2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS)

WaveKey: Secure Mobile Ad Hoc Access to RFID-Protected Systems

Dianqi Han*, Ang Li†, Jiawei Li‡, Yan Zhang§, Tao Li¶, Yanchao Zhang†
* University of Texas at Arlington † University of Michigan-Dearborn ‡ Arizona State University
§ University of Akron ¶ University of Michigan-Dearborn

***Abstract*—This paper presents the design and evaluation of WaveKey, a cross-modal deep learning-based method to enable *mobile ad hoc in-situ access* to RFID-protected cyber systems. Built upon the ever-growing popularity of user-carried mobile devices and RFID technologies, WaveKey is motivated by the need for secure and user-friendly data access in various application contexts. WaveKey explores a random gesture performed by the mobile user to induce correlated IMU data and RFID signals at the involved mobile device and RFID server, adopts deep learning techniques to extract the complex cross-modal correlation, and devises an Oblivious Transfer-based key-agreement protocol toward secure and efficient key establishment. Theoretical analysis and experimental human-based evaluation confirmed the high security and efficiency of WaveKey. In particular, WaveKey shows very high key-establishment success rates consistently exceeding 98% across all evaluated settings and renders extremely low success rates below 0.5% for all evaluated common attacks.**

***Index Terms*—mobile devices, RFID systems, key establishment.**

I. Introduction

This paper proposes **WaveKey**, a cross-modal deep learning-based method to enable *mobile ad hoc in-situ access* to RFID-protected systems. Leveraging the rising popularity of user-carried mobile devices and RFID technologies, WaveKey addresses the demand for secure and user-friendly data access in the following representative application contexts.

- *Context 1: RFID line-up service systems.* Visitors to service centers, customer support desks, government offices, hospitals, or similar locations receive a service ticket with a unique RFID tag. These tickets are distributed through automatic dispensers or receptionists. The RFID ticket communicates with the backend server, indicating the person's position in the queue for fair and organized service. Users can use their mobile devices to submit required paperwork wirelessly, which is tied to their RFID ticket numbers .
- *Context 2: RFID location-based access control.* In a restricted area, a non-removable, non-forgeable RFID card is installed to control and manage access to sensitive physical or electronic resources. Authorized personnel can conveniently access these resources using their mobile devices after verifying their physical proximity to the RFID card.
- *Context 3: RFID-assisted secure mobile system access.* RFID key fobs are commonly used in a wide range of settings, including residential and commercial buildings for keyless entry, employee access control in offices, vehicle keyless entry and ignition systems, and more. People may want to use their RFID key fobs to register random mobile devices for securely interacting with the corresponding system.

The above scenarios can be readily adapted to illustrate other application contexts that employ RFID access control. In all such applications, the establishment of an *ad hoc (cryptographic) key* between a user's mobile device and the RFID system becomes critical. This key is necessary to ensure the security of wireless communications between the mobile device and the RFID-protected system. Additionally, it can eliminate the necessity for a long-term user-specific cryptographic key, which might be challenging to establish/update and susceptible to key exposure.

WaveKey is specifically designed to fulfill the above critical requirement in a secure and user-friendly manner. As depicted in Fig. 1, its functionality involves the user waving their mobile device alongside the corresponding RFID device (e.g., a fresh RFID ticket, a secured RFID card, or a unique key fob) together randomly for a brief duration. The randomness introduced by the user's hand-waving gesture causes fluctuations in the IMU sensor on the mobile device and variations in the wireless signals detected by the backend RFID reader. These correlated fluctuations and variations are then leveraged to establish a common secret key, effectively confirming the co-location of the mobile and RFID devices.

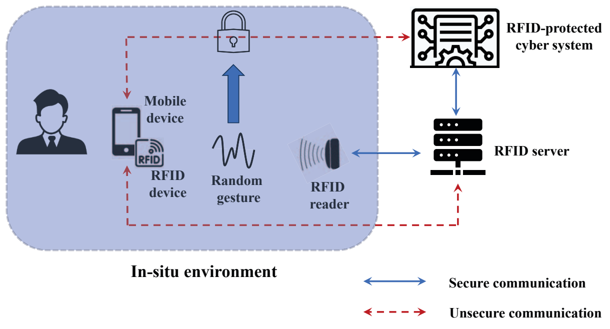


Fig. 1. WaveKey system context.

While the WaveKey design may appear straightforward, it encounters two critical challenges. The first challenge involves formulating the complex cross-modal relationship between the IMU data and RFID signals. The three-dimensional IMU data contains fine-grained spatial information about hand movements but is collected at a relatively low sampling rate (typically around 100 Hz) [1]. On the other hand, the RFID signal, with a higher sampling rate, provides richer temporal information about the gesture but only captures one-dimensional spatial information, specifically, the propagation distance. The presence of multi-path effects in RFID signal propagation further complicates this cross-modal relationship. The second challenge arises from the short duration of the random gesture, which introduces potential vulnerabilities to the system. The brevity of the gesture, necessary for ensuring a smooth user experience, limits the entropy that the gesture can provide for the key, making it potentially susceptible to guessing attacks. Additionally, the gesture is vulnerable to shoulder-surfing, wherein an attacker could easily infer the key by observing the gesture during its brief execution.

To address these challenges, we incorporate deep learning techniques and a sophisticated key-agreement protocol into the scheme design. Motivated by the success of deep learning in cross-modal tasks such as image caption generation [2], [3], we utilize it to extract the cross-modal correlation between the IMU data and RFID signal. Specifically, we train two autoencoders to project the IMU data and RFID signal to a latent space related only to the hand gesture. In this way, the mobile device and RFID system server can obtain two highly similar feature representations of the gesture. Based on the two feature representations, we design an Oblivious Transfer (OT)-based key-agreement protocol to generate a common key at both devices, which leverages a time-sensitive challenge-response mechanism to defeat possible attacks.

We implemented WaveKey on a commodity RFID reader and various mobile devices. Through extensive evaluation with six volunteers, we achieved negligible success rates ($<$ 0.5%) for all evaluated attacks, confirming the robust security of WaveKey. The scheme also demonstrated exceptional adaptability, with key-establishment success rates consistently exceeding 98% across all evaluated settings. Furthermore, the experimental results affirmed the high efficiency and low latency of the WaveKey system.

This paper is structured as follows. §II discusses related studies. §III presents the problem statement and adversary model. §IV illustrates the WaveKey design. §V analyzes the security of WaveKey against common attacks. §VI shows experimental evaluation results. §VII concludes this paper.

II. Related Work

Our WaveKey scheme is most germane to the rich literature on pairing two mobile devices or equivalently establishing a shared key between them. For example, ShakeUnlock [4] and WristUnlock [5] explore correlated IMU sensor data on two mobile devices for shared-key establishment. There are many schemes exploring the reciprocity of wireless channels [6]–[8] and acoustic channels [9]–[11] for shared-key generation. These schemes also rely on the same type of signals, such as Channel State Information (CSI), at two involved devices. Furthermore, researchers have designed many cryptography-based device pairing schemes that are effective in their targeted scenarios [12]–[15]. These schemes typically relies on a pre-shared user-specific cryptographic key [12]–[14] or a trusted third party [15], which may not be available in the application contexts WaveKey targets.

WaveKey differs significantly from existing device-pairing methods. First, WaveKey is a cross-modal deep learning-based approach that applies to disparate IMU and RFID data. Its methodology can be easily adapted to work with other heterogeneous sensing data for key agreement. Second, WaveKey does not rely on any pre-shared secret or trusted third party. Lastly, WaveKey achieves location-based in-situ access control for its dependence on a physically and information-wisely secure RFID device in the targeted application context.

WaveKey is also complementary to the numerous studies on RFID system security. For instance, hardware fingerprinting of RFID tags against tag forgery was explored in [16]–[18]. Two-factor authentication for RFID systems was proposed in [19], [20], designed for user's mobile devices and biometric information, respectively. The interrelation among RFID tags in a federated tag array was utilized to authenticate them in [21], [22]. Additionally, researchers have proposed various cryptographic designs to enhance the security of RFID communication protocols [23], [24]. While these schemes focus on securing the RFID system itself, they contribute to the RFID system security on which WaveKey relies.

