Swipe2Pair: Secure and Fast In-Band Wireless Device Pairing

Yaqi He
yhe15@gmu.edu
Wireless Cyber Center
George Mason University
Fairfax, Virginia, USA

Kai Zeng
kzeng2@gmu.edu
Wireless Cyber Center
George Mason University
Fairfax, Virginia, USA

Long Jiao
ljiao@umassd.edu
Department of Computer Information Science
University of Massachusetts Dartmouth
Dartmouth, Massachusetts, USA

Brian L. Mark
bmark@gmu.edu
Wireless Cyber Center
George Mason University
Fairfax, Virginia, USA

Khaled N. Khasawneh
kkhasawn@gmu.edu
Electrical and Computer Engineering Department
George Mason University
Fairfax, Virginia, USA

ABSTRACT

Wireless device pairing is a critical security mechanism to bootstrap the secure communication between two devices without a pre-shared secret. It has been widely used in many Internet of Things (IoT) applications, such as smart-home and smart-health. Most existing device pairing mechanisms are based on out-of-band channels, e.g., extra sensors or hardware, to validate the proximity of pairing devices. However, out-of-band channels are not universal across all wireless devices, so such a scheme is limited to certain application scenarios or conditions. On the other hand, in-band channel-based device pairing seeks universal applicability by only relying on wireless interfaces. Existing in-band channel-based pairing schemes either require multiple antennas separated by a good distance on one pairing device, which is not feasible in certain scenarios, or require users to repeat multiple sweeps, which is not optimal in terms of usability.

Therefore, an in-band wireless device pairing scheme providing high security while maintaining high usability (simple pairing process and minimal user intervention) is highly desired. In this work, we propose an easy-to-use mutual authentication device pairing scheme, named Swipe2Pair, based on the proximity of pairing devices and randomization of wireless transmission power. We conduct extensive security analysis and collect considerable experimental data under various settings across different environments. Experimental results show that Swipe2Pair achieves high security and usability. It only takes less than one second to complete the pairing process with a simple swipe of one device in front of the other.

CCS CONCEPTS

• **Security and privacy → Mobile and wireless security**.

WiSec '24, May 27–30, 2024, Seoul, Republic of Korea

ACM ISBN 979-8-4007-0582-3/24/05.
https://doi.org/10.1145/3643833.3656127

KEYWORDS

Security Protocol, Authentication, In-band Device Pairing, Wireless Communication System

ACM Reference Format:
Yaqi He, Kai Zeng, Long Jiao, Brian L. Mark, and Khaled N. Khasawneh. 2024. Swipe2Pair: Secure and Fast In-Band Wireless Device Pairing . In *Proceedings of the 17th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '24), May 27–30, 2024, Seoul, Republic of Korea.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3643833.3656127

1 INTRODUCTION

The widespread adoption of Internet of Things (IoT) technology across multiple fields has ushered in a new era of societal transformation, from commercial sectors to our daily lives. IoT systems generate, transmit, and process a vast amount of data, much of which includes sensitive/personal information. Secure IoT device communication and networking are of utmost importance. Inadequate security measures may lead to a range of serious consequences including unauthorized access, privacy breaches, and even threats to life [34]. Among various security mechanisms/protocols to safeguard IoT device communication, wireless device pairing has been widely adopted to bootstrap the secure communication between the IoT device and the user device (e.g., smartphone) or access network without a pre-shared secret.

Existing device pairing protocols can be generally classified into two categories: out-of-band (OOB) channel-based and in-band channel-based. OOB channel-based schemes usually rely on auxiliary interfaces and hardware (sensors) to prove/validate the location or context proximity of pairing devices. For example, the passkey-based schemes require the auxiliary I/O hardware. QR-code-based schemes require cameras for scanning. OOB pairing cannot run on IoT devices without user interfaces and auxiliary hardware/sensors. As an alternative, in-band pairing protocols have become popular due to the potential to rely only on wireless interfaces. However, the existing in-band based device pairing schemes either require well-separated multiple antennas on one pairing device (i.e., two antennas at two ends on top of the screen of a laptop in the Good-Neighbor scheme [4]), which is not always available especially for small-size wireless devices, or require users to repeat multiple

sweeps (i.e., SFIRE in [7]), which is not optimal in terms of usability. Furthermore, GoodNeighbor only provides one-way authentication.

To the best of our knowledge, in-band device pairing schemes that provide mutual authentication with high usability are missing from the literature. To fill this gap, we propose Swipe2Pair, a novel in-band device pairing scheme based on the location proximity of pairing devices and randomized wireless transmission power. The advantages of Swipe2Pair are multifold: a) It does not need auxiliary OO*B* channels or sensors. Swipe2Pair can run on IoT devices with standard wireless interfaces (e.g., WiFi, Bluetooth, Zigbee, etc.), which makes it a viable solution for most IoT devices. b) It does not involve tedious user interactions or sequential actions and only requires a simple swiping action. Swipe2Pair thus has a short pairing delay (i.e., less than 1 second) and provides better usability than existing schemes. c) It not only supports mutual authentication but can also defend against powerful attackers with capabilities to accurately infer locations/motions of pairing devices. These advantages make Swipe2Pair a robust and usable device pairing solution. The main contributions of this work are summarized as follows:

- We propose a new in-band device pairing scheme, Swipe2Pair, which is a universal solution that only utilizes available wireless interfaces on pairing devices.
- We adopt transmission power randomization and exploit intrinsic channel fading properties (i.e., a remote attacker experiences higher fading channel variations than proximate pairing devices) to defend against the advanced attackers who may infer the precise distance/location and motion information of pairing devices.
- We conduct a thorough security analysis to assess the security strength of Swipe2Pair under different types of attackers, ranging from general attackers to supreme attackers who can accurately infer the location/motion information of pairing devices.
- We conduct extensive experiments to validate the security and usability of Swipe2Pair in various environments, demonstrating the robustness of Swipe2Pair.

The remainder of the paper is organized as follows. In Section II, we review the state-of-the-art techniques in secure device pairing. In Section III, we describe our system model and attack model. In Section IV, we propose the Swipe2Pair protocol. Section V analyzes the security strength of Swipe2Pair under general, advanced, and supreme attacks. Section VI presents the experimental results under different environments. Concluding remarks are given in Section VII.

2 RELATED WORK

Wireless device pairing schemes can be generally classified into two categories: OOB channel-based [10, 12, 15, 16, 29, 33, 37] and in-band channel-based [4, 7].

OOB channel-based schemes usually rely on auxiliary hardware or sensors to verify the location or context proximity of the pairing devices. Location-proximity based schemes in the OOB channel classify two different categories depending on the physical accessibility of the IoT device. The inaccessible IoT schemes include Move2Auth [37], which leverages the significant RSSI variation with the inertial accelerometer or gyroscope of the user's smartphone in proximity. Similar scenarios are commonly seen

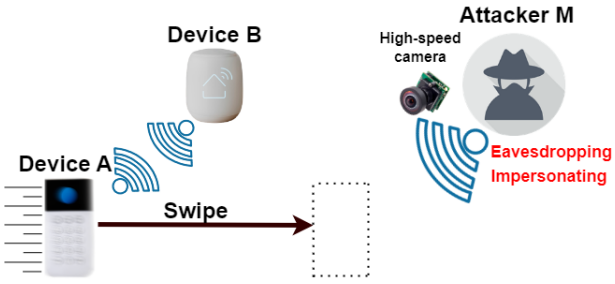


Figure 1: System model comprising proximate pairing devices *A* and *B*, and a remote attacker *M* utilizing a high-speed camera to capture the precise instant motion of device *A* as it is swiped by the user.

in healthcare involving Implantable Medical Devices (IMDs). In these scenarios, extra sensors such as ultrasound transducer [27] or electrocardiogram heartbeat reader [24] are implemented through the touch-to-access principle to communicate and authenticate the inaccessible IMD inside patients' bodies.

The accessible IoT schemes include T2Pair [15], which involves engaging the button on the IoT device and the smartphone or wristband worn by the user to produce the random pause intervals for authentication. The scheme in [12] involves a closely attached magnetometer, while the scheme in [16] involves shaking the accelerometers embedded in two smartphones together as the shared secret. Other innovative schemes in this category authenticate pairing devices through physiological values derived from daily activities of users, transmitted over the wireless body area network (WBAN) [21, 23, 32]. These methods offer the advantage of not requiring deliberate pairing gestures, such as pressing, pushing, or rotating.

