Echoes of Fingertip: Unveiling POS Terminal Passwords Through Wi-Fi Beamforming Feedback

Siyu Chen, *Student Member, IEEE*, Hongbo Jiang, *Senior Member, IEEE*, Jingyang Hu, *Student Member, IEEE*, Tianyue Zheng, *Member, IEEE*, Mengyuan Wang, *Member, IEEE*, Zhu Xiao, *Senior Member, IEEE*, Daibo Liu, *Member, IEEE*, and Jun Luo, *Fellow, IEEE*

***Abstract*—Recent years, point-of-sale (POS) terminals are no longer limited to wired connections, with many relying on Wi-Fi for data transmission. Although Wi-Fi offers the convenience of wireless connectivity, it introduces significant security vulnerabilities. This work presents a non-intrusive method for eavesdropping POS passwords via Wi-Fi sensing, named BeamThief. Instead of conventional Wi-Fi Channel State Information (CSI) readings, our approach employs Wi-Fi Beamforming Feedback Information (BFI) for an eavesdropping attack. Compared to CSI, which can only be extracted through intruding into the Access Point (AP) or from a limited selection of commercial Wi-Fi cards (e.g., Intel-5300), BFI readings can be more readily obtained from a broad array of commercial Wi-Fi devices. A key technological contribution of BeamThief is the development of an analysis model for predicting finger motion trajectories. This model is based on the physical relationship between BFI readings and finger motion, thus eliminating the need for extensive labeled training data. Furthermore, we employ Maximum Ratio Combining (MRC) to enhance the BFI series, ensuring performance across various scenarios. We implement BeamThief using everyday commercial Wi-Fi devices and conduct a series of experiments to assess the impact of this attack. Experimental results demonstrate that BeamThief achieves an accuracy rate 79% in inferring 6-digit POS passwords within the top-100 attempts.**

Received 26 April 2024; revised 1 August 2024; accepted 12 September 2024. Date of publication 23 September 2024; date of current version 9 January 2025. The work of Hongbo Jiang was supported in part by the National Key Research and Development Program of China under Grant 2022YFE0137700, in part by the National Science Foundation of China (NSFC) under Grant U20A20181 and Grant 62372161, in part by the Science and Technology Innovation Program of Hunan Province under Grant 2021RC4023, and in part by the Key Research and Development Program of Hunan Province under Grant 2021WK2001. The work of Zhu Xiao was supported in part by the Natural Science Foundation Project of Chongqing, Chongqing Science and Technology Commission under Grant CSTB2024NSCQ-MSX0920, and in part by the Key R&D Program of Hunan Province under Grant 2024AQ2032. The work of Daibo Liu was supported in part by NSFC under Grant 62372166, and in part by the Hunan Provincial Natural Science Foundation of China under Grant 2023JJ30164. Recommended for acceptance by W. Wang. *(Corresponding authors: Hongbo Jiang; Zhu Xiao; Daibo Liu.)*

Siyu Chen, Hongbo Jiang, Jingyang Hu, Mengyuan Wang, and Daibo Liu are with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China (e-mail: csy990406@hnu.edu.cn; hongbojiang2004@gmail.com; fbhhjy@hnu.edu.cn; wmy1997@hnu.edu.cn; dbliu@hnu.edu.cn).

Tianyue Zheng is with the School of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: zhengty@sustech.edu.cn).

Zhu Xiao is with the Chongqing Research Institute, College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China (e-mail: zhxiao@hnu.edu.cn).

Jun Luo is with the School of Computer Science and Engineering, Nanyang Technological University,, Singapore 639798 (e-mail: junluo@ntu.edu.sg).

Digital Object Identifier 10.1109/TMC.2024.3465564

***Index Terms*—Beamforming feedback information, keystroke inference, POS terminals, password eavesdropping, Wi-Fi sensing.**

I. Introduction

POS terminals are digital devices employed to process card payments in banks, retail stores, museums, metro stations, pharmacies and restaurants. According to the Nilson Report [1] released in October 2022, the global supply of POS terminals reached 136 million units in 2021. Due to their popularity, various eavesdropping attacks aim to steal users' bank card passwords through POS machines. Traditional attack methods rely on pre-installed malicious software that can access the readings of motion sensors to obtain password information [2], [3], [4]. However, these types of attacks are difficult to carry out and can be easily defended against by anti-malware software.

Current research aims to develop non-invasive side-channel eavesdropping attacks that are more covert. Previous studies have focused on side-channel attacks using acoustics [5], [6], [7], [8], [9], [10], [11], [12], video recordings [13], and radio frequency (RF) [14], [15], [16], [17], [18], [19], [20], [21], [22], [23]. However, most of the above methods rely on impractical assumptions, which limit their deployment in real-world scenarios. For instance, acoustics-based eavesdropping is limited by the physical properties of sound signal, and can only be executed in environments with weak interference and within close proximity (no more than 90 cm) to the target for launching an attack. Vision-based attacks requires unobstructed line-of-sight (LOS), making it challenging to deploy flexibly. Additionally, the ambient brightness significantly impacts the success rate of vision-based attacks. Attacks based on electromagnetic radiation (EMR) are only effective against touchscreen smartphones and are difficult to migrate to POS terminals with physical buttons. Methods based on Wi-Fi Channel State Information (CSI) contain fine-grained information about keystrokes. However, one critical issue hindering the wide application of CSI-based attacks is that CSI can only be extracted from few commodity Wi-Fi cards through driver hacking.

In this paper, we propose utilizing Beamforming Feedback Information (BFI), which is compliant with the latest IEEE 802.11ac protocol [24], as a side-channel for eavesdropping and decrypting passwords entered via the keyboard of a POS terminal. As illustrated in Fig. 1, to enable Multi-User Multiple Input Multiple Output (MU-MIMO), the transmitting antenna must adjust the phase and amplitude of the signal to focus it at

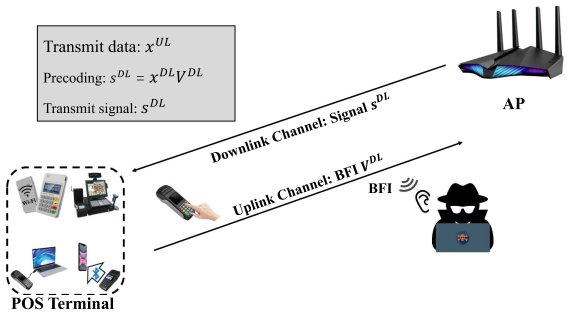


Fig. 1. Scenario of BeamThief: Keystroke actions can have an impact on the downlink channel, and POS terminal transmits channel information back to the AP through BFI. The transmission of clear-text BFI provides an opportunity for eavesdropping.

the receiver's end, thereby enhancing the signal's transmission performance. Consequently, the Access Point (AP) requires knowledge of the downlink (AP to POS terminal) channel state information, which is encapsulated and compressed into BFI by the POS terminal and transmitted back to the AP via the uplink (POS terminal to AP). During this process, the typing activity of a victim can alter the channel state, thereby enabling eavesdropping through BFI, akin to CSI. There are two obvious advantages of leveraging BFI for eavesdropping attack.