III. Problem Statement and Adversary Model

Problem Statement. We address a generic problem setting applicable to all three exemplary contexts mentioned in §I after minor adaptations. A user wants to use a handheld mobile device (e.g., a smartphone, smartwatch, or tablet) to securely interact with an RFID-protected information system. The user and their mobile device can be unknown to the system, as seen in Contexts 1 and 3. They might also have been enrolled into the system, as observed in Context 2. Our WaveKey scheme is designed to work in both scenarios without any modification. Notably, WaveKey provides an additional security factor (i.e., location-based access control) in the second scenario.

The RFID system comprises a frond-end RFID device, a backend server, and an RFID reader that serves as a communication proxy between the former two. The RFID device can take many forms such as a service ticket in Context 1, a secured RFID card in Context 2, or a key fob in Context 3. We assume that existing administrative and information-security mechanisms are in place to ensure the physical security of the RFID device and the communication security of the entire RFID system. For example, the RFID service ticket can be generated on the fly and contain a random ID that only works for a short duration. The RFID card can be non-removable via a chained cable and may also use a dynamic ID for each new user. Additionally, sophisticated cryptographic and non-cryptographic RFID security schemes such as fingerprinting [18] can be deployed to thwart common attacks such as RFID cloning and counterfeiting. In addition, the mobile device and the RFID server can communicate via a wireless channel such as WiFi or Bluetooth. However, since these channels lack a pre-shared secret key, they are inherently insecure for transmitting sensitive data.

WaveKey aims to establish an *ad hoc key* between the user's mobile device and RFID server with the aid of the front-end RFID device in a secure and user-friendly manner. Since the RFID device is presumed to be physically and information-wise secure within the application venue, the ad hoc key not only verifies the physical presence of the mobile user but also enables secure interactions with the targeted system in all subsequent transactions.

Adversary Model. We are concerned about adversaries attempting to deduce the secret key established with our WaveKey scheme. If successful, they could compromise the secure communications between the mobile user and information system, leading to serious consequences such as unauthorized access, data theft, and falsification of sensitive information. In this scenario, we assume a white-box attack model, where the adversary possesses complete knowledge of the internal workings and structure of WaveKey. The adversary may employ the following attack strategies.

- **Eavesdropping.** The adversary sniffs the communication between the mobile device and the RFID server to obtain useful information for key inference.
- **Man-in-the-Middle (MitM).** The adversary disrupts the communication between the mobile device and the RFID reader and creates a malicious channel by relaying their messages to each other. Through this method, the attacker can modify the exchanged messages between the two devices to manipulate the established key.
- **RFID signal spoofing.** The adversary uses spoofed RFID signals to manipulate the key establishment process.
- **Device spoofing.** The adversary attempts to impersonate the RFID server during key establishment with the mobile device while disrupting their communications.

IV. WaveKey Design

A. Overview of WaveKey Workflow

WaveKey is versatile and can work with Low-Frequency (LF), High-Frequency (HF), and Ultra-High-Frequency (UHF) RFID systems with minor adaptations. In this paper, we use UHF RFID systems as an example in which RFID devices communicate with the reader via signal backscattering.

The WaveKey key establishment process is composed of three distinct phases, as visually depicted in Fig. 2. The initial phase, known as *Data Acquisition*, commences when the user simultaneously holds their mobile device and the secured RFID device in the application context using the same hand. During this brief interaction, the user performs a random gesture that lasts for only a few seconds. The mobile device is equipped with standard IMU sensors, including an accelerometer, a gyroscope, and a magnetometer. It diligently records and calibrates the changes in IMU sensor data that occur during the user's gesture. Simultaneously, the RFID server processes the backscattered signal variations induced by the user's gesture to obtain relevant data. In the subsequent *Key-Seed Generation* phase, the mobile device employs an autoencoder named IMU-En, accompanied by quantization and encoding operations, to generate a key-seed denoted as f_{M} based on its IMU data. Similarly, the RFID server derives its key-seed, denoted as f_{R}, using a distinct autoencoder called RF-En. In the final *Key Agreement* phase, the mobile device and RFID server collaboratively execute a bidirectional OT (Oblivious Transfer) protocol to establish a common key of the desired length, utilizing the two key seeds, f_{M} and f_{R}. This concludes the secure and efficient key establishment process within the WaveKey scheme.

The two autoencoders, IMU-En and RF-En, play pivotal roles in the WaveKey scheme. Utilizing deep learning techniques, they are meticulously crafted and trained to generate two highly similar key-seeds from disparate sensor data: IMU and RFID data. It is worth highlighting that IMU-En and RF-En are designed to work with any arbitrary combination of a mobile device and an RFID server, rather than being tailored for specific pairs. Once trained, these autoencoders can be seamlessly deployed in any instance of WaveKey to secure mobile ad hoc access to RFID-protected systems. It is noteworthy that the security of our proposed scheme is not dependent on keeping the two autoencoders concealed from the attacker.

B. Data Acquisition

1) Data collection: The mobile device and RFID server record data from their IMU sensors and RFID reader, respectively, while the user is performing the random gesture. Since WaveKey relies on the correlation between the two devices' collected data for key establishment, clock synchronization between them is necessary. However, there is an inevitable clock shift between them, impeding the subsequent key generation. We slightly tailor the gesture to address this issue. Particularly, we require the user to shortly pause their hand before performing the random gesture. In this way, both devices can detect the start of the hand movement from the significant variance increases in their respective data and then begin data recording, whereby the collected data are synchronized. Moreover, we only require a brief random gesture to minimize user effort. According to our experimental results, two seconds are long enough for secure key establishment, so WaveKey requires the random gesture to be slightly longer than that. Correspondingly, the mobile device records data from its gyroscope, accelerometer, and magnetometer for two seconds after detecting the start of the random gesture. The RFID device also records the *phases and magnitudes* of the RFID signals backscattered by the tag for the same period.

2) Data processing: After data collection, the mobile device calibrates its accelerometer data to obtain its linear accelerations during the random gesture. In particular, the mobile device first aligns its gyroscope, accelerometer, and magnetometer data through interpolation. We set 100 Hz as the

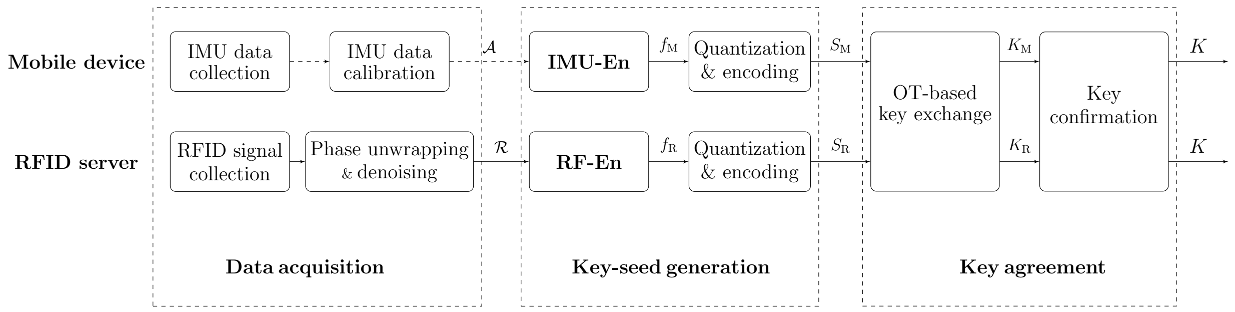


Fig. 2. The WaveKey workflow.

interpolation frequency since it fits the sampling rates of most modern mobile devices [1]. Next, the smartwatch estimates its poses, i.e., the directions of its x-, y-, and z-axes relative to the earth, across the random gesture. The initial pose at the beginning of the gesture can be obtained from the accelerometer and magnetometer measurements, while the subsequent poses can be estimated with the accumulated angular velocities measured by the gyroscope. During the short data collection period, the gyroscope drift is found to be negligible [25]. As a result, we opt to avoid the use of complex sensor fusion algorithms, such as the Kalman filter, in order to minimize computational overhead. Finally, the mobile device obtains 200 estimated linear accelerations by performing coordination transformation on the 200 accelerometer measurements. We represent the 200 linear accelerations with a 200×3 matrix and denoted it with $\mathcal{A}$.