Additionally, OOB context-proximity schemes serve as distinctive approaches to verify physical proximity, co-location, or shared environmental characteristics of devices within a physically secure boundary that is invisible to outside attackers. These encompass Perceptio [10] and NAN [25], which cooperate with the adjacent authenticated devices to validate a new IoT device through their diverse sensors. Moreover, the audio-based scheme [18] captures a common audio context through microphones to verify nearby devices. Furthermore, Eolo [11] leverages barometric pressure to facilitate key verification for proximate devices.

As we can see, OOB channel-based schemes require extra hardware or sensors, which may not be available on all IoT devices. Meanwhile, heavily relying on the extra hardware and sensors deviates from the purpose of the universal pairing solution. To achieve better applicability, in-band channel-based solutions are proposed to utilize only wireless interfaces to verify the location proximity of pairing devices.

The in-band proximity-based approaches [13] have gained prominence by capitalizing on alternative channels to establish the authenticity of the users and thwart impersonation by Man-in-the-Middle (MitM) attackers. These approaches include Good Neighbour [4], which uses multiple antennas and received signal strength ratio to filter out attackers. However, multiple antennas and an appropriate distance between antennas are not always achievable in all device pairing scenarios.

Pairing Approaches	In-band channel	Mutual authen	Sensors/Hardware requirements	User intervention	Delay
Shake well before use [16]	×	✓	Accelerometer	hold two devices together and shake	5s
WAVE [29]	×	✓	button and LED	wave user arm	4-6s
Move2Auth [37]	×	×	Accelerometer, Gyroscope	Move smartphone back and forth and rotate	3s
Perceptio [10]	×	✓	Environmental sensors	Null (Auto pairing)	>10s
T2pair [15]	×	✓	Button and smartwatch	Press button multiple times with random pause	7s
Good Neighbour [4]	✓	×	At least two antennas on one device	Hold smartphone and move it from one receive antenna to the other	11.6s
SFIRE [7]	✓	✓	Helper: Smartphone	Sweep helper at least 4 times (2s/sweep)	>8s
Swipe2Pair (this work)	✓	✓	No sensor, extra helper, or multiple antennas required	Quick swipe once	<1s

Table 1: Security and Usability Comparison between Representative Device Pairing Schemes

Other approaches in this category may also exploit in-band channels, exemplified by [7], which relies on the exploitation of physical signal propagation laws or [6], which leverages a helper to detect message manipulation and signal cancellation to resist MitM attacks. However, not every IoT device is accessible or can be authenticated with the assistance of a trustworthy service verifier. Though innovative, the particular requirements of the above device pairing approaches are not universally adaptable to all pairing contexts.

Besides, usability is another critical but ignored aspect of secure device pairing. The OOB channel-based schemes usually leverage rapid motion, pressing, and rotation multiple times to generate significant RSS variation, which harms simple usage. It seems that an in-band scheme with friendly usability and strong security has not been achieved yet. We jointly address these issues in Table 1 and then propose our solution.

3 SYSTEM AND ATTACKING MODELS

3.1 System Model

We assume a general device pairing system model illustrated in Fig. 1, which includes two legitimate pairing devices (A and B) in close proximity and a remote attacker (M). At least one of the pairing devices is movable, and we denote it as A, which could be a smartphone or an IoT device. The other pairing device B can either be movable or stationary, such as another IoT device or a wireless router/access point (AP). Both devices have a wireless interface (e.g., WiFi, Bluetooth, or Zigbee, etc.), allowing them to communicate through the wireless channel. They can perform lightweight computation for encryption and decryption.

The goal of device A and device B is to bootstrap a secure channel, i.e., establish a shared secret, through wireless communication without a pre-shared secret or the help of a trusted third party. One application example is an IoT device entering a home WiFi network for the first time. However, the IoT device does not have any input interface, such as a keyboard or touch screen, for the user to type the password. Then Device A is the IoT device and Device B is the wireless router under this setting. Another application example is pairing two IoT devices, such as a smart pen (Device A) with a smart notebook or smart board (Device B).

3.2 Attacking Model

We consider both passive and active attackers with different attacking capabilities, ranging from general attacks to supreme attacks.

3.2.1 ***Passive attacks.*** Eavesdropping is a typical type of passive attack in which the attacker does not interrupt the device pairing but passively monitors the wireless communications between pairing devices, aiming to steal sensitive information/credentials such as the WiFi password or the secret key established for secure communication. We assume the attacker can not only overhear all the messages exchanged between the pairing devices but also measure the wireless channel characteristics, such as channel state information or received signal strength (RSS) of the eavesdropping channel.

3.2.2 ***Active attacks.*** The man-in-the-middle (MitM) attack is considered the most insidious and deceptive active attack against device pairing. The MitM attacker can impersonate both devices by changing its identity to the victims or running an ARP (address resolution protocol) attack. In this work, we assume the active attacker can intercept the wireless communication channel of pairing devices and disrupt/overshadow legitimate signals by injecting arbitrary signals with arbitrary power.

We assume the active attacker can also observe/predict the location/motion of the pairing devices. This could be achieved by mounting a high-precision motion camera on the attacking device connected to a powerful computer that can perform real-time computing. We assume the attacker knows the device pairing protocol. The attacker cannot be as proximate as the two pairing devices (e.g, 10 cm) to the pairing devices; otherwise, it could be physically detected. The channel characteristics expose the remote location depending on the various circumstances. We will elaborate on this in section 6.

We consider three types of MitM attackers based on their attacking capabilities: **1) General attacker** who may not know the location or motion of the pairing devices and just follows the pairing protocol. **2) Advanced attacker** who can estimate/infer the location/motion of the pairing devices but with a certain degree of

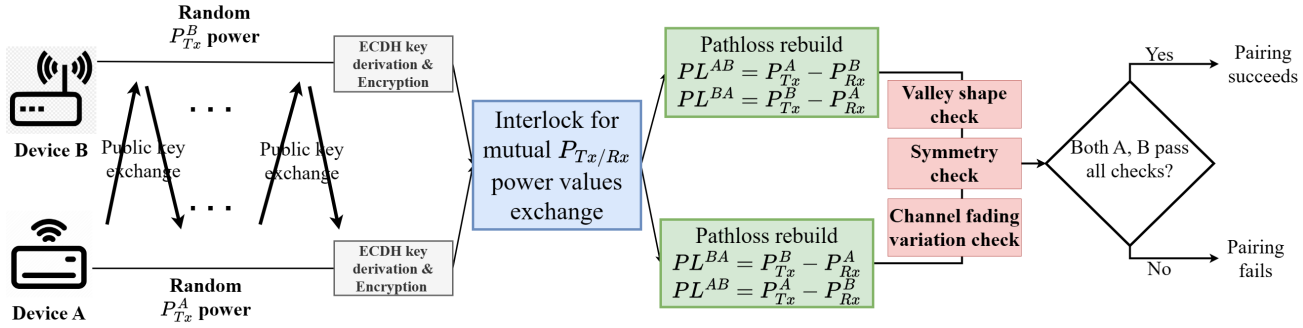


Figure 2: Workflow of Swipe2Pair

estimation error. **3) Supreme attacker** who is assumed to know or precisely estimate the location/motion of the pairing devices. We will elaborate on what these different attackers could do and how Swipe2Pair defends against these attacks in the following sections.

4 PROTOCOL DESIGN

In this section, we elaborate on our Swipe2Pair protocol in detail, as shown in Fig. 2. The protocol contains four stages: 1) Public key exchange with random transmission (Tx) power and receiving (Rx) power recording; 2) Interlock protocol to exchange encrypted Tx and Rx power information; 3) Computing the bidirectional pathloss between A and B; and 4) Authentication or proximity validation through valley shape check, channel symmetry check, and channel fading variation check.