- BFI is transmitted unencrypted, facilitating easier interception.
- BFI adheres to the IEEE 802.11 ac standard [24], allowing for extraction from all new-generation Wi-Fi devices that support MU-MIMO, without the need for intrusion or specialized hardware. This enables the easy execution of eavesdropping attacks using readily available commercial devices.

Though promising, translating this idea into a practical system entails multiple challenges. First of all, subcarrier diversity is attributed to frequency-selective fading, a well-known phenomenon in wireless communications. Due to variations in path loss, fading, and interference, different subcarriers experience distinct channel conditions, leading some subcarriers to primarily capture keystroke actions while others predominantly capture noise. How to judiciously select and optimize the subcarriers to achieve optimal perception of keystroke actions in the BFI series becomes a challenge. To address this challenge, we employ Maximum Ratio Combining (MRC) to maximize the Signal-to-Noise Ratio (SNR) of the BFI series. Collectively, to effectively combine these subcarriers, MRC applies complex-weighted averaging based on fading coefficients for each received subcarrier, assigning higher weights to subcarriers with favorable channel conditions and lower weights to those with poorer channel conditions. This emphasis on stronger components and suppression of weaker components contributes to SNR maximization. By leveraging MRC, we optimize the BFI series, ensuring it captures crucial information while minimizing the impact of noise and fading effects.

The second challenge lies in establishing the correlation between the captured BFI series and keystrokes. A straightforward approach involves training a model to correlate different keys with their respective BFI series waveforms. Sufficient data collection for model training allows for inputting a BFI series to determine the most likely key. Indeed, this method has been utilized in most existing inference attacks based on acoustics and radio signals [16], [17], [25]. However, acquiring a sufficiently large dataset is challenging due to variations in typing behavior among individuals. In real-world scenarios, obtaining an adequate number of training samples from a target victim to mitigate overfitting is difficult. To circumvent the impact of insufficient data on attack deployment, our aim is to develop a model that describes the physical relationship between BFI and keystrokes. Although existing method [26] have been able to establish the connection between CSI readings and their respective actions, BFI, as compressed information of CSI, cannot directly apply to existing CSI-based sensing models and algorithms. For ease of analysis, we divide the continuous BFI series of entering an entire 6-digit PINs into several segments, each associated with a key. By studying the transmission principles of BFI, we first derive a closed expression between the BFI series values and the distance of finger swipes. To refine this expression to directly reflect specific keystrokes, we further estimate the speed and direction of finger movement required when typing a pair of keys. Combining these parameters, we establish a model for the trajectory of finger movements during password entry, demonstrating the distance of finger movement in horizontal and vertical directions on the screen for entering a pair of keys. After such projection and transformation, we establish a clear relationship between the BFI series and finger movements. Meanwhile, we note that different key pairs may share identical finger motion trajectories. To alleviate inference ambiguity, we propose exploring the interdependencies between consecutive key pairs to narrow down the possible number of keystrokes. We model the entire PIN entry process as a Hidden Markov Model (HMM), treating the recovered finger motion trajectories as observations and the precise key pairs as hidden states. Finally, the HMM outputs a list of PINs, ranked based on their likelihood of being the target PIN.

The main contributions of this paper are highlighted as follows.

- BeamThief's analysis of BFI reveals its potential as a powerful side-channel attack for eavesdropping on passwords input through POS machine keypads (next we will refer to it as POS machine passwords), outperforming alternative techniques.
- BeamThief intelligently selects and optimizes subcarriers using MRC to enhance the perceptual performance of BFI, thereby increasing the success rate of eavesdropping attacks.
- By quantitatively analyzing the channel variations caused by finger keystroke movements, a functional relationship between BFI series and finger movements is established. This relationship enables the execution of keystroke inference tasks effectively without the need for training.
- We develop a prototype and demonstrated the severity of the threat. It surpasses the state-of-the-art inference attacks in terms of setup practicality, with far fewer deployment constraints. In extensive evaluations, the average top-100 keystroke inference accuracy of BeamThief is 79% .

II. Related Work

A. Vision-Based Method

Attackers can employ computer vision techniques to record and retrieve keystrokes secretly. Early vision-based keystroke inference attacks relied on directly observing the contents displayed on the screen [27]. To enhance practicality, subsequent research has explored side-channel visual cues involving the analysis of physical appearance changes in the device, such as shadows [28] and backside motions of tablets [29]. Recent studies have further shown that even video capturing of eye movements [13] or during video calls can leak keystrokes [30]. While these methods excel in achieving high accuracy in password theft, visual-based approaches pose considerable demands in attack scenarios. Firstly, they necessitate specific devices for gathering video information or unauthorized access to the victim's phone for data collection. Secondly, the attack can only be initiated under Line of Sight (LOS) conditions. Lastly, variations in lighting conditions can substantially impede the effectiveness of such methods.

B. Sensor-Based Method

Most sensor-based methods involve supervised training to establish the correlation between each keystroke and the corresponding sensing signal. For instance, smartphones' accelerometers can capture keyboard vibrations for both physical [31] and on-screen [32] keyboards, enabling keystroke inference. Additionally, smartwatches' accelerometers or gyroscopes can track hand movements during typing [33]. Some approaches, like collecting acoustic emanations of keystrokes through Voice-over-IP (VoIP) calls [34], require tricking the victim into installing malware on the smartwatch. Others [35], [36] leverage Time Difference of Arrival (TDoA) values for keystroke localization but have drawbacks, necessitating multiple synchronized microphones, proximity to the target keyboard, and pre-infecting the victim's phone with malware for intercepted acoustic signal transmission.

C. Acoustic-Based Method

The KeyListener [5] technique achieves keystroke inference by leveraging sound signal attenuation for keystroke localization. On the other hand, PatternListener [6] measures the relative motion of fingertips using acoustic signals reflected from them to infer pattern lines. Acoustic side-channel attacks are well-suited for password theft. However, current methods utilizing acoustic side-channels struggle to eliminate environmental influences, greatly impacting their performance. Additionally, due to acoustic properties, attack devices need to be in close proximity to the target, posing deployment challenges for attackers.