Data processing at the RFID server includes phase unwrapping and denoising. Phase unwrapping aims to reveal the real changing trend of the RFID phase data because it is measured as the modulus of 2π and is wrapped in the range $[0,\ 2\pi]$. Particularly, we eliminate any phase jumping point, which refers to the phase measurement whose difference with the previous one exceeds π, by adding 2π or -2π to it. Next, we denoise the phase and magnitude data with two filters. We use the Savitzky-Golay smoothing filter [26] for both phase and magnitude data since it can preserve important data features, such as local maxima and minima, for efficient key generation. We represent the processed RFID data with a $2n \times 2$ matrix and denote it with $\mathcal{R}$. The first and second columns of $\mathcal{R}$ correspond to the processed phases and magnitudes, respectively. n denotes the sampling rate of the RFID reader, which equals 200 in our prototype. Hereafter, we will use $n = 200$ as an example to demonstrate WaveKey design.

C. Key-seed Generation

After data acquisition, the mobile device and RFID server proceed to obtain two similar vectors for key-seed generation. In particular, the mobile device feeds its $\mathcal{A}$ to IMU-En and obtains a vector, denoted as f_{M}, from the autoencoder output. Similarly, the RFID server obtains a vector, denoted as f_{R}, by feeding $\mathcal{R}$ to RF-En. We design IMU-En and RF-En to ensure that f_{M} and f_{R} are of the same length, denoted as l_f. As l_f is a hyperparameter significantly impacting system performance, its empirical value is determined through an experiment demonstrated in §VI. IMU-En and RF-En are trained to extract feature representations related to the random gesture from $\mathcal{A}$ and $\mathcal{R}$, respectively, resulting in highly similar feature vectors f_{M} and f_{R}.

Next, the two devices each obtain a bit sequence as the key-seed by quantizing their feature vectors. Every element in the two feature vector is a variable following a certain distribution, which can be converted to a short bit sequence through a routine quantization and encoding process [27]. In particular, its range is divided into many bins, which each correspond to a pre-defined bit sequence. In key generation, these bins are designed so the variable falls into them with the same probability, which maximizes the randomness of converted bit sequence for a secure key. Therefore, the boundary between bins i and $i+1$ is determined by solving the equation below:

$$\Phi(b_i) = i/N_b, \tag{1}$$

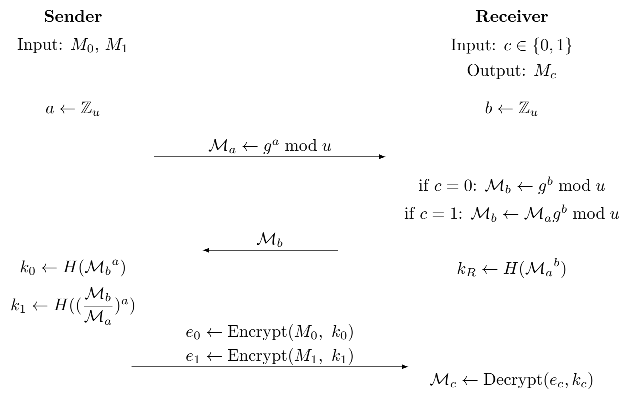
where $\Phi(\cdot)$ denotes the cumulative distribution function of the distribution, N_b denotes the total number of quantization bins, and b_i denotes the boundary between bins i and $i+1$. In WaveKey, we design the IMU-En and RF-En to both end with batch-norm layers, whereby all the elements in f_{M} and f_{R} follow the normal distribution and can employ the same bin setting for quantization. In addition, we determine the bin amount N_b through an experiment demonstrated in §VI and adopt the gray code [28] for encoding. Concatenating the bit sequences extracted from all the elements in their feature vectors, the mobile device and RFID server each obtain an l_s-bit long sequence as the key-seed, where

$$l_s = l_f \cdot \log_2 N_b. \tag{2}$$

We denote the mobile device's and the RFID server's key-seeds with S_{M} and S_{R}, respectively. Obviously, S_{M} and S_{R} are very similar and only slightly differ in a few bits.

D. Key Agreement

In the final stage, the mobile device and RFID server cooperate to perform key agreement that is designed upon the 1-out-of-2 Oblivious Transfer (OT) to obtain a common key in a secure fashion, which can be much longer than their key-seeds. This section first briefly reviews 1-out-of-2 OT and then introduces our key agreement protocol in detail.



Sender
Input: M_0, M_1
Receiver
Input: $c \in \{0,1\}$
Output: M_c
$a \leftarrow \mathbb{Z}_u$
$b \leftarrow \mathbb{Z}_u$
$\mathcal{M}_a \leftarrow g^a \bmod u$
if $c = 0$: $\mathcal{M}_b \leftarrow g^b \bmod u$
if $c = 1$: $\mathcal{M}_b \leftarrow \mathcal{M}_a g^b \bmod u$
$\mathcal{M}_b$
$k_0 \leftarrow H({\mathcal{M}_b}^a)$
$k_R \leftarrow H({\mathcal{M}_a}^b)$
$k_1 \leftarrow H((\frac{\mathcal{M}_b}{\mathcal{M}_a})^a)$
$e_0 \leftarrow \text{Encrypt}(M_0,\ k_0)$
$e_1 \leftarrow \text{Encrypt}(M_1,\ k_1)$
$\mathcal{M}_c \leftarrow \text{Decrypt}(e_c, k_c)$

Fig. 3. 1-out-of-2 OT nutshell.

1) 1-out-of-2 OT: 1-out-of-2 OT allows the receiver to selectively obtain one of the sender's two secrets, denoted as M_0 and M_1, respectively. In this process, the receiver cannot acquire the secret not selected, the sender is unaware of which of the two secrets is selected, and a third party gains no information regarding the two secrets or which of them is selected [29]. Among the many OT protocols, WaveKey incorporates the computational efficient one proposed in [30], with its nutshell demonstrated in Fig. 3. In particular, the sender and receiver agree on two large prime numbers, denoted as g and u, which are not necessarily hidden from a third party. The sender starts an OT instance by sending $\mathcal{M}_a = (g^a \bmod u)$ to the receiver, where a is a random integer in [0, u]. The receiver responds with a message crafted according to which secret the receiver would like to obtain. Particularly, the receiver responses with $\mathcal{M}_b = (g^b \bmod u)$ to obtain M_0 and responses with $\mathcal{M}_b = (\mathcal{M}_a g^b \bmod u)$ otherwise. The sender encrypts M_0 and M_1 with the hash values of ${\mathcal{M}_b}^a$ and $(\mathcal{M}_b/\mathcal{M}_a)^a$, respectively, and sends the two ciphers to the receiver, where only the one carrying the selected secret can be decrypted.

2) OT-based key agreement: The key agreement protocol is essentially a bidirectional 1-out-2 OT, which is designed to ensure that a malicious device cannot impersonate either of the two devices to the other one. In brief, the two devices obliviously transmit their many pairs of random bit sequences to each other as the parts of the eventually established key. The nutshell is demonstrated in Fig. 4. Particularly, the mobile device and RFID server each generate l_s pairs of random bit sequences, where l_s denotes the common length of their key-seeds. We denote the i-th sequence pairs of the mobile device and RFID server with $\langle x_i^0, x_i^1 \rangle$ and $\langle y_i^0, y_i^1 \rangle$, respectively. All the bit sequences are l_b-bit long, which is determined by the length of the desired key. Assuming that an l_k-bit long key is needed, they determine $l_b = \lceil l_k/(2 \cdot l_s) \rceil$.

Next, the two devices obliviously transmit one sequence in each of their sequence pairs to each other. Specifically, the mobile device obliviously transmits the sr_i-th sequence of the i-th sequence pair, i.e., $x_i^{sr_i}$, to the RFID server, where $sr_i \in \{0, 1\}$ denotes the i-th bit in the RFID server's key-seed S_{R}. Similarly, the RFID server obliviously transmits $y_i^{sm_i}$ to the RFID server, where sm_i denotes the i-th bit in the mobile device's key-seed S_{M}. After all the OT instances, the mobile device selects one sequence from each of its sequence pairs based on its key-seed and concatenate them with the sequences it obtained from the RFID server to obtain a preliminary key $K_{\text{M}} = (x_1^{sm_1} \parallel y_1^{sm_1} \parallel \cdots \parallel x_{l_s}^{sm_{l_s}} \parallel y_{l_s}^{sm_{l_s}})$. The RFID server obtains its preliminary key $K_{\text{R}} = (x_1^{sr_1} \| y_1^{sr_1} \| \cdots \| x_{l_s}^{sr_{l_s}} \| y_{l_s}^{sr_{l_s}})$ through a similar process. Obviously, the counterparts in K_{M} and K_{R}, i.e., $x_i^{sm_i} \parallel y_i^{sm_i}$ and $x_i^{sr_i} \parallel y_i^{sr_i}$, are the same if $sm_i = sr_i$, and may differ otherwise. Therefore, the ratio of mismatched bits between K_{M} and K_{R} is no more than that between S_{M} and S_{R}.