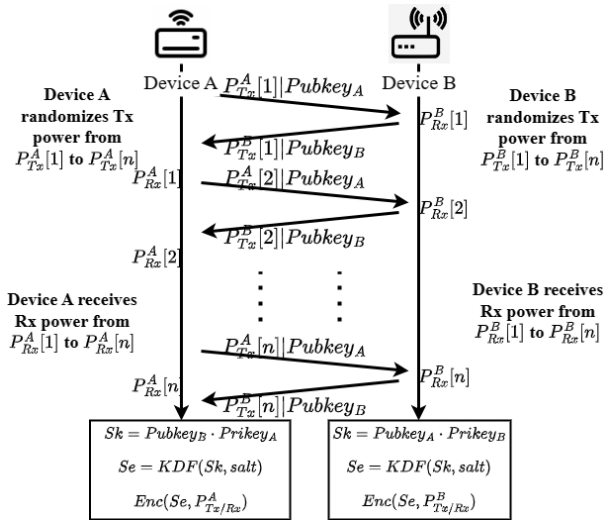


Figure 3: Public key exchange with randomized Tx power and ECDH key derivation

In this stage, the two pairing devices A and B run the Elliptic Curve Diffie Hellman (ECDH) key exchange protocol by repeatedly exchanging their public keys with random Tx power. ECDH is tailored to leverage the cryptographic properties of elliptic curves [28] to achieve computational efficiency [14] and reduced key size [36] while maintaining a comparable level of security. These attributes render ECDH particularly well-suited for resource-constrained platforms. The two pairing devices A and B follow the following steps:

(1) Device A initializes multiple identical packet Tx, with each packet i ($1 \le i \le n$) containing A's public key ($Pubkey_A$), transmitted at a randomized Tx power $P^A_{Tx}[i]$. At the same time, the user simply swipes Device A from the left side to the right side of Device B to accomplish the pairing process. In this process, Device A records its random Tx power $P^A_{Tx}[i]$.
(2) On receiving packet i from Device A, Device B immediately sends back a reply packet containing B's public key ($Pubkey_B$) with a randomized Tx power $P^B_{Tx}[i]$. At the same time, B records its random Tx power $P^B_{Tx}[i]$ and Rx power $P^B_{Rx}[i]$ of the received packet from A.
(3) When A receives the reply packet i from B, it records its corresponding Rx power $P^A_{Rx}[i]$.
(4) A and B repeat the above bidirectional public key exchange and channel probing multiple times with a different random Tx power each time until all the n packets have been exchanged. These can be done during the user's fast swipe within 1 second.
(5) Both A and B concatenate their recorded Tx power series and Rx power series together, denoted as $P^A_{Tx/Rx} = [P^A_{Tx}[1], ..., P^A_{Tx}[n]|P^A_{Rx}[1], ..., P^A_{Rx}[n]]$ and $P^B_{Tx/Rx} = [P^B_{Tx}[1], ..., P^B_{Tx}[n]|P^B_{Rx}[1], ..., P^B_{Rx}[n]]$, respectively.
(6) A and B derive the shared secret key independently at each side after the user's swiping by multiplying their private key with the exchanged public key received from the other side. Specifically, A derives the shared secret key through $S_k = Pubkey_B \cdot Prikey_A$ and B derives the shared secret key through $S_k = Pubkey_A \cdot Prikey_B$. Note that the multiplication operation is defined on the elliptic curve over a finite field. After that, an efficient key derivation function (KDF) [17] is applied to derive the session key S_e from the shared secret key to secure the power information exchange.
(7) A and B encrypt their power series information $P^A_{Tx/Rx}$ and $P^B_{Tx/Rx}$, respectively, with the shared session key S_e and exchange the ciphertexts using the interlock protocol described in the following subsection.

4.1 Interlock Protocol

The ciphertext is encrypted with the session key using the AES-ECB block cipher [31]. Each cipher block is 128 bits. The advantage of the interlock protocol [2] relies on the characteristic that half of an encrypted message cannot be decrypted even with the decryption key. It divides each 128-bit cipher block into two halves and consecutively exchanges the two halves of the cipher blocks between pairing devices. Notice the second half of the cipher block exchange will not start until the first half exchange has been completed from both sides.

The interlock protocol [22] ensures non-repudiation for both sides, which prevents M from intentionally falsifying the Tx and Rx power information to pass the authentication checks (to be described in Section 4.3). Without the interlock protocols, M can impersonate A or B to communicate with the other pairing device and infer the pathloss information to pass all the authentication checks.

4.2 Pathloss Computing

After the Tx and Rx power information exchange, both devices A and B can compute the bidirectional channel pathloss for each channel probing i as follows, assuming a lognormal shadowing pathloss model and considering a measurement error:

$$PL^{AB}[i] = P^{\mathrm{A}}_{Tx}[i] - P^{\mathrm{B}}_{Rx}[i] = 10\alpha \log_{10}\left(\frac{4\pi\ d_{\mathrm{AB}_i}}{\lambda}\right) + \chi_{AB_i} + e_{AB_i}$$
$$PL^{BA}[i] = P^{\mathrm{B}}_{Tx}[i] - P^{\mathrm{A}}_{Rx}[i] = 10\alpha \log_{10}\left(\frac{4\pi\ d_{\mathrm{BA}_i}}{\lambda}\right) + \chi_{BA_i} + e_{BA_i} \quad (1)$$

where the wavelength $\lambda = c/f = (3 \times 10^8/2.4 \times 10^9)$ and α is the pathloss exponent and is usually approximated by 2 for line-of-sight and 2.8 for non-line-of-sight scenarios [30]. χ_{AB_i} and χ_{BA_i} represent channel fading, which follow a Gaussian distribution $\mathcal{N}(0, \sigma^2_{AB})$. Assuming that channel reciprocity holds, the instance of channel fading $\chi_{AB_i} = \chi_{BA_i}$ and $d_{AB_i} = d_{BA_i}$. While e_{AB_i} and e_{BA_i} are measurement errors modeled as independently distributed Gaussian variables following $\mathcal{N}(0, \sigma^2_B)$ and $\mathcal{N}(0, \sigma^2_A)$, respectively. If the measurement errors are small, $PL^{AB}[i] \approx PL^{BA}[i]$.

4.3 Authentication

According to Fig. 2, after both devices A and B reconstruct their bidirectional pathloss, they can start to authenticate whether the other side is in proximity through three checks: valley shape check, symmetry check, and channel fading variation check.

4.3.1 ***Valley shape check.*** The valley shape in the rebuild pathloss reflects the user's swiping motion. The pathloss experiences a decrease and then an increase as A is swiped from the left to right of the stationary B. Note that according to the lognormal shadowing model, the pathloss contains two parts: a deterministic part, which is a function of the distance between the two devices and a random channel fading part. That is, the valley shape is not smooth but experiences noise, which may cause small local valleys. To prevent the oscillating noisy data from being identified as the real valley from the user's swipe, we leverage a robust peak-valley detection algorithm refer in [3] based on the principle of discrepancy. This algorithm constructs a separate moving mean and standard deviation to identify the valley from previous signals without corrupting the signal threshold for further signal data.

If the valley shape has been recognized, we leverage the approach in [20] to find the start and end points. This approach fits a model to data by iterative curve fitting and then uses the best-fit model to locate the start and end points by interpolation. It locates the start and end points through the cutoff parameter which is a fraction of the peak or valley height (usually 1%). For more complex shapes with multiple peaks or valleys, this approach can combine multiple simple models (e.g., Gaussian) to fit them. Once the start and end points have been identified, the pathloss data range for the valley shape can be extracted for the subsequent symmetry and channel fading variation checks.

4.3.2 ***Symmetry check.*** It is imperative to ensure the symmetry of the valley shape derived from both pathlosses, as this symmetry aids in filtering potential MitM attacks. From the modeling and analysis in Section 4.2, we know that the difference between each computed point of the two pathlosses theoretically should be small if the measurement error is small, so we employ the mean squared error (MSE)

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(PL^{\mathrm{AB}}[i] - PL^{\mathrm{BA}}[i])^2 \quad (2)$$

to assess discrepancies between each set of instantaneous pathloss data derived from the two pairing devices. This assessment is then compared with an empirical threshold to determine the symmetry of the valley shape from two pathlosses.

4.3.3 ***Channel fading variation check.*** The last check is the channel fading variation check, which is based on the observation that the channel fading variation experienced by a remote attacker is usually larger than that between the close-by legitimate pairing devices. As we will show in Section 5.2.3, a supreme attacker could pass the first two checks with the knowledge or accurate estimation of the locations/motions of Devices A and B, but will still fail this check.

The channel fading is sensitive to the change of multipath components. For example, a transmitted signal propagates to the receiver through multiple paths. Each path contributes to a differently delayed, attenuated, and phase-shifted signal. The slight change in certain multipath components may add up to a significant constructive or destructive relative phase of the delayed signals, which leads to considerable fluctuations in the receiving power. The multipath-rich indoor environment also leads the receiving power to fluctuate on the order of signal wavelength and contributes to large shadowing variance [35].