D. RF-Based Method

Recent research proves the effectiveness of RF signals for keystroke inference. RF-based techniques offer three key advantages over other side-channel attacks: they are ubiquitous and invisible, non-intrusive, and do not require close proximity to the victim [14], [16], [17], [19], [25], [37], [38], [39], [40], [41]. A recent study [17], [42] estimated finger movement trajectories by analyzing electromagnetic leakage from touchscreen interactions to infer passwords, but it may not be easily applicable to POS machines. Additionally, Wi-Fi-based keystroke inference has gained popularity in recent years. [14], [16], [19], [43] capture environment changes caused by key presses on keyboards through Channel State Information (CSI) and use analysis to achieve keystroke inference. However, CSI is hampered by its difficulty in collection and its applicability, posing challenges to the deployment of attacks. Therefore, BFI, which complies with the latest protocols and propagates in plaintext, holds greater potential in password theft scenarios, enabling more flexible handling of various situations. Additionally, many current password theft techniques rely on neural networks, but acquiring sufficient training data from real-world scenarios is challenging, making it difficult to mitigate overfitting. Starting from the principle of how human body movements affect BFI transformations, BeamThief avoids the use of neural networks, eliminating the need for a large number of training samples.

III. Attack Scenario and Preliminary

This section begins with an overview of the attack scenario involving eavesdropping password. We then delve into the fundamental principles underlying BFI technology. Subsequently, we conduct a comprehensive analysis of the feasibility of BFI as a tool for executing eavesdropping attacks. Finally, we present concrete evidence demonstrating how BFI can offer clear advantages over traditional CSI methods when utilized in the context of implementing various types of attacks.

A. Attack Scenario

In this section, we present a hack-free scenario for eavesdropping on POS machine passwords. The attacker can accomplish this eavesdropping attack without any pre-deployed devices or intrusion into any systems. Specifically, the existing POS Terminals include several types, as shown in Fig. 1, including integrated POS machines, network-enabled POS machines, POS machines connected to computers, and POS machines connected to mobile phones. Usually, all four types of POS Terminals have Wi-Fi capability. Our scenario is applicable to these four types of POS Terminals: POS Terminals are connected to nearby AP via Wi-Fi. Typically, the distance between the Wi-Fi antennas of POS terminal and POS machine's keypad is no greater than 30 cm. Additionally, there exists some background network programs (e.g. Square POS and Clover POS [44], [45]) to facilitate the required traffic transmission for the experiment. Notably, this assumption is not uncommon in practice, as 42 out of 50 surveyed stores met this particular scenario. To execute the attack, the perpetrator only requires a device with a NIC and knowledge of the AP's transmission channel. No special positioning or additional equipment deployment is necessary. When the victim enters their password while making a payment through the POS machine, the attacker can successfully launch

the attack by capturing the BFI transmitted from the POS terminal to the AP. Two distinct scenarios enable the initiation of the attack:

Visual Observation: In the first scenario, the attacker begins the attack by visually observing the POS machine. Once the victim starts entering key information on the POS machine, the attacker captures the BFI transmitted by the nearby device, facilitating the inference of the key entries.

Wi-Fi Data Packet Analysis: The second scenario involves the attacker leveraging a Wi-Fi data packet analyzer [46]. Upon detecting wireless signals transmitted during typing activities on the POS machine, the attacker utilizes the intercepted BFI to initiate the attack. We only consider the keyboard layout that POS machines with a compact numerical keypad layout, featuring palm-sized keys with the numbers '1', '2', and '3' situated at the top. For the purpose of our analysis, we assume that the victim's key entry actions solely involve pressing the keys without any additional body or limb movements. We will introduce this method in more detail in the Section V-D.

Additionally, we account for two key entry modes adopted by the victims:

Flat Surface Typing: Victims perform key entry using a single hand while the POS machine remains stationary on a flat surface.

Handheld Typing: Victims hold the POS machine with one hand while using the other hand for key entry.

B. The Principle of BFI

Beamforming Feedback Information (BFI) enables MU-MIMO (Multi-User Multiple-Input Multiple-Output) communication in the 802.11ac/ax standard. With the BFI feedback, the AP has the capability to adjust the complex weight of the transmitted signal at each antenna, augmenting the reception of signals at the nodes. As stipulated in [47], the AP initiates the process by dispatching Null Data Packets (NDP) to all POS terminals participating in the transmission. As part of the standard receiver operations, each POS terminal calculates the channel estimate using the long training symbols in the High Throughput Long Training Fields (HT-LTFs). HT-LTFs are modulated - as the data fields - through orthogonal frequency-division multiplexing (OFDM) by dividing the signal bandwidth into *T* partially overlapping and orthogonal sub-channels spaced by 1/*T*. The input bits are grouped into OFDM symbols, $x = [x_{-K/2}, \ldots, x_{K/2-1}]$, where each element x_K is one OFDM sample. These *K* OFDM samples are digitally modulated and transmitted through the *K* OFDM sub-channels in a parallel fashion. For each *K*-th sub-channel, POS terminal receives a CSI matrix $\mathbf{H}_k \in \mathbb{C}^{M\times N}$, where M is the number of receiving antennas at the POS terminal and N is the number of transmitting antennas at the AP. Upon receiving $\mathbf{H}_k$, POS terminal compresses it instead of directly feeding it back to AP. POS terminal first performs a Singular Value Decomposition (SVD) on $\mathbf{H}_k$ as follows:

$$\mathbf{H}_k = \mathbf{U}_k \mathbf{S}_k \mathbf{V}_k^{\dagger}, \tag{1}$$

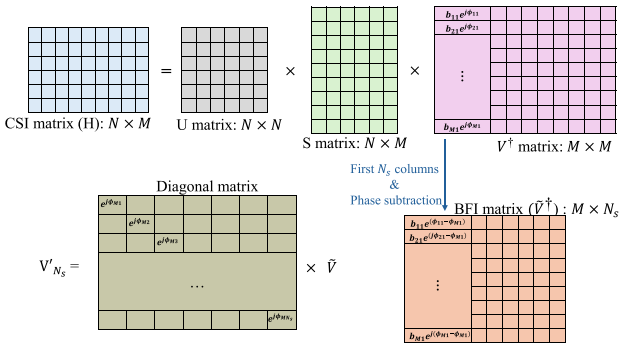


Fig. 2. The process of compressing CSI into BFI at the STA.

where $\mathbf{U}_k \in \mathbb{C}^{M\times M}$ and $\mathbf{V}_k \in \mathbb{C}^{N\times N}$ are unitary matrices, $\mathbf{S}_k \in \mathbb{C}^{M\times N}$ is a diagonal matrix of singular, and $\mathbf{V}_k^{\dagger}$ is the Hermitian (complex conjugate transpose) of $\mathbf{V}_k$.