Finally, the two devices negotiate to mitigate the difference between their preliminary keys and confirm the final key. Specifically, the mobile device sends the error correction code (ECC) of its key K_{M} together with a nonce N to the RFID server as a challenge. The RFID server adjusts its key K_{R} accordingly to obtain K_{M} as the final key K and responds to the mobile device with the hash-based message authentication codes (HMAC) of the nonce N using K as the password. After verifying the HMAC of N with K_{M}, the mobile device also adopt K_{M} as the final key K.

We combine all the OT instances of the same direction into one for a more efficient communication and set a deadline for the arrival times of critical messages to defeat possible message forgery. In particular, an OT instance involves the transmission of three messages, i.e., $\mathcal{M}_a$, $\mathcal{M}_b$, and $\langle e_0,\ e_1 \rangle$ in Fig. 3. The mobile device transmits the counterpart messages to $\mathcal{M}_a$, $\mathcal{M}_b$, and $\langle e_0,\ e_1 \rangle$ associated with all the OT instances together as three messages: $\mathcal{M}_{\mathcal{A},\text{M}} = m_1 \parallel m_2 \parallel \ldots \parallel m_{l_s}$, $\mathcal{M}_{\mathcal{R},\text{M}} = n_1 \| n_2 \| \ldots \| n_{l_s}$, and $\mathcal{M}_{\mathcal{E},\text{M}}$= $\langle e_1^0, e_1^1 \rangle \| \ldots \| \langle e_{l_s}^0, e_{l_s}^1 \rangle$, respectively. The RFID server acts similarly and only needs to transmit three messages, $\mathcal{M}_{\mathcal{A},\text{R}}$, $\mathcal{M}_{\mathcal{R},\text{R}}$, and $\mathcal{M}_{\mathcal{E},\text{M}}$, to obtain its preliminary key. In addition, the mobile device and RFID server must receive $\mathcal{M}_{\mathcal{A},\text{R}}$ and $\mathcal{M}_{\mathcal{B},\text{M}}$ no later than $2+\tau$ seconds after the start of the gesture. Otherwise, the key-establishment instance is discarded. We determine τ through experiment so that the two devices of key establishment have enough time to craft their respective messages, but an adversary cannot catch up. We demonstrate more details regarding the experiment in §VI.

E. Autoencoder Training

This section explores the process of obtaining the two autoencoders, IMU-En and RF-En, through dataset generation and model training. As described in §IV-C, the main goal of WaveKey is to leverage IMU-En and RF-En to extract the

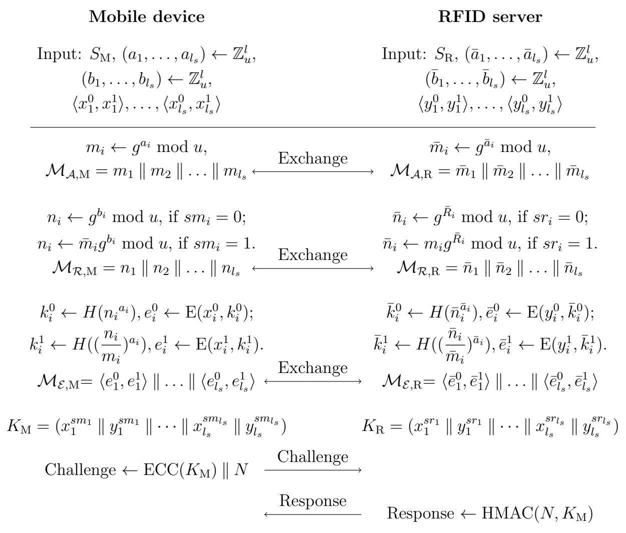

Mobile device
RFID server
Input: S_{M}, $(a_1, \ldots, a_{l_s}) \leftarrow \mathbb{Z}_u^l$, $(b_1, \ldots, b_{l_s}) \leftarrow \mathbb{Z}_u^l$, $\langle x_1^0, x_1^1 \rangle, \ldots, \langle x_{l_s}^0, x_{l_s}^1 \rangle$
Input: S_{R}, $(\bar{a}_1, \ldots, \bar{a}_{l_s}) \leftarrow \mathbb{Z}_u^l$, $(\bar{b}_1, \ldots, \bar{b}_{l_s}) \leftarrow \mathbb{Z}_u^l$, $\langle y_1^0, y_1^1 \rangle, \ldots, \langle y_{l_s}^0, y_{l_s}^1 \rangle$
$m_i \leftarrow g^{a_i} \bmod u$, $\mathcal{M}_{\mathcal{A},\mathrm{M}} = m_1 \parallel m_2 \parallel \ldots \parallel m_{l_s}$
Exchange
$\bar{m}_i \leftarrow g^{\bar{a}_i} \bmod u$, $\mathcal{M}_{\mathcal{A},\mathrm{R}} = \bar{m}_1 \parallel \bar{m}_2 \parallel \ldots \parallel \bar{m}_{l_s}$
$n_i \leftarrow g^{b_i} \bmod u$, if $sm_i = 0$; $n_i \leftarrow \bar{m}_i g^{b_i} \bmod u$, if $sm_i = 1$. $\mathcal{M}_{\mathcal{R},\mathrm{M}} = n_1 \parallel n_2 \parallel \ldots \parallel n_{l_s}$
Exchange
$\bar{n}_i \leftarrow g^{\bar{R}_i} \bmod u$, if $sr_i = 0$; $\bar{n}_i \leftarrow m_i g^{\bar{R}_i} \bmod u$, if $sr_i = 1$. $\mathcal{M}_{\mathcal{R},\mathrm{R}} = \bar{n}_1 \parallel \bar{n}_2 \parallel \ldots \parallel \bar{n}_{l_s}$
$k_i^0 \leftarrow H({n_i}^{a_i}), e_i^0 \leftarrow \mathrm{E}(x_i^0, k_i^0)$; $k_i^1 \leftarrow H((\frac{n_i}{m_i})^{a_i}), e_i^1 \leftarrow \mathrm{E}(x_i^1, k_i^1)$. $\mathcal{M}_{\mathcal{E},\mathrm{M}} = \langle e_1^0, e_1^1 \rangle \parallel \ldots \parallel \langle e_{l_s}^0, e_{l_s}^1 \rangle$
Exchange
$\bar{k}_i^0 \leftarrow H(\bar{n}_i^{\bar{a}_i}), \bar{e}_i^0 \leftarrow \mathrm{E}(y_i^0, \bar{k}_i^0)$; $\bar{k}_i^1 \leftarrow H((\frac{\bar{n}_i}{\bar{m}_i})^{\bar{a}_i}), \bar{e}_i^1 \leftarrow \mathrm{E}(y_i^1, \bar{k}_i^1)$. $\mathcal{M}_{\mathcal{E},\mathrm{R}} = \langle \bar{e}_1^0, \bar{e}_1^1 \rangle \parallel \ldots \parallel \langle \bar{e}_{l_s}^0, \bar{e}_{l_s}^1 \rangle$
$K_{\mathrm{M}} = (x_1^{sm_1} \parallel y_1^{sm_1} \parallel \cdots \parallel x_{l_s}^{sm_{l_s}} \parallel y_{l_s}^{sm_{l_s}})$
$K_{\mathrm{R}} = (x_1^{sr_1} \parallel y_1^{sr_1} \parallel \cdots \parallel x_{l_s}^{sr_{l_s}} \parallel y_{l_s}^{sr_{l_s}})$
Challenge $\leftarrow \mathrm{ECC}(K_{\mathrm{M}}) \parallel N$
Challenge
Response
Response $\leftarrow \mathrm{HMAC}(N, K_{\mathrm{M}})$

Fig. 4. The nutshell of the WaveKey key agreement protocol.

correlation between the linear acceleration matrix $\mathcal{A}$ and the RFID data matrix $\mathcal{R}$. The primary objective is to ensure that IMU-En and RF-En produce feature vectors with three key properties: i) relevance only to the random hand movement; ii) high similarity between the feature vectors; and iii) retention of crucial information about the hand movement trajectory for efficient key establishment. We carefully design the dataset generation and model training processes to achieve these objectives effectively.