It has been observed from several empirical studies that the longer the distance between the transmitter and receiver, the larger the channel fading variation [1, 26, 38]. Consequently, the remote M should experience more pronounced fluctuations caused by both multipath fading from the indoor environment and shadowing effects from environmental obstacles than the close-by pairing devices A and B. We leverage this phenomenon [5, 8, 19] to filter the supreme attacker M who could pass the valley shape check and symmetry check.

In summary, the Swipe2Pair protocol authentication succeeds only if both pairing devices pass all three checks in the authentication. Otherwise, the Swipe2Pair process aborts if any check from any device fails and the two pairing devices will be reset. We will analyze the security performance of Swipe2Pair in more detail in Sections 5 and 6.

5 SECURITY ANALYSIS

In this section, we analyze the security strength of Swipe2Pair against general, advanced, and supreme attacks, respectively.

5.1 Attacking Strategies

Before we start the security analysis, let us examine at what stage and how a MitM attacker could launch the attack. We assume the MitM attacker could compromise the communication between Devices A and B in the very beginning of the pairing process. That is, from A and B's perspective, they are each running the pairing protocol with M.

Although M has no control of or knows A or B's randomized Tx power, he can fully control his Tx power and can falsify both his Tx power and Rx power information during the tx/rx power exchange stage. Note that M can generate a unique shared key with A (Sk_{AM}) and B (Sk_{BM}), and therefore encrypt the corresponding falsified Tx/Rx power information respectively. A and B will decrypt this information and use the information to reconstruct the pathloss and run the three checks described in the previous section. The valley shape in the pathloss reconstruction varies a lot depending on the user's swipe technique. This may somehow affect the valley shape check as a usability problem. We will elaborate on its robustness in section 6.

In the following analysis, we will show that a general attacker is unlikely to pass either the valley shape check or channel fading variation check, but could pass the symmetry check. The advanced and supreme attackers could pass both the valley shape and symmetry checks, but not the channel fading variation check.

5.2 Security Analysis Against Attackers with Different Capabilities

A critical factor that significantly affects the security performance is M's capability to obtain accurate instantaneous estimates of the distance d_{AB_i} while A is moving during the pairing process. Meanwhile, the pathloss exponent α is considered known to M since M is in a similar environment with A and B. We categorize M into three types based on their capability, as discussed in Section 3: General M, Advanced M, and Supreme M.

5.2.1 ***Security Analysis Against General Attacker.*** Firstly, General M does not use any methods to estimate/infer A/B's location information and just follows the pairing protocol without falsifying Tx/Rx information. So the rebuilt pathlosses at A and B are unlikely to pass the valley shape check based on the Tx/Rx information provided by General M.

Pathlosses Rebuilt at *A* and *B* under General Attack: During the pathloss rebuild process, M can impersonate either A or B to communicate with the other. Similar to the modeling in Section 4.2, the eavesdropping/attacking channel between A/B and M is assumed to follow lognormal shadowing, with a channel fading variation χ_{MA}/χ_{MB} following $\mathcal{N}(0, \sigma^2_{MA/MB})$ and measurement error e_{MA}/e_{MB} following $\mathcal{N}(0, \sigma^2_{A/B})$. When M impersonates B, the bidirectional posslosses rebuilt at A are:

$$PL^{AM}[i] = P^{\mathrm{A}}_{Tx}[i] - P^{\mathrm{M}}_{Rx}[i] = 10\alpha \log_{10}\left(\frac{4\pi d_{\mathrm{AM}_i}}{\lambda}\right) + \chi_{AM_i} + e_{AM_i}$$
$$PL^{MA}[i] = P^{\mathrm{M}}_{Tx}[i] - P^{\mathrm{A}}_{Rx}[i] = 10\alpha \log_{10}\left(\frac{4\pi d_{\mathrm{MA}_i}}{\lambda}\right) + \chi_{MA_i} + e_{MA_i}. \tag{3}$$

Similarly, when M impersonates A, the bidirectional pathlosses rebuilt at B are:

$$PL^{BM}[i] = P^{\mathrm{B}}_{Tx}[i] - P^{\mathrm{M}}_{Rx}[i] = 10\alpha \log_{10}\left(\frac{4\pi d_{\mathrm{BM}_i}}{\lambda}\right) + \chi_{BM_i} + e_{BM_i}$$
$$PL^{MB}[i] = P^{\mathrm{M}}_{Tx}[i] - P^{\mathrm{B}}_{Rx}[i] = 10\alpha \log_{10}\left(\frac{4\pi d_{\mathrm{MB}_i}}{\lambda}\right) + \chi_{MB_i} + e_{MB_i}. \tag{4}$$

From Eq. (3), assuming the channel reciprocity holds between A and M, we have $PL^{AM}[i] \approx PL^{MA}[i]$ if we assume the measurement errors are small. Similarly, from Eq. (4), we can obtain $PL^{BM}[i] \approx PL^{MB}[i]$. So the general attacker could pass the symmetry check at both A and B.

However, since M is farther away from A/B compared to the distance between A and B, neither A nor B is likely to observe a deep valley shape. Since A is moving, depending on the relative motion between A and M, A could observe a valley shape, but it is likely to be shallow. As we will show in Section 6 Fig. 4d, in practice, the valley depth corresponding to the legitimate channel between A and B is about 15dB with a peak point around 55dB and the valley point around 40 dB. While for the attacking channels A/B-M, the valley shape is either not observable or is shallow with higher peak and valley points, e.g., around 65dB. So setting the valley peak around 55dB and valley point around 40dB and a valley width of about 0.2s is sufficient to differentiate a close-by pairing device from a remote General M.

From Eq. (3), we can see that both the channel fading variation and measurement error contribute to the observed channel variations. Therefore, after fully removing the deterministic pathloss components corresponding to distance, the channel variations observed at A under attack should follow $\mathcal{N}(0, \sigma^2_{MA} + \sigma^2_A)$. While from Eq. (1), the channel variations observed at A without attack should follow $\mathcal{N}(0, \sigma^2_{BA} + \sigma^2_A)$. As we discussed in Section 4.3.3, a farther away attacker usually experiences larger channel fading than close-by pairing devices. That is, $\sigma_{MA} > \sigma_{BA}$, so $\sigma^2_{MA} + \sigma^2_A > \sigma^2_{BA} + \sigma^2_A$. We will show in Section 6 Fig. 8 that this principle is validated through our experiments as well. By setting up a proper threshold (e.g., around 1.27 observed in our experiments) on the sampled standard deviation of the measured channel variations, we can detect the remote attacker.

5.2.2 ***Security Analysis Against Advanced Attacker.*** We assume Advanced M has enhanced capabilities compared to General M through observing or guessing a less accurate estimate of the instantaneous distances between itself and Device A/B (d_{MA_i/MB_i}) and the distance between A and B (d_{AB_i}) during the pairing process. With the estimate of these distances and the knowledge of the pathloss exponent in the environment, Advanced M can falsify its claimed Tx and Rx power information during the Tx/Rx power

exchange stage to try to make Device *A/B* believe it is in close proximity. That is, Advanced *M* aims to make the pathloss computed at *A/B* similar to the expected one between *A* and *B* by manipulating the claimed Tx and Rx information to *A/B*. In the following analysis, we will use the attack on Device *A* as an example. A similar analysis can be derived for the attack on Device *B*.

Note that Advanced *M* cannot control the channel fading variation, which is an intrinsic property of the fading channel, nor the measurement error. What Advanced *M* could achieve is to largely compensate the difference between the pathlosses in channels $A-B$ and $A-M$ by falsifying the claimed Tx and Rx power information, denoted as P'^{M}_{Tx} and P'^{M}_{Rx}, respectively. Considering the distance estimation error at *M*, we can model the estimated distances $\tilde{d}_{AM_i}$ and $\tilde{d}_{AB_i}$ as

$$\tilde{d}_{AM_i} = d_{AM_i} \cdot e^{r_{AM_i}},\ \tilde{d}_{AB_i} = d_{AB_i} \cdot e^{r_{AB_i}}, \tag{5}$$

where r_{AM_i} and r_{AB_i} are relative estimation errors and can be modeled as Gaussian random variables following $\mathcal{N}(0, \sigma_d^2)$.

