D and injecting m'_D. However, to pass verification, A has to modify $[h(m_D)]$ to $[h(m'_D)]$. Since, m_D is unknown to the adversary while $[h(m_D)]$ is being transmitted due to the one-wayness of $h(\cdot)$, A cannot predict the signal transmitted from D.

To modify $[h(m_D)]$, the adversary must launch a signal cancellation on $[h(m_D)] + m_H$ and inject $[h(m'_D)]$ at the same time. Moreover, all the ON slots denoted in the helper's location vector $\mathbf{s}$ must remain as ON slots in $[h(m'_D)]$. Also, the *BS* must decode $[h(m'_D)]$ after m_H is removed. This can be achieved if A does not apply any cancellation on the ON slots indicated in $\mathbf{s}$ and modifies the rest of the slots (OFF slots in m_H) to decode to the desired message. The signal injections of A are made according to Table 1.

The derivation of the probability δ that the adversary's modification is accepted at the *BS* is performed in two parts. In the first part, we derive the probability that A's cancellation/injection is detected, when A modifies the transmission one bit. We then compute the probability of detecting signal modifications by A over all bits. Consider the i^{th} bit of $h(m'_D)$ which corresponds to Manchester-coded slots t_{2i-1} and t_{2i}.

Here, we assume a probability p_I, which is the probability of inference of detecting the presence of H's signal. This is discussed in details in the Section 6. Here we state an assumption, that if H's signal is detected the adversary does not cancel the signal. The probability of cancel is $(1 - p_I)$.

The adversary is detected for i^{th} bit on which H is active, for two conditions with wrong inference $(1 - p_I)$. (a) First, the helper bit is zero *i.e.* H injects energy on t_{2i} slot, device bit is one slot and adversary bit is one. (b) Second, the helper bit is one *i.e.* H injects energy on the t_{2i-1} slot, device bit is zero and the adversary bit is zero.

Let P_r denote the probability that the *BS* rejects the corresponding bit of $[h(m'_D)]$ at bit b_i due to cases (a) and (b). This probability can be calculated as:

$$\begin{aligned} p_r &= (\Pr[b_i^H = 0, b_i^D = 1, b_i^A = 1] \\ &\quad + \Pr[b_i^H = 1, b_i^D = 0, b_i^A = 0]) \\ &\qquad \Pr[\text{wrong inference}] \\ &= \left(\frac{1}{2}\cdot\frac{1}{2}\cdot\frac{1}{2} + \frac{1}{2}\cdot\frac{1}{2}\cdot\frac{1}{2}\right)(1 - p_I) \\ &= \frac{1 - p_I}{4}, \qquad (5) \end{aligned}$$

In (5), b_i^X denotes the transmitted value of device X at bit b_i, and p_I is the probability of inference of helper's activity by the A on a given bit. For (5), we have used the fact that a strictly universal hash function is the part of HELP. For a strictly universal hash function, output hashes for two different inputs differ on each bit with probability 1/2.

The probability δ of accepting the modified message of A at the *BS* is computed by taking into account all $|\mathbf{s}|$ cardinality of the set of bits on which the helper was active. The adversary's modified message is accepted by the *BS* if *none of the bits* in $|\mathbf{s}|$ is rejected. Each bit b_i is rejected with probability p_r given by (5). As rejection on each slot occurs independently, the overall probability of accepting $[h(m'_D)]$ is computed via the Binomial distribution with parameter p_r. That is,

$$\begin{aligned} \delta &= 1 - \sum_{x=1}^{|\mathbf{s}|} B(x, |\mathbf{s}|, p_r) \\ &= 1 - \sum_{x=0}^{|\mathbf{s}|} B(x, |\mathbf{s}|, p_r) + B(0, |\mathbf{s}|, p_r) \\ &= (1 - p_r)^{|\mathbf{s}|} \\ &= (1 - \frac{1 - p_I}{4})^{|\mathbf{s}|}. \qquad (6) \end{aligned}$$

where $B(\alpha, \beta, \gamma)$ is the Binomial probability density function.