1) Dataset generation: We implemented WaveKey, as described in §VI-A, and engaged six volunteers to perform random gestures for data collection. To ensure the adaptability of the trained autoencoders to various key-establishment scenarios, we conducted data collection in diverse environments and with different mobile devices. The data collection used three cellphones and one smartwatch. Each volunteer performed a total of 30 random gestures, each lasting longer than 15 seconds, with each of the four mobile devices. Out of the 30 gestures, 20 were performed in two distinct static environments, with 10 gestures in each environment. The remaining 10 gestures were conducted in a dynamic environment with human movement. Overall, we collected data from $6 \times 4 \times 30 = 720$ gestures. For each gesture, we randomly selected 20 time windows, each of which was two seconds long and may have overlapped with others. We processed the data collected during each time window, as demonstrated in §IV-B, to obtain a linear acceleration matrix and an RFID signal matrix, which together constituted a data sample. In the end, we amassed a dataset comprising 14,400 samples, denoted as $\mathcal{D} = \{\langle \mathcal{A}_i, \mathcal{R}_i \rangle | \ i = 1, 2, \cdots, 14400\}$. Here, $\mathcal{A}_i$ and $\mathcal{R}_i$ represent the linear acceleration matrix and the RFID signal matrix of the i-th sample, respectively. This extensive dataset forms the foundation for training the autoencoders, enabling the effective correlation extraction between the linear acceleration and RFID data for secure key establishment.

2) Model training: We design IMU-En and RF-En as two Convolutional Neural Networks (CNNs) and incorporate an auto-decoder, denoted as De, for their training. The diagrams of these three neural networks are illustrated in Fig. 5. IMU-En and RF-En both comprises two convolutional layers, which are followed by ReLu units, one fully connected layer, and one batch-norm layer. De comprises four layers, with the first and third layers being deconvolutional layers, and the second and fourth layers being fully connected layers. ReLu activation units are employed after the first three layers of De. A notable aspect of our design is that we choose to conclude IMU-En and RF-En with batch-norm layers. This strategic decision ensures that the elements of the output feature vectors $f_{\mathrm{M},i}$ and $f_{\mathrm{R},i}$ conform to a normal distribution, which greatly facilitates the quantization setting discussed in §IV-C.

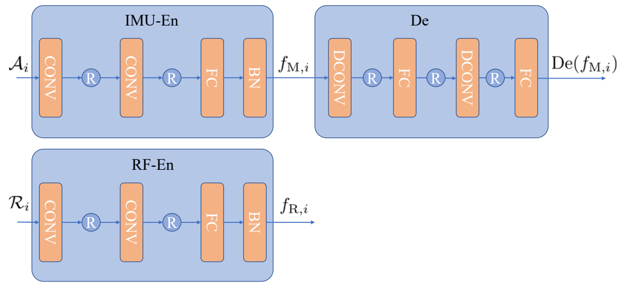


Fig. 5. WaveKey encoder-decoder diagram.

We jointly train IMU-En, RF-En, and De, as demonstrated in Fig. 6. Given a training sample $\langle \mathcal{A}_i, \mathcal{R}_i \rangle$, IMU-En and RF-En take $\mathcal{A}_i$ and $\mathcal{R}_i$ as their respective inputs and produce two feature vectors, denoted as $f_{\mathrm{M},i}$ and $f_{\mathrm{R},i}$, respectively. We intend to minimize the difference between the two feature vectors so that the key-seeds generated from them can be close enough to facilitate key agreement. Meanwhile, we utilize De to recover $\mathcal{R}_i$ from $f_{\mathrm{M},i}$ so that the two feature vectors still retain enough randomness induced by the gesture to produce two highly random key-seeds. However, during our experiments, we observed that the RFID phase data is highly sensitive to changes in the environment, whereas such variations have negligible impacts on the IMU data. As a consequence, attempting to accurately recover $\mathcal{R}_i$ solely from $f_{\mathrm{M},i}$ has proven to be almost impossible. To address this issue, we design De to recover the RFID magnitude data instead, which is more robust against environment changes. We utilize the following loss function for model training:

$$\mathcal{L} = \sum_{\langle \mathcal{A}_i, \mathcal{R}_i \rangle \in \mathcal{D}} (\| f_{\mathrm{M},i}, \ f_{\mathrm{R},i} \|_2 + \lambda \cdot \| \mathrm{De}(f_{\mathrm{M},i}), \ \mathcal{R}_i^{\mathrm{Mag}} \|_2), \quad (3)$$

where $\|\cdot\|_2$ measures the Euclidean distance between two vectors. $\mathcal{R}_i^{\mathrm{Mag}}$ denotes the magnitude part of $\mathcal{R}$, i.e., a vector comprising the 400 RFID signal magnitudes. λ is a scalar, which we determined as 0.4 through empirical experiments.

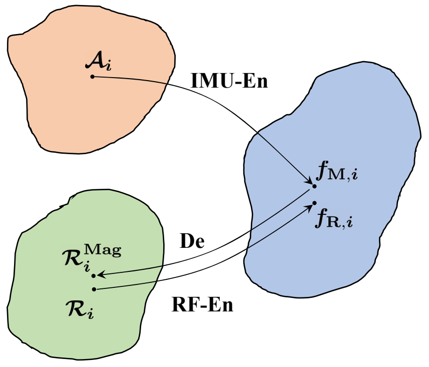


Fig. 6. The intuition of the WaveKey autoencoder training.

V. Security Analysis

This section analyzes the high security of WaveKey against various attacks, including eavesdropping, RFID signal spoofing, device spoofing, and the MitM attack.

A. Eavesdropping and RFID Signal Spoofing

In the eavesdropping attack, the mobile device and RFID server exchange messages during the key-agreement stage, making it possible for an adversary to intercept these communications. However, our key-agreement protocol is based on the highly secure OT primitive, which is immune to eavesdropping attempts. Therefore, WaveKey inherently maintains its robustness against eavesdropping attacks.

In the signal-spoofing attack, the adversary aims to manipulate the RFID signals reaching the RFID server. However, this attack will have no impact on the IMU data collected by the mobile device, leading to a disruption in the correlation between the IMU and RFID data. Consequently, the key-seeds generated by the mobile device and RFID server will be significantly different, leading to key-agreement failure. As a result, signal-spoofing attacks cannot assist the adversary in obtaining the common key and are easily detectable through increasing key-establishment failure rates.

B. Device Spoofing

In device spoofing, the adversary impersonates the RFID server to perform key establishment with the mobile device while jamming the communication between them. Once successful, the adversary can directly request sensitive information from the mobile device. Obtaining the key-seed S_{M} generated at the mobile device is the key to successful device spoofing. Generally, the adversary may manage to obtain S_{M} through three different ways listed as follows.

1) Random guessing: The adversary makes a random guess on S_{M} whereby to perform key establishment with the mobile device. We denote the adversary's guess, which is a bit sequence, with S_{G}. Since the key-agreement protocol adopts ECC to mitigate the inevitable small difference between the two benign key-seeds, the attack would succeed if the ratio of mismatch bits between S_{G} and S_{M} is below the error correction rate of ECC. Accordingly, we derive the success rate of device spoofing based on random guessing as

$$P_{\mathrm{g}} = \sum_{i=0}^{l_s \cdot \eta} \binom{l_s}{i} \Big/ 2^{l_s} \,, \tag{4}$$

where η denotes the error correction rate of the ECC code, and l_s denotes the length of the key-seed. η and l_s are two hyperparameters of our scheme, which we manage to obtain through experiments as 0.04 and 38, respectively. The corresponding success rate P_{g} is around 0.04%. Therefore, the adversary can hardly succeed in device spoofing by making a random guess on S_{M}.

2) Gesture mimicking: The adversary attempts to infer S_{M} by imitating the user's random gesture. In this scenario, the adversary observes the user's gesture in close proximity and replicates it while holding their own mobile device. Subsequently, the adversary processes the IMU data collected by their mobile device and generates a key-seed, denoted as S_{C}, for device spoofing. To assess the effectiveness of this attack, we conducted experiments and found that the success rate is no more than 0.2%. Therefore, device spoofing based on gesture mimicking does not compromise the security of WaveKey.