In order to make the pathloss computed at *A* similar to that between *A* and *B*, Advanced *M* sends the following falsified Tx/Rx power information ($P'^{M}_{Rx}[i]$ and $P'^{M}_{Tx}[i]$) to *A*:

$$\begin{aligned} P'^{M}_{Rx}[i] &= P^{M}_{Rx}[i] + 10\alpha log_{10}\left(\frac{\tilde{d}_{AM_i}}{\tilde{d}_{AB_i}}\right) \\ P'^{M}_{Tx}[i] &= P^{M}_{Tx}[i] - 10\alpha log_{10}\left(\frac{\tilde{d}_{AM_i}}{\tilde{d}_{AB_i}}\right) \end{aligned} \tag{6}$$

Then the computed pathloss $PL'^{AM}[i]$ at *A* is:

$$\begin{aligned} PL'^{AM}[i] &= P^{\mathrm{A}}_{Tx}[i] - P'^{\mathrm{M}}_{Rx}[i] = P^{\mathrm{A}}_{Tx}[i] - \left[P^{\mathrm{M}}_{Rx}[i] + 10\alpha \log_{10}\left(\frac{\tilde{d}_{AM_i}}{\tilde{d}_{AB_i}}\right)\right] \\ &= 10\alpha \log_{10}\left(\frac{4\pi d_{AB_i}}{\lambda}\right) + \chi_{AM_i} + \frac{10\alpha \cdot (r_{AB_i} - r_{AM_i})}{ln(10)} + e_{AM_i}. \end{aligned} \tag{7}$$

Similarly, we can derive the computed pathloss $PL'^{MA}[i]$ as:

$$\begin{aligned} PL'^{MA}[i] &= P'^{\mathrm{M}}_{Tx}[i] - P^{\mathrm{A}}_{Rx}[i] = \left[P^{\mathrm{M}}_{Tx}[i] - 10\alpha \log_{10}\left(\frac{\tilde{d}_{AM_i}}{\tilde{d}_{AB_i}}\right)\right] - P^{\mathrm{A}}_{Rx}[i] \\ &= 10\alpha \log_{10}\left(\frac{4\pi d_{AB_i}}{\lambda}\right) + \chi_{MA_i} + \frac{10\alpha \cdot (r_{AB_i} - r_{AM_i})}{ln(10)} + e_{MA_i}. \end{aligned} \tag{8}$$

From Eqs. (7) and (8), we can see that the distance-dependent deterministic part, $10\alpha \log_{10}\left(\frac{4\pi d_{AB_i}}{\lambda}\right)$, in the rebuilt pathloss at *A* equals to that between *A* and *B*. Therefore, Advanced *M* could pass the valley shape check.

Furthermore, although the extra variation, $\frac{10\alpha \cdot (r_{AB_i} - r_{AM_i})}{ln(10)}$, is introduced by the distance estimation errors, it is the same in Eqs. (7) and (8). If the channel reciprocity also holds, i.e., $\chi_{AM_i} = \chi_{MA_i}$, Advanced *M* could also pass the symmetry check.

However, Advanced *M* is unlikely to pass the channel variation check since the channel variation is increased due to the distance estimation errors and the larger channel fading variation between *A* and *M*.

5.2.3 ***Security Analysis Against Supreme Attacker.*** Supreme *M* is a special case of Advanced *M*. We assume the supreme *M* can gain the precise instantaneous distance d_{AM_i} and d_{AB_i} without any estimation error. That is, $r_{AB_i} = r_{AM_i} = 0$, so $\tilde{d}_{AM_i} = d_{AM_i}$ and $\tilde{d}_{AB_i} = d_{AB_i}$. We also assume Supreme *M* does not have any measurement error, i.e., $e_{AM_i} = 0$. Therefore, Eq. (7) becomes

$$PL'^{AM}[i] = 10\alpha \log_{10}\left(\frac{4\pi d_{AB_i}}{\lambda}\right) + \chi_{AM_i}. \tag{9}$$

Similarly, Eq. (8) becomes

$$PL'^{MA}[i] = 10\alpha \log_{10}\left(\frac{4\pi d_{AB_i}}{\lambda}\right) + \chi_{MA_i} + e_{MA_i}. \tag{10}$$

Note that *A* may still experience the measurement error.

Similar to the security analysis against Advanced *M*, we can conclude that Supreme *M* could pass both the valley shape check and symmetry check. Although there is no distance measurement error introduced variation, the variation in the rebuilt pathloss is introduced by the channel fading between *A* and *M*. As long as $\sigma_{MA} > \sigma_{BA}$, Supreme *M* is likely to fail the channel fading variation check.

5.2.4 ***Importance of Tx Power Randomization.*** From Eq. (3), we can see that if $P^{A}_{Tx}[i]$ was fixed and known to Supreme *M*, he would be able to accurately compute the instantaneous channel fading χ_{AM_i} given that *M* is not subject to any measurement error, i.e., $e_{AM_i} = 0$.

$$\chi_{AM_i} = P^{\mathrm{A}}_{Tx_i}[i] - P^{\mathrm{M}}_{Rx_i}[i] - 10\alpha \log_{10}\left(\frac{4\pi d_{\mathrm{AM}_i}}{\lambda}\right) \tag{11}$$

With the knowledge on χ_{AM_i}, Supreme *M* can falsify $P'^{M}_{Rx}[i]$ and $P'^{M}_{Tx}[i]$ (assuming channel reciprocity holds, i.e., $\chi_{AM_i} = \chi_{MA_i}$) as follows:

$$\begin{aligned} P'^{M}_{Rx}[i] &= P^{M}_{Rx}[i] + 10\alpha log_{10}\left(\frac{\tilde{d}_{AM_i}}{\tilde{d}_{AB_i}}\right) + \chi_{AM_i} \\ P'^{M}_{Tx}[i] &= P^{M}_{Tx}[i] - 10\alpha log_{10}\left(\frac{\tilde{d}_{AM_i}}{\tilde{d}_{AB_i}}\right) - \chi_{MA_i} \end{aligned} \tag{12}$$

Similar to the derivation of Eqs. (9) and (10), we can obtain that

$$\begin{aligned} PL'^{AM}[i] &= 10\alpha \log_{10}\left(\frac{4\pi d_{AB_i}}{\lambda}\right) \\ PL'^{MA}[i] &= 10\alpha \log_{10}\left(\frac{4\pi d_{AB_i}}{\lambda}\right) + e_{MA_i} \end{aligned} \tag{13}$$

We can see that the computed $PL'^{AM}[i]$ is not subject to any fading or random error. Supreme *M* could add a very small randomness into the falsified report to make it look random and keep the variation below the threshold to pass the channel fading variation check.

Therefore, it is very important to randomize the Tx power at *A* and *B* to hide the channel fading variation from Supreme *M*. The randomization should be large enough to cover χ_{AM_i} and χ_{BM_i}. Next, we will show that Supreme *M* could construct P'^{M}_{Rx} and P'^{M}_{Tx} in a different way from Eq. (12) trying to pass the channel fading variation check. Then we will discuss what distribution the P^{A}_{Tx} and

P^B_{Tx} should follow to prevent Supreme M from passing the channel fading variation check.

5.2.5 ***Standard Deviation of Randomized Transmission Power.*** Since the analysis on P^A_{Tx} and P^B_{Tx} is similar, we will use P^A_{Tx} as an example. We will first show Supreme M can estimate the average Tx power $\overline{P}^A_{Tx}$ by averaging all the estimated $P^A_{Tx}[i]$ as follows:

$$\begin{aligned}\overline{P}^A_{Tx} &= \frac{1}{n}\sum_{i=1}^{n} P^A_{Tx}[i] = \frac{1}{n}\sum_{i=1}^{n}\left[P^M_{Rx}[i] + 10\alpha\log_{10}\left(\frac{4\pi d_{AM_i}}{\lambda}\right) + \chi_{AM_i}\right] \\ &\approx \frac{1}{n}\sum_{i=1}^{n}\left[P^M_{Rx}[i] + 10\alpha\log_{10}\left(\frac{4\pi d_{AM_i}}{\lambda}\right)\right]\end{aligned}, \tag{14}$$

where n is the total number of packets/samples. The approximation comes from the fact that χ_{AM_i} follows zero mean Gaussian distribution, so $\frac{1}{n}\sum_{i=1}^{n}\chi_{AM_i}$ approaches 0. Therefore, Supreme M can falsify its Rx power as follows:

$$P'^M_{Rx}[i] = \overline{P}^A_{Tx} - 10\alpha\log_{10}\left(\frac{4\pi d_{AB_i}}{\lambda}\right) \tag{15}$$

Then the pathloss computed at A will be:

$$\begin{aligned}PL'^{AM}[i] &= P^A_{Tx}[i] - P'^M_{Rx}[i] = P^A_{Tx}[i] - \overline{P}^A_{Tx} + 10\alpha\log_{10}\left(\frac{4\pi d_{AB_i}}{\lambda}\right) \\ &= 10\alpha\log_{10}\left(\frac{4\pi d_{AB_i}}{\lambda}\right) + \Upsilon_i,\end{aligned} \tag{16}$$

where $\Upsilon_i = P^A_{Tx}[i] - \overline{P}^A_{Tx}$, which should have zero mean and follow the same distribution type as $P^A_{Tx}[i]$ with the same standard deviation. Denoting the standard deviation of the random Tx power as σ_t, to prevent Supreme M from passing the channel fading variation check, we should have $\sigma_t > \sigma_{AM}$.