We now show that δ is a negligible function of $|\mathbf{s}|$.

In (6), δ is a negligible function if $(1 - p_r)^{|\mathbf{s}|}$ is shown to be a negligible function. To prove the latter, let $\mu(|\mathbf{s}|) = a^{-|\mathbf{s}|}$ where $a = \frac{1}{1-p_r}$. For $\mu(|\mathbf{s}|)$ to be a negligible function, $\forall\ c \in \mathbb{N}$ there exists a $n_0 \in \mathbb{N}$ such that $|\mathbf{s}| > n_0$ and $\mu(|\mathbf{s}|) < n^{-c}$. Let $n_0 = c^{\frac{a}{a-1}}$. Then

$$\begin{aligned} a^{|\mathbf{s}|} &= (a^{\log_a |\mathbf{s}|})^{-\frac{|\mathbf{s}|}{\log_a |\mathbf{s}|}} \\ &= (|\mathbf{s}|)^{-\frac{|\mathbf{s}|}{\log_a |\mathbf{s}|}}, \end{aligned}$$

Since $|\mathbf{s}| > n_0$, it follows that

$$\frac{|\mathbf{s}|}{\log_a |\mathbf{s}|} > \frac{n_0}{\log_a n_0} > \frac{n_0}{n_0^{\frac{1}{a}}} > c.$$

Therefore,

$$\begin{aligned} \mu(|\mathbf{s}|) &= a^{-|\mathbf{s}|} \\ &= (|\mathbf{s}|)^{-\frac{|\mathbf{s}|}{\log_a |\mathbf{s}|}} \\ &< n^{-c}. \end{aligned}$$

This proves that $(1 - p_r)^{|\mathbf{s}|}$ is a negligible function for $a \neq 1$ or equivalently $p_r \neq 0$, thus concluding the proof on the negligibility of δ for $p_r \neq 0$. □

Appendix B

Proposition. *A legitimate device D pairs with a rogue BS with probability* $\delta + \varepsilon$, *where*

$$\delta = (p'_I)^{|\mathbf{s}'|}, \qquad (7)$$

and ε is a negligible function of the hash length. Here $|\mathbf{s}'| < |\mathbf{s}|$ corresponds to the number of helper's ON slots only during the transmission of m_D in the $[h(m_D), m_D]$, p_I' is the probability of inferring the helper's activity during one MC ON-OFF bit when D and H do not co-transmit, and δ is a negligible function of $|\mathbf{s}'|$ when $p_I' < 1$.

Proof. Assume that the adversary A wants to decode the m_D which contains the key public parameter z_D from $[h(m_D), m_D] + m_H$ without the knowledge of set $\mathbf{s}$.

For $[h(m_D), m_D]$ a bit zero corresponds to (OFF, ON) whereas a bit one corresponds to (ON, OFF). With superimposing H's signal, the BS will also receive slots combinations of (ON, ON). The adversary can extract some information of m_D from the (OFF, ON) and (ON, OFF) slots in the $[h(m_D), m_D] + m_H$. But to extract the information from (ON, ON) slots without the knowledge of $\mathbf{s}$. The adversary has to make intelligent guesses for received (ON, ON) slots, which is parameterized as the probability of inferring the helper's activity by A.

Let p_I' be the inference probability for detecting the presence of H's signal. This is discussed in details in Section 6. Note that, if H's signal is wrongly inferred (with probability $(1 - p_I')$), A maps the received bit on which H was active to a wrong outcome.

The adversary makes wrong mapping when it receives (ON, ON) slots on received $[h(m_D), m_D] + m_H$. It happens when A cannot detect the presence of the helper's signal on the slot where D has injected no energy.

$$p_r = \Pr[\text{wrong inference}] = (1 - p_I'). \tag{8}$$

In (8), p_I' is the probability that A detects the H's signal correctly on a particular bit.