3) Data recovery: The adversary employs various sensors to track the user's hand movement trajectory during the execution of random gestures, aiming to estimate the IMU data S_{M} and deduce the established key. Notable sensor types for this purpose include acoustic transceivers [31], [32], mmWave radars [33], [34], conventional RGB cameras [35], and advanced RGB-D cameras [36], [37]. The accuracy of the estimated IMU sensor data is critically dependent on the precision of the captured hand movement trajectory, making highly accurate hand tracking essential for successful key inference. Since object tracking accuracy decreases with increasing distance between the sensor and the object, positioning the sensor close to the user is imperative to achieve the necessary hand tracking precision. Consequently, bulky RGB-D cameras are impractical for implementation of this attack.

Among other three sensor types, conventional RGB cameras exhibit the best hand tracking accuracy. The adversary may opt for two distinct approaches: deploying a concealed camera to record the user's gesture and stream the video in real-time to a remote server for video processing, or utilizing a portable device, such as a smartphone with an integrated camera, to directly record the gestures and execute on-device video processing. These methods are referred to as remote and in-situ recording strategies, respectively. As demonstrated in §VI-E, the remote recording-based attack, when combined with advanced 3D object positioning algorithms, demonstrates a trivial probability of 0.5% in successfully obtaining a valid key-seed. Moreover, the inherent latency associated with video streaming impedes the adversary's from replying the valid key-seed to the user's mobile device within the stipulated time window. In-situ recording is confined to real-time execution of comparatively less complex object positioning algorithms, which do not yield IMU data estimations of adequate fidelity to generate a valid key-seed.

In summary, WaveKey is highly secure against all three kinds of device spoofing attacks.

C. Man-in-the-Middle (MitM) Attack

The MitM attack on wireless communications can only occur during the key agreement stage. In this attack, the adversary jams the benign communication channel between the two devices and establishes a malicious channel by relaying their messages to each other. This allows the adversary to manipulate the exchanged messages between the two devices. However, the adversary is unaware of two device's secret parameters selected for OT and cannot manipulate the established key in this way. In particular, any modification of the messages between the two devices results in a significant difference between their preliminary keys. This leads to the failure of the key establishment and also exposes the adversary.

VI. Evaluation

In this section, we evaluate the performance of WaveKey with prototyped experiments.

A. Implementation

For evaluation purposes, we developed a prototype of WaveKey utilizing a commodity RFID reader. For hardware, we employed an Impinj Speedway R420 RFID reader with one Laird S9028 antenna to collect RFID data. The reader's sample rate was set to 200 Hz, and it was connected to a Dell Precision laptop for data processing. We used four mobile devices, including a Google Pixel 8, two Samsung Galaxy 5 phones, and a Samsung Galaxy Watch, to collect IMU data for evaluation. The evaluation involves 6 tags of three distinct models as RFID devices, including two Alien 9640, two Alien 9730, and two SMARTRAC DogBone. Regarding software, we implemented a mobile application to collect and process IMU data and a Python application running on the laptop to process the RFID data. We implemented and trained the two autoencoders using PyTorch [38]. In addition, we recruited six volunteers for our evaluation who are all graduate students.

B. Default Experiment Settings

Our evaluation defaulted to the following experimental settings: the experiments were conducted in a laboratory room devoid of any moving objects; the Samsung Galaxy Watch and an Alien 9640 tag were employed as the mobile and RFID devices, respectively; the volunteer tasked with performing random gestures for key generation remained 5 m away from the RFID antenna with an azimuth angle of 0°. Unless specified otherwise, these settings were consistently applied to all subsequent experiments.

C. Hyperparameter Determination

The performance of WaveKey relies heavily on the appropriate configuration of several important hyperparameters, including the length l_f of the feature vectors produced by the two autoencoders, the quantization setting for key-seed generation, and the time window τ. This section presents how we determined the values of these hyperparameters through rigorous experimentation.

1) l_f: The factor l_f influences the length and randomness of the key-seeds, both of which are critical for the security of WaveKey. A larger l_f results in longer key-seeds that are difficult for an adversary to guess. Paradoxically, a larger l_f also introduces more redundancy into the feature vectors, which reduces the randomness of the key-seeds and thus potentially compromises their cryptographic security. Our goal is to configure l_f in a way that balances the length and randomness of the key-seeds for the optimal security.

In the experiment, we derived the value of l_f through model pruning. Specifically, we initially implemented and trained the two autoencoders with $l_f = 50$, which is sufficient to fully capture the useful information from the input data for key-seed generation. Next, we gradually reduced l_f by eliminating two neurons each time from the fully connected layers of the two autoencoders, one from each. The neuron removal followed an ascending order with respect to the output variance of the neuron. In particular, we computed the output variances of all these neurons using the training dataset $\mathcal{D}$ and removed the neuron with the lowest variance as it contained less useful information for the key-seeds. Following each adjustment, we retrained the autoencoders with $\mathcal{D}$ and recorded the loss calculated according to Eq. (3). We halted the pruning process once the loss increased by more than 5% after one round of adjustment, signifying that further pruning would result in a significant loss of information during the feature extraction process. Eventually, we configured l_f as 12 and also obtained the corresponding well-trained autoencoders for the subsequent experiments.

2) Quantization settings: As discussed in §IV-C, the number of quantization bins, denoted as N_b, is the only parameter to be determined since all the elements to be quantized follows the normal distribution. N_b is another hyper parameter affecting the security of our scheme in a way similar to l_f. As with l_f, a large N_b results in longer key-seeds that are hard to guess for an adversary. However, it may also cause a larger bit mismatch rate between S_{M} and S_{R}, which requires a higher error correction rate η for the ECC code. According to Eq. (4), a high η could undermine the robustness of WaveKey against device spoofing and also increase the likelihood of a successful attack via gesture mimicking, as the ECC would tolerate a larger difference between the benign gesture and the mimicry gesture.

To optimally configure N_b, we evaluated 12 values ranging from 4 to 15 as N_b and selected the one that yielded the best security. With each evaluated value, we first obtained the corresponding η. Specifically, we measured the bit mismatch rates between key-seeds S_{M} and S_{R} using the 14,400 samples in dataset $\mathcal{D}$. For a desirable user experience, we designed WaveKey to maintain a high key-establishment success rate above 99%. To this end, we set the value of η to be higher than the bit mismatch rate for 99% of the samples. Based on this determined η, we calculated the success rate of random guess with Eq. (4) and evaluated the success rate of gesture mimicking through an experiment demonstrated in §VI-E. Fig. 7 presents the results, clearly indicating that $N_b = 9$ is the optimal setting, effectively thwarting both random guessing and gesture mimicking.

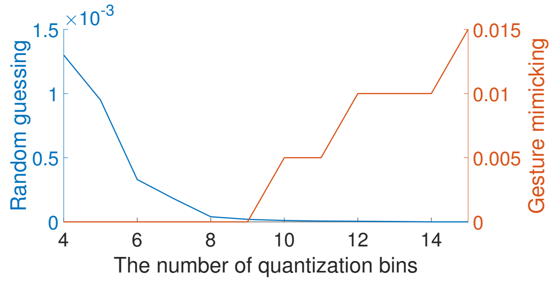


Fig. 7. Random-guessing and gesture-mimicking success rates.

3) τ: The parameter τ represents the allowable time delay for the two messages $\mathcal{M}_{\mathcal{A},\mathrm{M}}$ and $\mathcal{M}_{\mathcal{A},\mathrm{R}}$, and we determined its value by measuring the time consumed in preparing these messages. To do this, we used the four mobile devices and the laptop to generate the two messages from the 14,400 data records in $\mathcal{D}$, recording their respective time consumption for each data record. We found that all four mobile devices and the laptop were able to prepare their messages within 100 ms for every data record. Considering that the mobile device and the RFID server communicate directly through short-range communication technologies, the transmission delay is negligible. Based on these observations, we set τ as 120 ms, which provides sufficient time for the two benign devices to get their messages ready without compromising the efficiency of the key-establishment process. However, it is important to note that an adversary attempting to forge the two messages using advanced computer vision techniques would find it extremely challenging to meet this tight deadline.

D. Randomness Test

This section evaluates the randomness of the established key. Each of the six volunteers performed 200 random gestures for this evaluation. All the data were collected in a static environment which contributed negligible randomness to the key-establishment. We generated a 256-bit key from each gesture and pieced the 200 keys from the same volunteer together to a sequence of 51,200 bits. We refer to such a sequence as a key-chain. Since the key-seeds is also critical to the system security, we pieced the 200 key-seed pairs with the same volunteer into two sequences of 7,600 bits each, which are referred to as key-seed-chains. The run test suggested for randomness evaluation by NIST [39] was employed to evaluate these sequences. The average and minimum p-values for the six key-chains are 0.92 and 0.90, respectively. The average and minimum p-values for the 12 key-seed-chains are 0.78 and 0.72, respectively. A p-value of 0.05 is commonly adopted as a threshold for randomness text. Therefore, this result affirms the high randomness of the key produced by WaveKey.