Note that the threshold value in the channel fading variation check should be adapted to different environments. In addition to distance, environmental factors can also influence channel fading. These environmental determinants of channel fading will be explored in detail in Section 6.

6 PERFORMANCE EVALUATION

In this section, we present experimental results to validate the security strength and usability of Swipe2Pair.

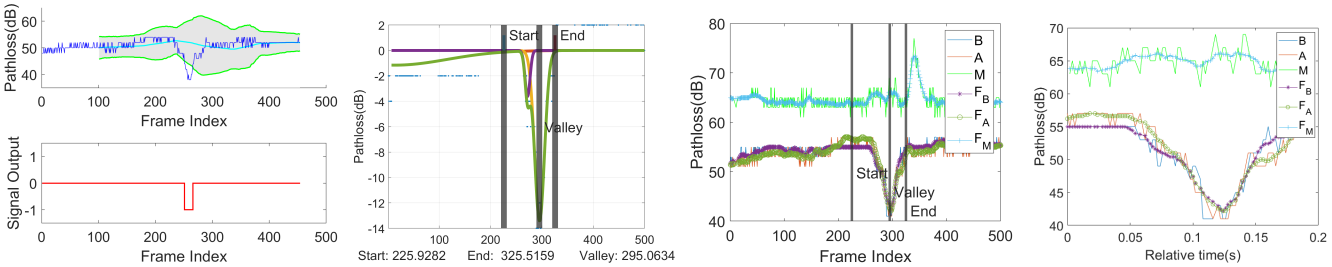
6.1 Experiment Setup

In our experiments, we used three Raspberry Pi4s to implement A, B, and M, as illustrated in Fig. 6. A and B were placed $0.1m$ from each other and M was at least $2m$ away from them. Each Pi4 was equipped with a PAU05 wireless dongle that was operated on channel 6 at 2.4 GHz. All three Pi4s were set to ad-hoc mode so they could extract the received signal strength indicator (RSSI) information for received/eavesdropped packets. As described in Section 4, A sent duplicated packets (including public key) at a rate of $500pkts/s$ during the swipe in front of B from left to right. Device B listened to packets from A and then quickly sent its public key back to A to ensure the reciprocity of the bidirectional channels largely holds. A and B randomized Tx power uniformly ranging from 0 to 30 dBm. Since the security of Swipe2Pair is fundamentally based on the pathlosses of the legitimate channel $A-B$ and attacking channels $A-M/B-M$, it is sufficient to put M in monitor mode to measure the RSSI for the packets sent by A/B rather than fully implement the three types of M. The randomized Tx power and monitored Rx power were logged locally on each device. The total pairing process was accomplished within one second.

6.2 Thresholds for Three Checks

We illustrated the process of the Valley shape detection and check from one experimental data in Fig.4. Initially, a valley shape was identified from noisy data in Fig. 4a. The green curves in the upper subplot represent the boundary for detecting the valley shape, calculated through the first 100 data points. The lower subplot identifies the valley position. Subsequently, we fit a model to locate the start and end points of the valley using the method proposed in [20] as illustrated in Fig. 4b and applied these to the original pathloss data in Fig. 4c to extract the range of the valley shape data in Fig. 4d. Typically, the valley shape shows a depth of 15 dBm with a valley point at around 40 dBm, spanning $100 \sim 200$ data points within 0.2 seconds. Note that no obvious valley shape was observed for the pathloss in the attacking channel.

By setting true positive rate $\beta = 90\%$ and the false positive rate $\alpha = 10\%$ in the symmetry check with 50 runs of experiments, a



(a) Valley shape detection from pathloss data **(b) Start, end, and valley points identification** **(c) Identified points in original pathloss data** **(d) Valley shape ranged data extraction**

Figure 4: From the original pathloss data computed through the Tx and Rx powers, the robust peak-valley detection algorithm is implemented to detect a valley shape from noisy data. If the valley shape is detected, we search for the start and end points of the valley shape and extract the valley shape ranged data for further checks.

threshold value $TH = 3.6$ for the MSE differentiated the legitimate channel *A/B* from the attacking channel *A/M* or *B/M* in Fig. 7. The MSE of the legitimate channel with the line-of-sight (LoS) path and minimal channel fading effect is below the threshold. In contrast, the MSE of the attacking channel that has a longer distance than the legitimate channel exceeds the threshold.

Similarly to channel fading variation with α and β in Fig.8, the threshold value of the standard deviation σ was set up to identify the attacking channel *A/M*. Although the distance measurement error has been eliminated from supreme *M*, the distribution of the standard deviation σ of the channel fading χ between legitimate and attacking channels is closed but still distinguishable. According to the same β and α, the threshold value $TH = 1.27$ effectively separates most of their σ distribution into two parts. In addition, we tested the attacking channel across various distances in our 50 runs of experiments and concluded the longer distance increases the separation between the two distributions. Besides, the environment is another factor influencing the channel fading variation σ, which will be further detailed in the subsequent discussion.

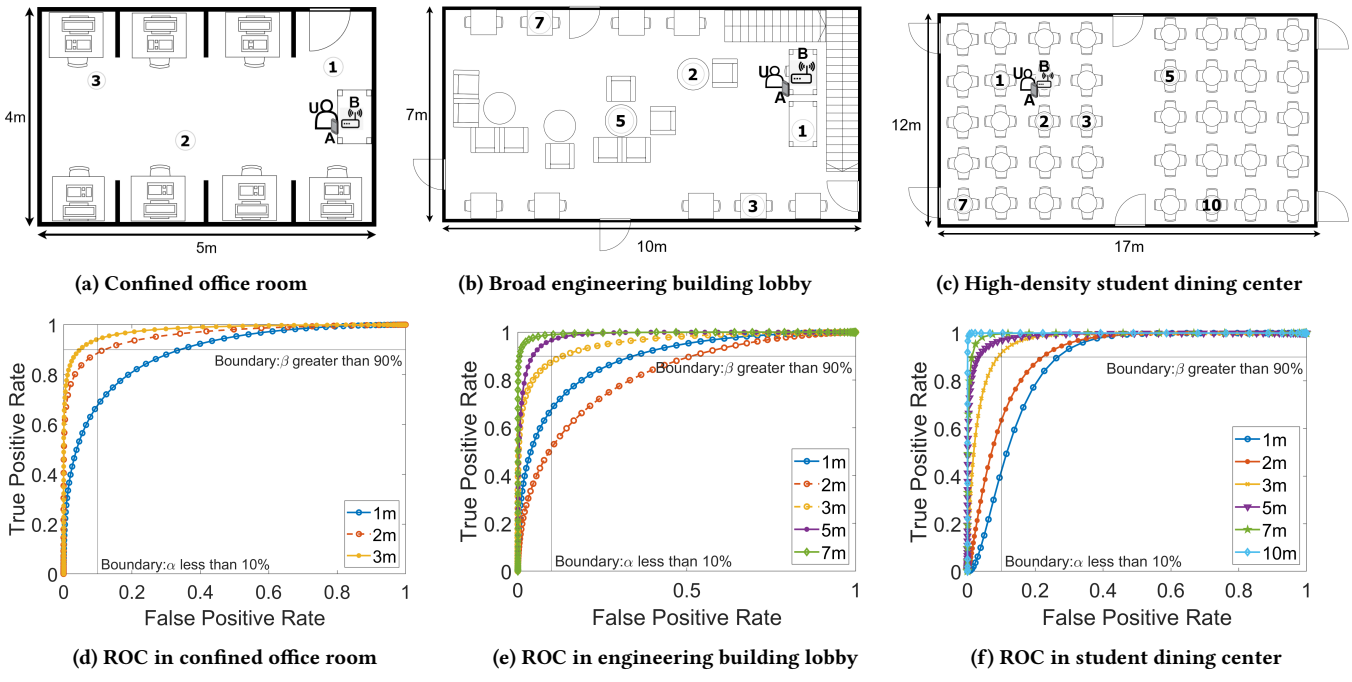
6.3 Security under Different Environments

To identify the appropriate distances for the remote Supreme *M* detection through his channel fading variation, we choose three typical environments and evaluate their channel fading performance with the same β and α in Fig.5: the confined office room, broad building lobby, and high-density student dining center.