The probability δ of extracting correct m_D from received signal $[h(m_D), m_D] + m_H$ by A. The adversary can decode correct m_D if *none of the bits* are decoded wrong. Each bit is wrongly mapped with probability p_r, given by (8). As rejection on each slot occurs independently, the overall probability of correctly decoding m_D from $[h(m_D), m_D] + m_H$ is computed via the Binomial distribution with parameter p_r. That is,

$$\begin{aligned}
\delta &= 1 - \sum_{x=1}^{|\mathbf{s}'|} B\left(x, |\mathbf{s}'|, p_r\right) \\
&= 1 - \sum_{x=0}^{|\mathbf{s}'|} B\left(x, |\mathbf{s}'|, p_r\right) + B\left(0, |\mathbf{s}'| p_r\right) \\
&= (1 - p_r)^{|\mathbf{s}'|} \\
&= \left(1 - (1 - p_I')\right)^{|\mathbf{s}'|} \\
&= \left(p_I'\right)^{|\mathbf{s}'|}.
\end{aligned} \tag{9}$$

where $B(\alpha, \beta, \gamma)$ is the Binomial probability density function and $|\mathbf{s}'| \subset |\mathbf{s}|$, which corresponds to the number of helper's ON signals only during the transmission of m_D in the $[h(m_D), m_D]$.

We now show that δ is a negligible function of $|\mathbf{s}'|$.

In (9), δ is a negligible function if $(1 - p_r)^{|\mathbf{s}'|}$ is shown to be a negligible function. To prove the latter, let $\mu(|\mathbf{s}'|) = a^{-|\mathbf{s}'|}$ where $a = \frac{1}{1-p_r}$. For $\mu(|\mathbf{s}'|)$ to be a negligible function, $\forall\ c \in \mathbb{N}$ there exists a $n_0 \in \mathbb{N}$ such that $|\mathbf{s}'| > n_0$ and $\mu(|\mathbf{s}'|) < n^{-c}$. Let $n_0 = c^{\frac{a}{a-1}}$. Then

$$\begin{aligned}
a^{|\mathbf{s}'|} &= \left(a^{\log_a |\mathbf{s}'|}\right)^{-\frac{|\mathbf{s}'|}{\log_a |\mathbf{s}'|}} \\
&= \left(|\mathbf{s}'|\right)^{-\frac{|\mathbf{s}'|}{\log_a |\mathbf{s}'|}},
\end{aligned}$$

Since $|\mathbf{s}'| > n_0$, it follows that

$$\begin{aligned}
\frac{|\mathbf{s}'|}{\log_a |\mathbf{s}'|} &> \frac{n_0}{\log_a n_0} \\
&> \frac{n_0}{n_0^{\frac{1}{a}}} \\
&> c.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\mu(|\mathbf{s}'|) &= a^{-|\mathbf{s}'|} \\
&= \left(|\mathbf{s}'|\right)^{-\frac{|\mathbf{s}'|}{\log_a |\mathbf{s}'|}} \\
&< n^{-c}.
\end{aligned}$$

This proves that $(1 - p_r)^{|\mathbf{s}'|}$ is a negligible function for $a \neq 1$ or equivalently $p_r \neq 0$.

After the attacker extracts m_D, the rogue BS needs to pass the challenge-response authentication in the key confirmation phase. Assuming the use of a strongly universal hash function to compute the response $h_{k_{D,BS'}}(ID_{BS}||C_D||0)$, he can only pass this authentication if he has the correct key $k_{D,BS'}$. Otherwise, his successful probability ε is negligible. But he can only obtain the correct key by extracting the correct m_D value. Therefore, the success probability of the rogue BS to pair with the device is upper bounded by $\delta + \varepsilon$, where ε is a negligible function (of the length of the hash function). Since δ is a negligible function of $|\mathbf{s}'|$ which can be the same as the message length (and here the m_D is a DH public number, whose bit length is typically larger or equal to the hash length), the overall probability is a negligible function. This concludes the proof. □