Subsequently, as shown in Fig. 2, the POS terminal will compress the $\mathbf{V}_k$ matrix, specifically, it will perform the following operations:

- Step 1: POS terminal will extract the first N_s singular vectors from the $\mathbf{V}_k$ to get a new matrix $\mathbf{V}_{k,N_s}$, where N_s is the number of spatial streams which is no more than $\min(N, M)$.
- Step 2: The matrix $\mathbf{V}_{k,N_s}$ necessitates pre-processing to ensure that all elements are transformed into non-negative real numbers, thereby satisfying the requirements of Givens rotation [48].
- Step 3: Deriving a diagonal matrix

$$\mathbf{D}_k = \text{diag}(I_{i-1}, e^{j\phi_{ii}}, \ldots, e^{j\phi_{(M-1)i}}, 1) \tag{2}$$

  from $\mathbf{V}_{k,N_s}$ to obtain $\mathbf{V}'_{k,N_s}$.
- Step 4: Since the last element in $\mathbf{D}_k$ is 1, this results in no alteration to the last column of $\mathbf{V}'_{k,N_s}$ compared to $\mathbf{V}_{k,N_s}$ after the step 3. Hence, performing phase subtraction operation: subtracting the $\phi_{M,i}$ from $\phi_{j,i}$, where the $\phi_{j,i}$ is the j-th row and i-th column in $\mathbf{V}'_{k,N_s}$, and the $\phi_{M,i}$ is the phase of the last element in j-th row.
- Step 5: The resultant matrix $\tilde{\mathbf{V}}_k$ (BFI) obtained after phase subtraction can be viewed as a compression of the matrix $\mathbf{V}_k$. And there is a relationship between $\mathbf{V}'_{k,N_s}$ and $\tilde{\mathbf{V}}_k$ as

$$\mathbf{V}'_{k,N_s} = \text{diag}(e^{j\phi_{M1}}, e^{j\phi_{M2}} \ldots e^{j\phi_{MN_s}})\tilde{\mathbf{V}}_k. \tag{3}$$

- Step 6: POS terminal utilizes Givens rotations operation to reducing the number of bits required for beamforming feedback and thereby enhancing transmission efficiency.

The resulting BFI can be expressed as:

$$\tilde{\mathbf{V}_k} = \frac{a(f,t)}{\sigma} e^{j(\phi(f,t) - \Delta\phi(f,t))}, \tag{4}$$

where σ is the scaling factor for the amplitude of BFI relative to CSI, and $\Delta\phi$ represents the phase shift between BFI and CSI. Thus, the BFI after Givens rotation is transmitted back to the AP via the uplink. Since the BFI also contains channel state, it can be utilized for sensing similar to CSI. However, due to the absence of phase subtraction details, the AP can only reconstruct

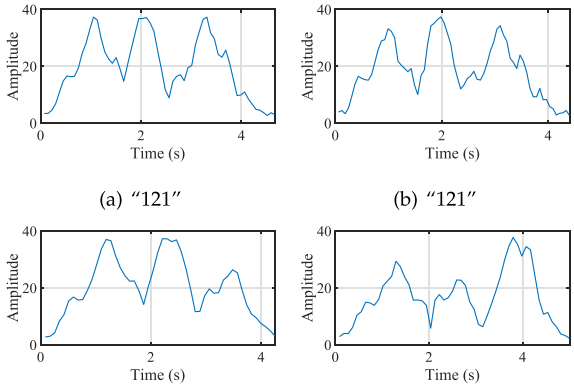


Fig. 3. BFI series of 4 different key combinations: The same subject types "123", "134", and "121" twice.

the $\tilde{\mathbf{V}}_K$ matrix but not the CSI, traditional sensing models and algorithms based on CSI cannot be directly applied to BFI. The good news is that in communications, BFI merely needs to provide the antenna weights for beamforming. This means that to reduce high transmission overhead, BFI is transmitted unencrypted [24], making it susceptible to eavesdropping by any third party in the environment. In contrast, existing CSI-based sensing methods require intrusion or specialized hardware, making BFI-based approaches undoubtedly more flexible in attack scenarios.

C. The Feasibility of BFI Attacks

To investigate the feasibility of eavesdropping attacks using BFI, we conduct a series of experiments. The same subject performed the following keystrokes: "123", "134", and "121" twice, and the results are shown in Fig. 3. Firstly, we observe that the BFI corresponding to the two occurrences of "121" exhibited similar waveforms. However, due to differences in keystroke speeds, there are distinct transitions between "1-2" and "2-1". Additionally, we notice that the first '1' and the second '1' in "121" also display waveform similarities. However, due to differences in keystroke habits, the direct typing of '1' and typing '2' followed by '1' causes variations. Similar examples are observed with the '3' in "123" and "134", where the two '3' keys show similar BFI series waveforms, but slight differences arose due to the variations in keystroke habits during the transitions. The similarities observed in Fig. 3 support the feasibility of keystroke inference using BFI.

IV. Design

A. Overview

Shown as Fig. 6, BeamThief is designed for conducting eavesdropping attacks on individuals using POS payments by implementing keystroke inference. To achieve this, BeamThief leverages MRC for dynamic subcarrier recognition and selects an optimal subset for efficient combination, thereby enhancing the BFI series. Then, the BFI series representing

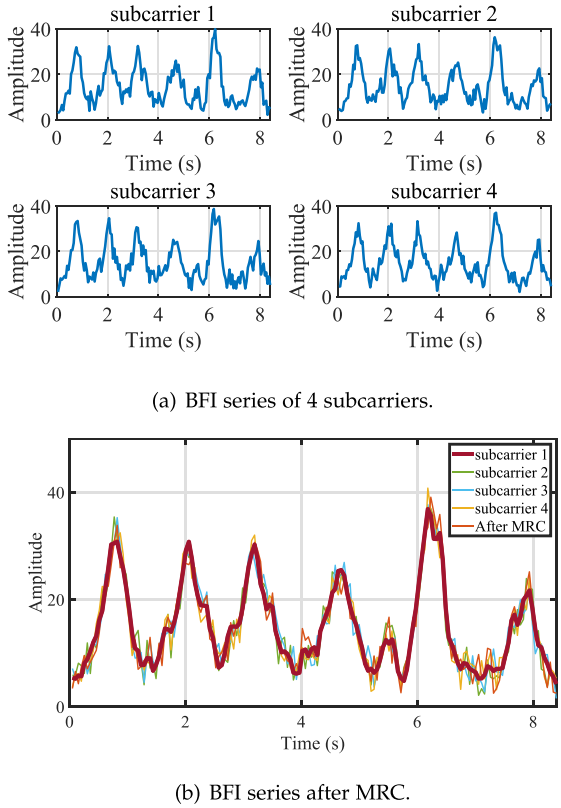


Fig. 4. The BFI time series: (a) The BFI series on different subcarriers. (b) BFI series on subcarriers and after MRC.

the complete PIN code is segmented into five key pairs using threshold segmentation. BeamThief analyzes BFI and finger tapping trajectories to obtain parameters of movement direction and speed, estimating possible key pairs. Finally, by modeling the key pair sequence using HMM, the highest probability PIN code is obtained.

B. Signal Preprocessing

1) BFI Series Enhancement: In our hypothetical attack scenario, where there is a distance of approximately one meter between the device and the POS machine, the perceptibility of keystroke actions by BFI may be slightly diminished. This issue is further exacerbated when there are additional interferences present. To enhance the perceptual performance of BFI under these circumstances, we propose the BFI series enhancement utilizing subcarrier diversity.