1: Confined office room. Our office room furnishes 7 modern tables and chairs in Fig. 5a. The user (labeled as **U**) held device *A* (labeled as **A**) named 'Smartphone' to pair the device *B* (labeled as **B**) named 'IoT' on the right side of the room over the table. Their distance was 0.1 meters in proximity. The remote supreme *M* was placed over different positions with a similar height to *A* and *B*, tagged in numbers, which also represented the distance in meters the supreme *M* away from the pairing devices *A* and *B* as well. In Fig. 5d, the remote supreme *M* positioning at 2 barely satisfied both α and β because the confined office was narrow which caused more reflections from the wall but fewer obstacles of the shadowing effect so a closed distance could filter supreme *M*. Hence, we need at least two meters to have enough channel fading to filter the remote Supreme *M* in our confined office room.

2: Broad engineering building lobby. Our engineering building lobby serves as a more complex environment containing multiple 3-piece round bar table sets and coffee tables and sofa sets in Fig. 5b. *A* and *B* were still in proximity over the table and supreme *M* located at several different positions. Compared to the confined office, the remote Supreme *M* needed to be at least three meters away from pairing devices *A* and *B* to satisfy the α and β in Fig. 5e. A Broad lobby diminishes reflections from walls, yet more obstacles and human activities enhance the shadowing effect leading to a higher channel fading variation. Therefore, a longer distance with more obstacles is necessary to produce a sufficient channel fading to detect the remote Supreme *M*.

3: High-density student dining center. The last environment is the crowded student dining center with massive round dinner tables with four chairs each in Fig. 5c. We placed devices *A* and *B* on one table and examined the distance of Supreme *M* in various tables. As we expected, we observed increased channel fading variation in Fig. 5f. These data demonstrated that the longer the distance of remote Supreme *M*, the more pronounced the fading variation in his channel becomes. Meanwhile, this channel fading variation intensified significantly beyond a certain distance (e.g., 3 m) in



(a) Confined office room (b) Broad engineering building lobby (c) High-density student dining center

(d) ROC in confined office room (e) ROC in engineering building lobby (f) ROC in student dining center

Figure 5: ROC of channel fading variation corresponds to three different circumstances.

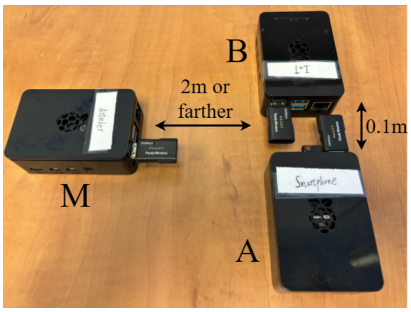


Figure 6: Layout of Two proximate pairing devices 'B' and 'A' and one remote attacker 'M'.

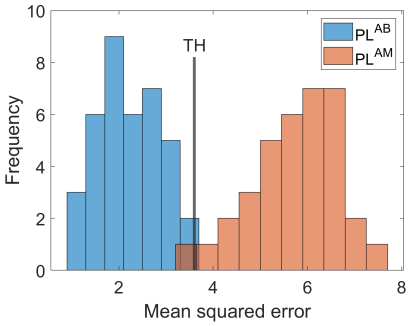


Figure 7: Distribution of mean squared error computed from proximate channel pathloss AB and remote channel pathloss AM.

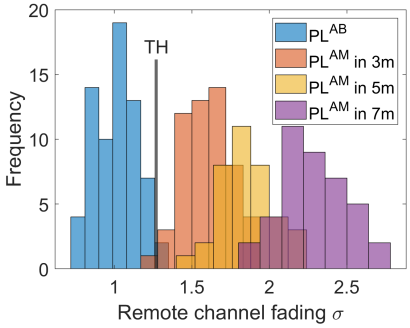


Figure 8: Distributions of the standard deviation σ from proximate and remote channel fading variation χ in different distances.

both the lobby and dining center. We attribute this to the presence of tables/chairs and people activities nearby [9]. Eventually, this pronounced channel fading variation can be utilized to identify the Supreme M in the channel fading check.

6.4 Valley Shape Robustness

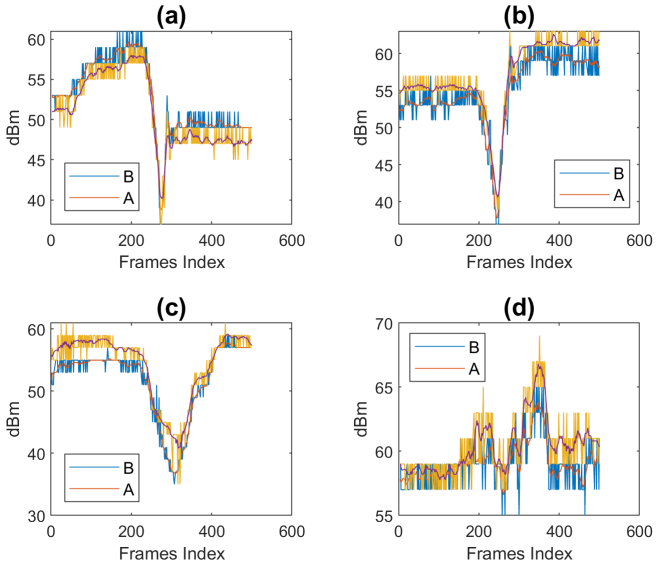


Figure 9: Four various valley shapes between the proximate pairing device *A* and *B* computed from the user's arbitrary one-quick swipe attempts.

We investigated the formation of a valley shape during the pairing process and listed four valley shapes generated from distinctive pairing motion in Fig. 9. Our observations revealed that in instances where two pairing devices were positioned within 10 cm of each other and a 500pkt/s sampling rate was utilized over a 1-second pairing duration, an identifiable valley shape could be generated through the algorithm [3]. This recognizable valley shape was consistent no matter swiping horizontally in 9a or symmetrical in 9b. It is also irrespective of the speed in Fig. 9c from a slow swiping. However, we observed the valley shape in Fig. 9d became flatter and less likely to be recognized as a valley shape as the distance between two legitimate devices increased. We conclude the valley shape is more influenced by the distance of pairing devices than the speed, alignment, and symmetry. A comprehensive study is necessary to explore the accurate relationships among these factors for a more robust valley shape detection.

7 CONCLUSION

We have introduced Swipe2Pair as a novel approach for achieving secure pairing of a new device in a wireless network that does not require a sophisticated user interface. Swipe2Pair provides a ubiquitous solution for in-band proximate device pairing as it only requires basic wireless communication capability, which is straightforward and quick to use. This approach enhances security by utilizing random transmission power and incorporates user interaction, channel reciprocity, and fading variation to ensure a robust pairing process.

We conducted a comprehensive security analysis of Swipe2Pair considering three typical attacking scenarios with three levels of attacker capability: General, Advanced, and Supreme. The security performance of Swipe2Pair was further validated through experimental data collected at different distances and various environments. Our analysis and experimental results demonstrate the robustness, reliability, and efficiency of Swipe2Pair as a secure in-band device pairing protocol.

8 ACKNOWLEDGEMENT

This material is based on research sponsored by the Defense Advanced Research Projects Agency (DARPA) under the agreement number HR001120C0154. The views, opinions, and/or findings contained in this article are those of the author(s) and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.

REFERENCES

[1] Omotayo G Adewumi, Karim Djouani, and Anish M Kurien. 2013. RSSI based indoor and outdoor distance estimation for localization in WSN. In *2013 IEEE International Conference on Industrial Technology (ICIT)*. IEEE, 1534–1539.

[2] Steven M Bellovin and Michael Merritt. 1994. An attack on the interlock protocol when used for authentication. *IEEE Transactions on Information Theory* 40, 1 (1994), 273–275.

[3] J.P.G van Brakel. 2014. Robust peak detection algorithm using z-scores. https://stackoverflow.com/questions/22583391/peak-signal-detection-in-realtime-timeseries-data/22640362#22640362. https://stackoverflow.com/questions/22583391/peak-signal-detection-in-realtime-timeseries-data/22640362#22640362

[4] Liang Cai, Kai Zeng, Hao Chen, and Prasant Mohapatra. 2011. Good Neighbor: Secure pairing of nearby wireless devices by multiple antennas. In *NDSS*.