E. Security Evaluation

As analyzed in §V, WaveKey demonstrates high resilience to eavesdropping, RFID signal spoofing, and Man-in-the-Middle (MitM) attacks. Therefore, this section is dedicated to evaluating the potential impact of device spoofing attacks, with a specific emphasis on those that leverage gesture mimicking and camera-aided data recovery.

1) Gesture mimicking: Our evaluation process proceeded as follows. The six volunteers acted as the victims for 20 key-establishment instances, performing 20 brief random gestures each. The remaining five volunteers were tasked with mimicking the victim's gestures while holding mobile devices for data collection. In total, we evaluated 600 mimicking instances. For each mimicking instance, we generated key-seeds from both the victim's and the adversary's IMU data and calculated their bit mismatch rate. A mimicking instance was considered successful if the bit mismatch rate was less than the ECC's error correction rate. In our experiment, all 600 mimicking instances failed in device spoofing, providing strong evidence of the robust security of WaveKey against this type of attack.

2) Camera-aided data recovery: We initially evaluated the remote recording strategy for the device spoofing attack as follows. A volunteer, designated as the victim, performed 200 random gestures for key establishment. During this process, we recorded the victim's gestures using an ALPCAM Webcam, which boasts a high frame rate of 260 FPS and a resolution of 1080p. This setup was chosen to emulate the optimal recording fidelity an attacker might achieve with a hidden camera. The camera was positioned to maintain a line-of-sight view of the victim's hand from a distance of three meters. Subsequently, we implemented Complexer-yolo [35], a sophisticated algorithm, to extract the 3D position of the user's hand from each captured video frame. Furthermore, we calculated the instantaneous hand velocity for each frame based on the change in hand position relative to the previous frame, from which we derived the linear acceleration of the mobile device. After deriving the key-seed, we proceeded with the key agreement phase. The attack was considered successful if a common key could be established. Out of the 200 attack instances, only 1 were successful, yielding a trivial success rate of 0.5%. In addition, practical implementation of this attack, which requires streaming high-resolution video to a remote, powerful server for computationally intensive image/video processing, introduces significant delays. These delays prevent the adversary from obtaining a valid key-seed within the required time window for subsequent key agreement, rendering the remote recording strategy infeasible for device spoofing against WaveKey.

We proceed to evaluate in-situ recording strategy for the device spoofing attack with a Google Pixel 8 acting as the attacking device. In particular, the same volunteer was designated as the victim and performed 200 random gestures for key establishment. The attacking device was positioned to maintain a line-of-sight view of the victim's hand from a distance of three meters. Due to the constrained computing resources, the attacking device was not able to perform the 3D hand trajectory estimation in real-time. Instead, we implemented YoloV5 [40] on the attacking device to extract the 2D location of the user hand and trained a neural network to estimate the linear acceleration of the mobile devices. Even with this carefully designed attacking strategy, none of the 200 attacking instances succeeded in obtaining a valid key-seed.

Our evaluation confirmed the high robustness of WaveKey against all kinds of device spoofing attacks.

F. Generality Evaluation

This section assess the generality of WaveKey by evaluating its performance across various experiment settings.

1) Environmental changes: Environmental changes can have a significant impact on wireless signal propagation, making it crucial to assess the performance of WaveKey in different environmental settings. While all the experiments were conducted in the same laboratory room, we varied the location and orientation of the RFID reader to emulate four distinct environments. WaveKey's performance was evaluated in each of these four environments under both static and dynamic conditions. In the static condition, only the volunteer performing the gesture for data collection was present in the laboratory room. On the other hand, in the dynamic condition, the other five volunteers walked around the RFID reader while one volunteer performed the gesture for data collection. In each environment setting, all six volunteers performed 50 random gestures, simulating 50 key-establishment instances. The key-establishment success rates, denoted as P_{k} in the different environment settings are shown in Tab. I, where the static and dynamic conditions are abbreviated as S and D, respectively. The results demonstrate that WaveKey's performance across different static environments was satisfactory and remained stable. However, in the dynamic environment, the key-establishment success rate dropped slightly. This is mainly due to the dynamic environment introducing additional variations to the RFID data, which may increase the bit mismatch rate between a pair of key-seeds. Overall, WaveKey proved to be robust against environmental changes.

TABLE I
KEY-ESTABLISHMENT SUCCESS RATES IN DIFFERENT ENVIRONMENTS.

Envr.	1		2		3		4	
Condition	S	D	S	D	S	D	S	D
P_{k}	99.7	99.0	100	98.6	99.7	99.0	99.3	99.0

2) User's relative position to RFID antenna: This experiment aimed to assess the impact of the user's position relative to the RFID antenna on key generation performance. A volunteer, designated as the user, conducted gestures for key establishment in both static and dynamic settings. Initially, with the azimuth angle set at 0 degrees, distances of 1, 3, 5, 7, and 9 meters from the user to the antenna were evaluated. Subsequently, at a constant distance of five meters, azimuth angles of -60, -30, 0, 30, and 60 degrees were assessed. The user executed 400 gestures for each configuration: 200 in a static environment and 200 in a dynamic environment. The corresponding success rates are presented in Tab. II. The distance had negligible impact on key-establishment success rate in static conditions. However, in dynamic environments, a larger distance notably decreased success rates since the channel variations resulting from environmental changes and multipath effect are more pronounced. The azimuth angle showed minimal influence on key generation in both environments.

TABLE II
KEY-ESTABLISHMENT SUCCESS RATES WITH DIFFERENT DEVICE SETTINGS.

Distance (meter)	1	3	5	7	9
Static condition	99.5%	100%	99.5%	100%	99.5%
Dynamic condition	99.5%	99.5%	99%	99%	99%
Angle (degree)	-60	-30	0	30	60
Static condition	100%	100%	99.5%	100%	99.5%
Dynamic condition	99.5%	99%	99%	98.5%	99%

3) Various mobile devices and RFID devices: This experiment assessed the impact of inherent hardware imperfections in the involved mobile and RFID devices on the key generation performance. In particular, we evaluated all the 24 distinct combinations derived from four mobile devices and nine RFID tags. For consistency, a single volunteer executed 200 gestures for each combination under the default experiment settings. The success rates corresponding to these device settings demonstrated remarkable consistency, ranging from a minimum of 99% to a maximum of 100%. These findings underscore the adaptability and reliability of our scheme across a wide array of devices.

G. Time Consumption

Lastly, we conducted an evaluation of the key-establishment time consumption, which measures the time taken from the start of the random gesture until a key is successfully established. As the message transmission delay is negligible, the key-establishment time consumption is calculated as the sum of the computational time consumption and the 2 seconds required to perform the gesture. To assess the performance for different key lengths, we measured the key-establishment time consumption for various key lengths, including 128 bits, 192 bits, and 256 bits for AES; 168 bits for 3DES; and 2048 bits for RC4. For each key length, we measured the average time consumption using the 14,400 data records in dataset $\mathcal{D}$. The results are listed in Tab. III.

TABLE III
TIME CONSUMPTION FOR DIFFERENT KEY LENGTHS.

Key length (bit)	128	168	192	256	2048
Time consumption (ms)	2345	2332	2347	2357	2362

Notably, the time consumption of key establishment did not vary significantly for different key lengths. Even when establishing a 2,048-bit key with WaveKey, the process took only around 2.4 seconds. This highlights WaveKey's high efficiency, especially in generating long keys, making it well-suited for practical applications that require secure and rapid key establishment.

VII. Conclusion

This paper presented the design and evaluation of WaveKey, an authenticated key-establishment scheme to enable secure mobile ad hoc in-situ access to RFID-protected systems. WaveKey explores a random gesture performed by the mobile user to induce correlated IMU data and RFID signals at the involved mobile device and RFID server, adopts deep learning techniques to extract the complex cross-modal correlation, and devises an OT-based key-negotiation protocol toward secure and efficient key establishment. Theoretical analysis and experimental evaluation confirmed the high security and efficiency of WaveKey.