Subcarrier diversity is a well-known technique employed in wireless communication systems to mitigate the effects of frequency-selective fading. In this technique, the data intended for transmission is subdivided into multiple subcarriers, each experiencing unique fading conditions due to their distinct frequencies. By combining the received signals from these diverse subcarriers at the receiver, the overall signal quality and reliability can be significantly improved. Fig. 4(a) illustrates an example of keystrokes, where certain subcarriers capture the primary keystroke actions while others predominantly observe

noise. Furthermore, due to the complex nature of multipath propagation, the selection of the most sensitive subcarriers may vary randomly over time. Consequently, it becomes crucial to dynamically identify the optimal subset of subcarriers and effectively combine them to maximize the SNR.

Given that contemporary wireless devices and access points commonly employ multi-antenna systems, we utilize MRC to perform a weighted summation of the multiple subcarriers. Since the noise terms on different subcarriers are statistically independent, we can maximize the SNR by MRC as: $\hat{y}(\tau) = \sum_{f\in F} w(f)\tilde{y}(f,\tau)$, where *w(f)* denotes the normalized weight for combining subcarrier $f(\sum_{f\in F} w(f) = 1)$, and *F* is the set of all subcarriers. Here we use the maximum combining principle and set the weight coefficients as the conjugate of the channel estimation values at each receiving antenna. Specifically, for each receiving antenna *f*, the weight coefficient *w(f)* can be calculated as:

$$w(f) = conj(f)/\sum_{i\in F} conj(h_i). \tag{5}$$

It should be noted that certain commonly used intuitive criteria, such as average or variance of amplitude, cannot be considered as optimal weights for MRC. This is because subcarriers with higher average or amplitude deviation may not necessarily provide better capture of the sensing signal. As shown in Fig. 4(b), this method optimizes the received signal-to-noise ratio (SNR), allowing the BFI to better perceive keystroke actions.

2) Key Pair Segmentation: In this section, we employ a thresholding method to extract the peaks of the BFI series to achieve key pair segmentation. Specifically, we discretize the entire BFI series values to construct a histogram, where the number of bins equals the square root of the number of BFI points. Subsequently, we utilize the Otsu [49] method to calculate the between-class variance for all possible thresholds and select the threshold that maximizes this variance to divide the histogram into two parts (peaks and non-peaks). Finally, this threshold is used to segment the peaks within the BFI series. It is important to note that since our goal is to segment out key pairs, each segmentation must span two peaks.

C. Keystroke Inference

1) Trajectory Analysis: In an $M \times N$ transceiver system, wireless channel propagation is described as $Y = H \times X + noise$, where X represents the transmitted signal, Y the received signal, and H the Channel State Information (CSI) matrix. And the signal between a pair of transmitter and receiver antennas in the communication system is divided into static components $h_s(f)$ and dynamic components $h_d(f,t)$ along its propagation path. As shown in Fig. 5, the CSI is composed of these two components combined:

$$h(f,t) = h_s(f) + h_d(f,t) = h_s(f) + A_d e^{-2j\pi f d(t)/c}, \tag{6}$$

where A_d denotes the signal amplitude attenuation, and d(t) represents the length of the dynamic path. Alternatively the CSI in an ideal environment can be represented through amplitude

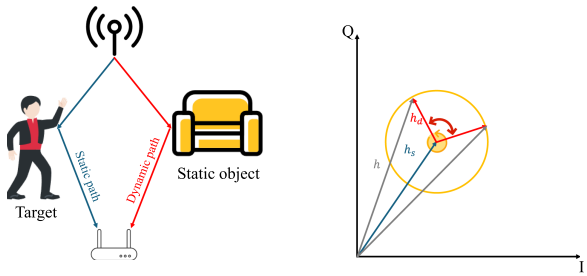


Fig. 5. CSI can be decomposed into static and dynamic components.

$a(f,t)$ and phase $\psi(f,t)$: $h(f,t) = a(f,t)e^{j\psi(f,t)}$. In a real-world scenario, due to phase offset $\Delta\psi(f,t)$ affecting a pair of transmitting and receiving antennas, the channel state needs to be adjusted to

$$\hat{h}(f,t) = a(f,t)e^{j(\psi(f,t)+\Delta\psi(f,t))}. \tag{7}$$

As two antennas at the AP share the same phase offset, the (m_1, m_2) element in $\hat{\mathbf{H}}^{\dagger}\hat{\mathbf{H}}$ can be written as:

$$\left[\hat{\mathbf{H}}^{\dagger}\hat{\mathbf{H}}\right]_{m_1,m_2} = \sum_{n=1}^{N} a_{n,m_1} a_{n,m_2} e^{j(\psi_{n,m_2}-\psi_{n,m_1})}, \tag{8}$$

where $(m_1, m_2) = 1, 2, \ldots, M$ and (m_1, m_2) are the antenna index at the AP while n is the antenna index at the POS terminal. On the other hand, from Section III-B through SVD, we understand that

$$\hat{\mathbf{H}}^{\dagger}\hat{\mathbf{H}} = \mathbf{V}\mathbf{S}^{\dagger}\mathbf{U}^{\dagger}\mathbf{U}\mathbf{S}\mathbf{V}^{\dagger} = \mathbf{V}\mathbf{S}^{\dagger}\mathbf{S}\mathbf{V}^{\dagger}. \tag{9}$$

Consequently, the (m_1, m_2) element in $\hat{\mathbf{H}}^{\dagger}\hat{\mathbf{H}}$ can be written as:

$$\left[\hat{\mathbf{H}}^{\dagger}\hat{\mathbf{H}}\right]_{m_1,m_2} = \sum_{n=1}^{N} \sigma_n^2 b_{m_1,n} b_{m_2,n} e^{j(\phi_{m_2,n}-\phi_{m_1,n})}, \tag{10}$$

where $b_{m,n}$ and $\phi_{m,n}$ are the amplitude and phase of (m,n) element in $\mathbf{V}$, σ_n is the (n,n) element in the diagonal matrix $\mathbf{S}$. Upon solving (8) and (10) simultaneously, we obtain

$$\sigma_n^2 b_{m_1,n} b_{m_2,n} e^{j(\phi_{m_2,n}-\phi_{m_1,n})} = a_{n,m_1} a_{n,m_2} e^{j(\psi_{n,m_2}-\psi_{n,m_1})}. \tag{11}$$