[5] Michele Gallo, Peter S. Hall, Yuriy I. Nechayev, and Michele Bozzetti. 2008. Use of Animation Software in Simulation of On-Body Communications Channels at 2.45 GHz. *IEEE Antennas and Wireless Propagation Letters* 7 (2008), 321–324. https://doi.org/10.1109/LAWP.2008.928484

[6] Nirnimesh Ghose, Loukas Lazos, and Ming Li. 2017. {HELP}:{Helper-Enabled}{In-Band} Device Pairing Resistant Against Signal Cancellation. In *26th USENIX Security Symposium (USENIX Security 17)*. 433–450.

[7] Nirnimesh Ghose, Loukas Lazos, and Ming Li. 2020. In-band secret-free pairing for COTS wireless devices. *IEEE Transactions on Mobile Computing* 21, 2 (2020), 612–628.

[8] Zhenhua Gong and Martin Haenggi. 2014. Interference and Outage in Mobile Random Networks: Expectation, Distribution, and Correlation. *IEEE Transactions on Mobile Computing* 13, 2 (2014), 337–349. https://doi.org/10.1109/TMC.2012.253

[9] Elyes Ben Hamida and Guillaume Chelius. 2010. Investigating the impact of human activity on the performance of wireless networks - An experimental approach. In *2010 IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE, 1–8.

[10] Jun Han, Albert Jin Chung, Manal Kumar Sinha, Madhumitha Harishankar, Shijia Pan, Hae Young Noh, Pei Zhang, and Patrick Tague. 2018. Do you feel what I hear? Enabling autonomous IoT device pairing using different sensor types. In *IEEE Symposium on Security and Privacy (SP)*. 836–852.

[11] Omar Adel Ibrahim, Gabriele Oligeri, and Roberto Di Pietro. 2022. Eolo: IoT Proximity-based Authentication via Pressure Correlated Variations. In *2022 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 109–117.

[12] Rong Jin, Liu Shi, Kai Zeng, Amit Pande, and Prasant Mohapatra. 2014. MagPairing: Exploiting magnetometers for pairing smartphones in close proximity. In *IEEE Conf. on Comm. and Netw. Security*. 445–453.

[13] Sameh Khalfaoui, Jean Leneutre, Arthur Villard, Jingxuan Ma, and Pascal Urien. 2021. Security analysis of out-of-band device pairing protocols: a survey. *Wireless Communications and Mobile Computing* 2021 (2021), 1–30.

[14] K. S. Kumar and R. Sukumar. 2019. Achieving energy efficiency using novel scalar multiplication based ECC for Android devices in Internet of Things environments. *Cluster Computing* 22, 5 (2019), 12021–12028.

[15] Xiaopeng Li, Qiang Zeng, Lannan Luo, and Tongbo Luo. 2020. T2pair: Secure and usable pairing for heterogeneous IoT devices. In *ACM SIGSAC*. 309–323.

[16] Rene Mayrhofer and Hans Gellersen. 2009. Shake well before use: Intuitive and secure pairing of mobile devices. *IEEE Trans. Mobile Comput.* 8, 6 (2009), 792–806.

[17] Jason McGinthy and Alan Michaels. 2018. Session key derivation for low power IoT devices. In *2018 IEEE 4th International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing,(HPSC) and IEEE International Conference on Intelligent Data and Security (IDS)*. IEEE, 194–203.

[18] Shijia Mei, Zhihong Liu, Yong Zeng, Lin Yang, and Jian Feng Ma. 2019. Listen!: Audio-based smart iot device pairing protocol. In *2019 IEEE 19th International Conference on Communication Technology (ICCT)*. IEEE, 391–397.

[19] Salim Latif Mohammed. 2016. Distance estimation based on RSSI and Log-Normal shadowing models for ZigBee wireless sensor network. *Engineering and Technology Journal* 34, 15 (2016), 2950–2959.

[20] T O'Haver. 2022. Pragmatic Introduction to Signal Processing: Applications in scientific measurement: An illustrated handbook with free software and spreadsheet templates to download.

[21] Jafar Pourbemany, Ye Zhu, and Riccardo Bettati. 2022. Breathe-to-Pair (B2P) Respiration-Based Pairing Protocol for Wearable Devices. In *Proceedings of the 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. 188–200.

[22] Ronald L Rivest and Adi Shamir. 1984. How to expose an eavesdropper. *Commun. ACM* 27, 4 (1984), 393–394.

[23] Marc Roeschlin, Ivan Martinovic, and Kasper Bonne Rasmussen. 2018. Device Pairing at the Touch of an Electrode. In *NDSS*, Vol. 18. 18–21.

[24] Masoud Rostami, Ari Juels, and Farinaz Koushanfar. 2013. Heart-to-heart (H2H) authentication for implanted medical devices. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. 1099–1112.

[25] Shubham Saloni and Achyut Hegde. 2016. WiFi-aware as a connectivity solution for IoT pairing IoT with WiFi aware technology: Enabling new proximity based services. In *IEEE IOTA*. 137–142.

[26] Fengjun Shang, Wen Su, Qian Wang, Hongxia Gao, and Qiang Fu. 2014. A location estimation algorithm based on RSSI vector similarity degree. *International Journal of Distributed Sensor Networks* 10, 8 (2014), 371350.

[27] Muhammad Ali Siddiqi, Robert HSH Beurskens, Pieter Kruizinga, Chris I De Zeeuw, and Christos Strydis. 2021. Securing implantable medical devices using ultrasound waves. *IEEE Access* 9 (2021), 80170–80182.

[28] Zeinab Vahdati, Sharifah Yasin, Ali Ghasempour, and Mohammad Salehi. 2019. Comparison of ECC and RSA algorithms in IoT devices. *Journal of Theoretical and Applied Information Technology* 97, 16 (2019).

[29] Wei Wang, Zhan Wang, Wen Tao Zhu, and Lei Wang. 2015. WAVE: Secure wireless pairing exploiting human body movements. In *2015 IEEE Trustcom/BigDataSE/ISPA*, Vol. 1. IEEE, 1243–1248.

[30] Ye Wang, Wen-jun Lu, Hong-bo Zhu, et al. 2012. An empirical path-loss model for wireless channels in indoor short-range office environment. *International Journal of Antennas and Propagation* 2012 (2012).

[31] Yongdong Wu, Binbin Chen, Zhigang Zhao, and Yao Cheng. 2017. Attack and countermeasure on interlock-based device pairing schemes. *IEEE Transactions on Information Forensics and Security* 13, 3 (2017), 745–757.

[32] Weitao Xu, Chitra Javali, Girish Revadigar, Chengwen Luo, Neil Bergmann, and Wen Hu. 2017. Gait-key: A gait-based shared secret key generation protocol for wearable devices. *ACM Transactions on Sensor Networks (TOSN)* 13, 1 (2017), 1–27.

[33] Zhenyu Yan, Qun Song, Rui Tan, Yang Li, and Adams Wai Kin Kong. 2019. Towards touch-to-access device authentication using induced body electric potentials. In *The 25th Annual International Conference on Mobile Computing and Networking*. 1–16.

[34] Yuchen Yang, Longfei Wu, Guisheng Yin, Lijie Li, and Hongbin Zhao. 2017. A Survey on Security and Privacy Issues in Internet-of-Things. *IEEE Internet of Things Journal* 4, 5 (2017), 1250–1258. https://doi.org/10.1109/JIOT.2017.2694844

[35] Zheng Yang, Zimu Zhou, and Yunhao Liu. 2013. From RSSI to CSI: Indoor localization via channel response. *ACM Computing Surveys (CSUR)* 46, 2 (2013), 1–32.

[36] Ling-Yu Yeh, Po-Jen Chen, Chen-Chun Pai, and Tsung-Te Liu. 2020. An energy-efficient dual-field elliptic curve cryptography processor for Internet of Things applications. *IEEE Transactions on Circuits and Systems II: Express Briefs* 67, 9 (2020), 1614–1618.

[37] Jiansong Zhang, Zeyu Wang, Zhice Yang, and Qian Zhang. 2017. Proximity based IoT device authentication. In *IEEE INFOCOM*.

[38] Haiping Zhu and Talal Alsharari. 2015. An improved RSSI-based positioning method using sector transmission model and distance optimization technique. *International Journal of Distributed Sensor Networks* 11, 9 (2015), 587195.