Acknowledgement

This work was supported in part by the U.S. National Science Foundation under grants CNS-2325563 and CNS-2055751. Research was also sponsored in part by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-23-2-0225. Research was also supported in part by the STARs program of the University of Texas System under grant AR911847. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the University of Texas System, the Army Research Laboratory, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

[1] A. Filippeschi, N. Schmitz, M. Miezal, G. Bleser, E. Ruffaldi, and D. Stricker, "Survey of motion tracking methods based on inertial sensors: A focus on upper limb human motion," *Sensors*, vol. 17, no. 6, p. 1257, 2017.

[2] H. Wang, Y. Zhang, and X. Yu, "An overview of image caption generation methods," *Computational intelligence and neuroscience*, vol. 2020, 2020.

[3] H. Zhang, J. Koh, J. Baldridge, H. Lee, and Y. Yang, "Cross-modal contrastive learning for text-to-image generation," in *IEEE/CVF CVPR*, June 2021.

[4] R. Findling, M. Muaaz, D. Hintze, and R. Mayrhofer, "ShakeUnlock: Securely transfer authentication states between mobile devices," *IEEE Transactions on Mobile Computing*, vol. 16, no. 4, pp. 1163–1175, 2017.

[5] L. Zhang, D. Han, A. Li, T. Li, Y. Zhang, and Y. Zhang, "WristUnlock: secure and usable smartphone unlocking with wrist wearables," in *IEEE CNS*, Washington, DC, June 2019.

[6] A. Sayeed and A. Perrig, "Secure wireless communications: Secret keys through multipath," in *ICASSP*, Las Vegas, NV, March-April 2008.

[7] H. Liu, Y. Wang, J. Yang, and Y. Chen, "Fast and practical secret key extraction by exploiting channel response," in *IEEE INFOCOM*, Turin, Italy, April 2013.

[8] J. Zhang, A. Marshall, R. Woods, and T. Duong, "Efficient key generation by exploiting randomness from channel responses of individual ofdm subcarriers," *IEEE Transactions on Communications*, vol. 64, no. 6, pp. 2578–2588, 2016.

[9] W. Xu, Z. Li, W. Xue, X. Yu, B. Wei, J. Wang, C. Luo, W. Li, and A. Zomaya, "Inaudiblekey: Generic inaudible acoustic signal based key agreement protocol for mobile devices," in *ACM/IEEE IPSN*, San Antonio, TX, May 2021.

[10] P. Xie, J. Feng, Z. Cao, and J. Wang, "Genewave: Fast authentication and key agreement on commodity mobile devices," *IEEE/ACM Transactions on Networking*, vol. 26, no. 4, pp. 1688–1700, 2018.

[11] Y. Lu, F. Wu, S. Tang, L. Kong, and G. Chen, "Free: A fast and robust key extraction mechanism via inaudible acoustic signal," in *ACM MobiHoc*, Washington, DC, October 2019.

[12] S. Bellovin and M. Merritt, "Encrypted key exchange: Password-based protocols secure against dictionary attacks," 1992.

[13] J. Ding, S. Alsayigh, J. Lancrenon, S. Rv, and M. Snook, "Provably secure password authenticated key exchange based on rlwe for the post-quantum world," in *Cryptographers' Track at the RSA conference*. Springer, 2017.

[14] J. Hershey, A. Hassan, and R. Yarlagadda, "Unconventional cryptographic keying variable management," *IEEE Transactions on Communications*, vol. 43, no. 1, pp. 3–6, 1995.

[15] T. Dierks and E. Rescorla, "The transport layer security (tls) protocol version 1.2," Tech. Rep., 2008.

[16] G. Wang, H. Cai, C. Qian, J. Han, X. Li, H. Ding, and J. Zhao, "Towards replay-resilient rfid authentication," in *ACM MobiCom*, New Dehli, India, October-November 2018.

[17] X. Chen, J. Liu, X. Wang, H. Liu, D. Jiang, and L. Chen, "Eingerprint: Robust energy-related fingerprinting for passive rfid tags," in *NSDI*, Santa Clara, California, February 2020.

[18] J. Li, A. Li, D. Han, Y. Zhang, T. Li, and Y. Zhang, "RCID: Fingerprinting passive rfid tags via wideband backscatter," in *INFOCOM*, May 2022.

[19] A. Li, J. Li, Y. Zhang, D. Han, T. Li, and Y. Zhang, "Secure UHF RFID authentication with smart devices," *IEEE Transactions on Wireless Communications*, vol. 22, no. 7, pp. 4520–4533, July 2023.

[20] J. Li, C. Wang, A. Li, D. Han, Y. Zhang, J. Zuo, R. Zhang, L. Xie, and Y. Zhang, "Rhythmic rfid authentication," *IEEE/ACM Transactions on Networking*, vol. 31, no. 2, pp. 877–890, April 2023.

[21] L. Yang, P. Peng, F. Dang, C. Wang, X. Li, and Y. Liu, "Anti-counterfeiting via federated rfid tags' fingerprints and geometric relationships," in *IEEE INFOCOM*, Kowloon, Hong Kong, April 2015.

[22] C. Zhao, Z. Li, T. Liu, H. Ding, J. Han, W. Xi, and R. Gui, "Rf-mehndi: A fingertip profiled rf identifier," in *IEEE INFOCOM*, Paris, France, April 2019.

[23] T. Li, W. Luo, Z. Mo, and S. Chen, "Privacy-preserving rfid authentication based on cryptographical encoding," in *IEEE INFOCOM*, Orlando, FL, May 2012.

[24] L. Yang, Q. Lin, C. Duan, and Z. An, "Analog on-tag hashing: Towards selective reading as hash primitives in gen2 rfid systems," in *ACM MobiCom*, Snowbird, UT, October 2017.

[25] H. Lee and S. Jung, "Gyro sensor drift compensation by kalman filter to control a mobile inverted pendulum robot system," in *IEEE International Conference on Industrial Technology*. IEEE, 2009, pp. 1–6.

[26] A. Savitzky and M. Golay, "Smoothing and differentiation of data by simplified least squares procedures." *Analytical chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.

[27] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, "Radio-telepathy: Extracting a secret key from an unauthenticated wireless channel," in *ACM MobiCom*, San Francisco, California, September 2008.

[28] R. Doran, "The gray code," Department of Computer Science, The University of Auckland, New Zealand, Tech. Rep., 2007.

[29] J. Kilian, "Founding crytpography on oblivious transfer," in *Proceedings of the twentieth annual ACM symposium on Theory of computing*, 1988, pp. 20–31.

[30] T. Chou and C. Orlandi, "The simplest protocol for oblivious transfer," in *International Conference on Cryptology and Information Security in Latin America*. Springer, 2015, pp. 40–58.

[31] H. Chen, F. Li, and Y. Wang, "Echotrack: Acoustic device-free hand tracking on smart phones," in *IEEE INFOCOM*, Atlanta, GA, May 2017.

[32] H. Jiang, M. Wang, D. Liu, and S. Zhou, "Ctrack: Acoustic device-free and collaborative hands motion tracking on smartphones," *IEEE Internet of Things Journal*, vol. 8, no. 19, pp. 14 658–14 671, 2021.

[33] S. Regani, C. Wu, B. Wang, M. Wu, and K. Liu, "mmwrite: Passive handwriting tracking using a single millimeter-wave radio," *IEEE Internet of Things Journal*, vol. 8, no. 17, pp. 13 291–13 305, 2021.

[34] Z. Zhang, X. Wang, D. Huang, X. Fang, M. Zhou, and Y. Zhang, "Mrpt: Millimeter-wave radar-based pedestrian trajectory tracking for autonomous urban driving," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–17, 2021.

[35] M. Simon, K. Amende, A. Kraus, J. Honer, T. Samann, H. Kaulbersch, S. Milz, and H. Michael, "Complexer-yolo: Real-time 3d object detection and tracking on semantic point clouds," in *IEEE/CVF CVPR*, Long Beach, CA, June 2019.

[36] B. Tang, X. Shen, Y. Hu, and Q. Fan, "A robust real-time hand detection and tracking," in *ICVRV*, Zhengzhou, China, October 2017.

[37] N. Muller, Y.-S. Wong, N. J. Mitra, A. Dai, and M. Nießner, "Seeing behind objects for 3d multi-object tracking in rgb-d sequences," in *IEEE/CVF CVPR*, June 2021.

[38] "PyTorch," 2004, https://pytorch.org/.

[39] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, and E. Barker, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," 2001, https://csrc.nist.gov/publications/detail/sp/800-22/rev-1a/final.

[40] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *IEEE/CVF CVPR*, 2016.