Subsequently, our analysis will focus on the same receiving antenna, denoted as $m = m_1 = m_2$. (11) can then be expressed as: $\sigma_n^2 b_{m,n}^2 = a_{n,m}^2$. Then we can derive the amplitude in $\mathbf{V}$:

$$b_{m,n} = \frac{a_{n,m}}{\sigma_n}. \tag{12}$$

As $\mathbf{V}$ is a unitary matrix, we have

$$\sum_{m-1}^{M} b_{m,n}^2 = 1. \tag{13}$$

By combining (13) with (12), we can further deduce the (n,n) element of $\mathbf{S}$:

$$\sigma_n^2 = \sum_{m=1}^{M} a_{n,m}^2. \tag{14}$$

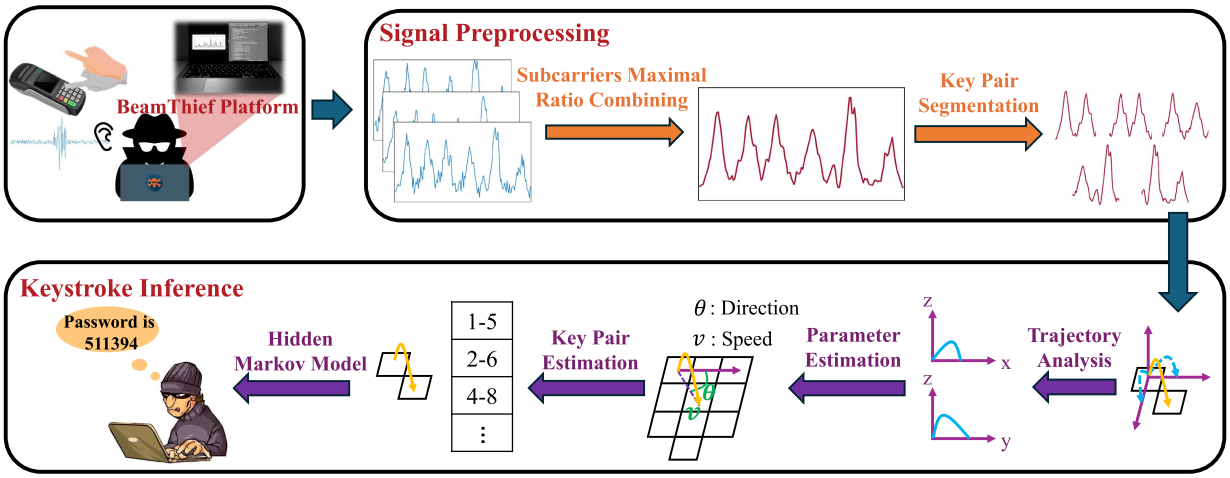


Fig. 6. Overview of BeamThief.

When we focus on the different receiving antennas, the phase in $\mathbf{V}$ can be derived as:

$$\phi_{m_1,n} - \phi_{m_2,n} = \psi_{n,m_2} - \psi_{n,m_1}. \tag{15}$$

In Section III, it's mentioned that in order to reduce communication overhead, the POS terminal needs to efficiently compress the $\mathbf{V}$ to minimize feedback volume. Therefore, by employing phase adjustment, we obtain $\tilde{\mathbf{V}}$. In summary, the (m, n) element in $\tilde{\mathbf{V}}$ can be expressed as

$$\left[\tilde{\mathbf{V}}\right]_{m,n} = b_{m,n} e^{j(\phi_{m,n} - \phi_{M,n})} = \frac{a_{n,m}}{\sigma_n} e^{j(\psi_{n,M} - \psi_{n,m})}. \tag{16}$$

Next, we can characterize the finger keystroke displacement $d(t)$ using the amplitude of the BFI:

$$\left|\left[\tilde{\mathbf{V}}\right]_{m,n}\right|^2 = \frac{\mathcal{A}_{m,n}}{\sum_{\mu}^{M} \mathcal{A}_{\mu,n}}, \tag{17}$$

where the $\mathcal{A}_{m,n}$ can be derived as:

$$\mathcal{A}_{m,n} = |h_{s_{n,m}}|^2 + A_{d_{n,m}}^2 + 2A_{d_{n,m}} |h_{s_{n,m}}| \cos\left(\frac{2\pi f d(t)}{c} + \Delta\psi_{n,m}\right). \tag{18}$$

Similarly, the equation of $d(t)$ and phase of the BFI can be shown as follow:

$$\angle\left[\tilde{\mathbf{V}}\right]_{m,n} = \psi_{n,M} - \psi_{n,m} = \mathcal{P}_{n,M} - \mathcal{P}_{n,m}, \tag{19}$$

where

$$\mathcal{P}_{n,m} = \angle h_{n,m} - \frac{A_{d_{n,m}} \sin(\frac{2\pi f d(t)}{c} + \Delta\psi_{n,m})}{a_{n,m}}. \tag{20}$$

While the closed-form relationship between finger movement displacement and BFI has been established, the minute amplitudes are highly susceptible to noise interference, compounded by uncertain phase offsets. Consequently, relying solely on either amplitude or phase for reconstructing keystroke trajectories presents significant challenges. Therefore, we propose self-dividing the BFI to mitigate the aforementioned

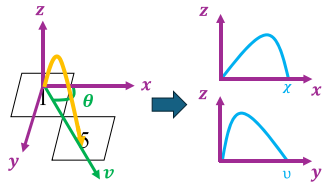


Fig. 7. Trajectory analysis.

interferences:

$$\frac{\tilde{\mathbf{V}}_{n,m_1}}{\tilde{\mathbf{V}}_{n,m_2}} = \frac{\frac{a_{n,m_1}}{\sigma_n} e^{j(\psi_{n,M} - \psi_{n,m_1})}}{\frac{a_{n,m_2}}{\sigma_n} e^{j(\psi_{n,M} - \psi_{n,m_2})}} = \frac{h^{\dagger}_{s_{n,m_1}} + A_{n,m_1} e^{j(\frac{2\pi f d(t)}{c}) + \Delta_{n,m_1}}}{h^{\dagger}_{s_{n,m_2}} + A_{n,m_2} e^{j(\frac{2\pi f d(t)}{c}) + \Delta_{n,m_2}}}, \tag{21}$$

Henceforth, we can utilize BFI to characterize finger keystroke movement displacement as:

$$d(t) = \mathcal{F}\left(\frac{\tilde{\mathbf{V}}_{n,m_1}}{\tilde{\mathbf{V}}_{n,m_2}}\right). \tag{22}$$

After obtaining keystroke displacement $d(t)$ from BFI, it needs to be decomposed into three-dimensional coordinates for subsequent estimation. We will introduce the decomposition steps in conjunction with Fig. 7. Taking the 1-5 key pair as an example, assuming the finger keystroke trajectory equation is $z = f(x, y)$, where

$$x(t) = vt\cos(\theta), \tag{23}$$

$$y(t) = vt\sin(\theta), \tag{24}$$

with v representing the finger movement speed and θ the direction of movement, these parameters will be obtained in

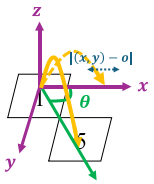


Fig. 8. Estimation of θ.

Section IV-C2. Then, $z(t)$ can be approximated by a high-dimensional polynomial:

$$z(t) = \sum_{i=0}^{n} \alpha_i t^i, \tag{25}$$

where the coefficients α_i can be determined by solving a system of linear equations. For the trajectory equation $z = f(x, y)$, $d(t)$ is the trajectory length of the curve:

$$d(t) = \int \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2 + \left(\frac{dz}{dt}\right)^2}\, dt. \tag{26}$$

Simply by solving (23), (24), (25), and (26) simultaneously, we can obtain the coordinates of the finger endpoint (χ, υ) when $z = 0$.

2) Parameter Estimation: In Section IV-C1, we establish the relationship between trajectory of typing a key pair and BFI, setting the initial key pair position at the origin and obtaining the endpoint key position (χ, υ). Next, we need to address two remaining unknown parameters. Among them, the directional angle θ, as shown in Fig. 8, can be derived from the obtained endpoint key position as follows:

$$\ell = 1 - \frac{|(x, y) - (\chi, \upsilon)|}{\sum_{\theta \in \Theta} |(x, y) - (\chi, \upsilon)|}, \tag{27}$$

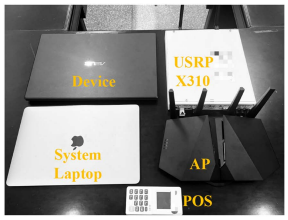
where $\ell \in [0, 1]$ represents the confidence level, and $\Theta = \{\theta_1, \ldots, \theta_i\}$ represents the set of possible directions of finger movement for entering a key pair. By solving $\theta = \arg\max_{\theta \in \Theta} \ell$, we can determine the directional angle. For estimating typing speed, we can simply refer to existing work [50], using a constant speed to represent the typing speed for all individuals. However, this approach sacrifices a certain degree of accuracy. Another idea is to determine it based on some prior knowledge. From our observations, many POS machine password entries require confirmation twice, which undoubtedly provides us with an opportunity to estimate typing speed. For two identical passwords, we only need to apply the approximate speed of the first entry to predict the second entry.

3) Key Pair Estimation: After obtaining the two parameters of directional angle and typing speed, it becomes relatively easy to determine the relative positional relationship between key pairs. However, this relative position can have multiple possibilities, as illustrated in Fig. 8, such as '1-5', '5-1', '4-8', '8-4', and so on. Although the asymmetry of the POS machine keyboard can reduce some interference (such as the difference between '1-6' and '1-8'), the positional ambiguity between key pairs remains significant. Therefore, we suggest modeling the transition between key pairs as a Hidden Markov Model (HMM).

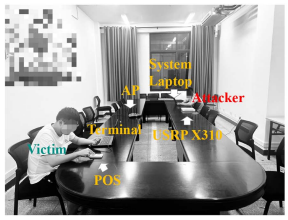
4) Hidden Markov Model: After conducting the aforementioned analysis, we can anticipate potential key pairs by analyzing the displacement distances in both the x and y directions resulting from finger taps. Despite the inherent layout imbalances of the keys on the POS terminal, which aid in eliminating certain key pair possibilities, there remains a multitude of potential combinations that meet the established criteria, consequently resulting in keystroke inference errors. To mitigate such inaccuracies, we propose a method for modeling the interplay between consecutive key pairs. By establishing the relationship where the end key of the previous pair matches the start key of the next pair, we can significantly narrow down the potential combinations. Specifically, if a key pair has a displacement of 2 units in the x-direction and 1 unit in the y-direction, and the next key pair also has a displacement of 2 units in the x-direction and 1 unit in the y-direction, then the possible combinations for this key pair sequence are reduced to only three: '1-6-1', '1-6-7', and '4-9-4', significantly reducing the number of candidates and thereby lowering the error rate. Expanding this approach to encompass a sequence of five key pairs for a 6-digit PIN code yields even more accurate predictions. Hence, we advocate for the utilization of Hidden Markov Models to effectively capture and model the sequential dependencies inherent in key pair sequences.

After the above analysis, BeamThief can predict the possible key pairs that satisfy by obtaining the displacement distances in the x and y directions when fingers tap. Although the keys on the POS terminal exhibit imbalance in terms of layout, which helps eliminate certain key pairs, there are still multiple possible key pairs that meet the criteria, leading to errors in keystroke inference. To reduce such errors, we consider modeling the interdependence of consecutive key pairs. Specifically, if a key pair has a displacement of 2 units in the x-direction and 1 unit in the y-direction, and the next key pair also has a displacement of 2 units in the x-direction and 1 unit in the y-direction, then the possible combinations for this key pair sequence are reduced to only three: '1-6-1', '1-6-7', and '4-9-4', significantly reducing the number of candidates and thereby lowering the error rate. Extending this approach to a sequence of five key pairs for a 6-digit PIN code can lead to even better predictions. Therefore, we propose using Hidden Markov Model to model the sequences of interdependent key pairs.

BeamThief models the keystroke process as a HMM with features represented by $\gamma = (N, M, \wp, \mathbf{S}, \mathbf{O})$, where $N = 100$ is the number of hidden states, representing the number of key pairs. $M = 3 \times 4$ (the number of x-direction displacement values $\times$ the number of y-direction displacement values) is the number of observation values. $\wp$ is the initial state probability vector, representing the probability distribution of the system starting in each possible hidden state. Due to the equal probability of each key position, BeamThief adopts a uniform probability distribution here. $\mathbf{S} \in \mathbb{C}^{N \times N}$ is the state transition probability matrix, representing the probability of the system transitioning from one hidden state to another, which can be generated based on the relationships between key pairs. For

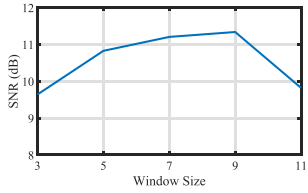


(a) Hardware implementation.

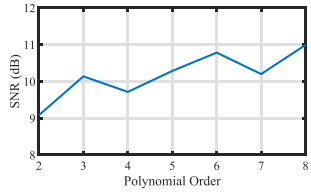


(b) An experimental scenario in empty room.

Fig. 9. Implementation of BeamThief.



(a) Different window sizes.



(b) Different polynomial orders.

Fig. 10. Benchmark study of MRC.

example, the transition probability from key pair '1-6' to '6-x' is 0.1. $\mathbf{O} \in \mathbb{C}^{N \times M}$ is the observation probability matrix, representing the probability distribution of generating observation values in each hidden state, which can be obtained by evaluating the probability of generating certain observation values given a specific key pair. The goal of BeamThief is to find the optimal hidden sequence $Y = \{y_1, y_2, y_3, y_4, y_5\}$ given the sequence of key pairs $X = \{x_1, x_2, x_3, x_4, x_5\}$ to achieve $\max(P(Y|X, \gamma))$. This optimization problem can be solved using the Forward-Backward Algorithm [51].

V. Implementation and Evaluation

A. Implementation

We develop the BeamThief system on a MacBook Pro computer. By leveraging the built-in sniffer of MacBook in monitor mode, we can capture the Action No ACK frames on specific Wi-Fi channels (in our experiment the channel is set to 6, which is 2437MHz). To extract the BFI, we utilize Wiresharkb software [52] to analyze the captured packets. Specifically, we focus on the "Compressed_Beamforming_Report" field within these frames, which contains the essential BFI data. The experimental hardware involves a ASUS Mars 15 laptop and a MPOS machine [53] as the POS terminal, using iPerf [54] to generate consistent traffic. The access point is the Asus RT-AX82U router. Please refer to Fig. 9(a) for the visual representation of all the hardware. In the MRC algorithm, we set the window size to 9 and the polynomial order to 8.

B. Methodology

1) Experiment Setup: The experimental setup for this study involves recruiting 10 subjects, including 6 males and 4 females, aged between 20 and 30 years. The volunteers are asked to enter passwords on a POS machine's numeric keypad. Each subject is required to enter 100 passwords, which are selected from a pool of 10,000 randomly generated 6-digit passwords. In addition, subjects also need to individually input each numerical key (0-9) 100 times according to their habits to provide data for single keystroke identification evaluation. We consider two typing patterns that conform to the majority of the subject's password entry behavior. Pattern 1 is placing the POS machine on a table for single-handed typing, while Pattern 2 involves holding the POS machine with one hand and using the other hand for typing. Subjects are required to complete the above two types of collection in both patterns. The subjects are instructed to complete their typing within 4 to 10 seconds. Each BFI segment is extracted for a duration of 10 seconds. We conduct the experiments in empty rooms (ER), shopping malls (SM), and grocery stores (GS). In addition to collecting BFI series, we also simultaneously obtain CSI from USRP X310 devices and AP as the comparative baselines for WINK [16] and WiPOS [19]. The distance between the Wi-Fi antenna of POS terminal and the subject ranges from 0 to 2 meters, and between the AP and the Wi-Fi antenna of POS terminal ranges from 1 to 10 meters. Fig. 9(b) shows an experimental scenario. During the data collection process, all volunteers were informed about the purpose and application of the experimental data. To ensure confidentiality and anonymity, we recorded only volunteer identification numbers. The data collection process strictly adhered to the standard procedures required by our Institutional Review Board (IRB).

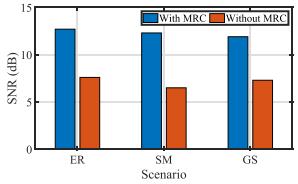
2) Metrics: To evaluate the performance of BeamThief, we define several metrics as follows.

F1-Score. The F1-Score is a metric that combines precision and recall to evaluate the performance of single keystroke identification. The F1-score is defined as $F1 - score_k = 2 \times \frac{P_k + R_k}{P_k \times R_k}$, where $P_k = m_k^T/(m_k^T + m_k^F)$ represents the precision of identifying key k and $R_k = m_k^T/n_k$ represents the recall of identifying key k. m_k^T is the number of keystrokes correctly identified as key k, m_k^F is the number of keystrokes mistakenly identified as key k but are actually other keys, and n_k represents the total number of keystrokes for key k.

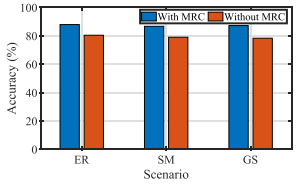
Top-n Word Accuracy. Given n inferred word candidates, the top-n word accuracy is defined as a metric to evaluate the overall performance of keystroke inference. Assuming that there are k texts during input, the top-n word accuracy is calculated as $A^n = i/k$, where i represents the number of inferences where the top-n word candidates contain the ground truth.

C. Overall Performance

1) Series Enhancement.: Our study applies MRC techniques to enhance the BFI series, aiming to evaluate its effectiveness using SNR as the evaluation metric. Fig. 10(a) visually presents the optimal parameters we carefully selected while adjusting the window size. our experimental setup, where we examine the enhanced capability of MRC across three distinct scenarios. Moreover, Fig. 10 illustrates the selection of the polynomial



(a) SNR w/wo MRC in 3 scenarios.



(b) Single keystroke identification accuracy w/wo MRC in 3 scenarios.

Fig. 11. Evaluation of series enhancement.

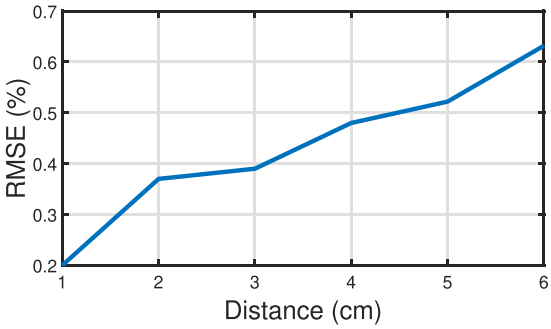
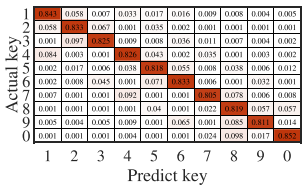
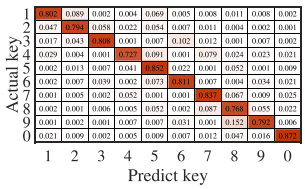


Fig. 12. The RMSE of theoretical and actual BFI values.



(a) BeamThief.



(b) WiPOS.

Fig. 13. Confusion matrix for single keystroke identification accuracy of BeamThief and WiPOS.

order parameter after determining a window size of 9. Next, we present the performance of MRC in Fig. 11. Across three different scenarios, MRC demonstrates an average improvement of 4.8 dB in SNR. Furthermore, regarding single keystroke identification accuracy, MRC achieves a notable enhancement of 8.1%. Fig. 9(b) shows an experimental scenario.

2) Performance of Single Keystroke Identification: We evaluate the correctness of the relationship between the BFI series and finger movements by comparing the Root Mean Squared Error (RMSE) between theoretical and actual values. As shown in Fig. 12, we plot the relationship between the sliding distance $d(t)$ and the RMSE. It can be observed that the theoretical and actual values consistently maintain a high correlation, thereby demonstrating the correctness of the model representing the relationship between the BFI series and finger movements. Furthermore, we validate this correctness by evaluating the performance of single-key recognition and comparing it with the WiPOS system (since WINK is a sequence recognizer and does not provide individual key recognition capability). Fig. 13 gives

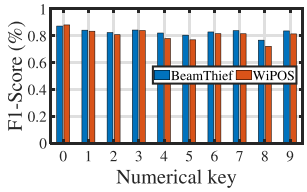


Fig. 14. Comparison for F1-Score of single keystroke identification.

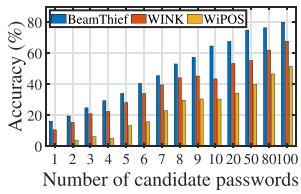
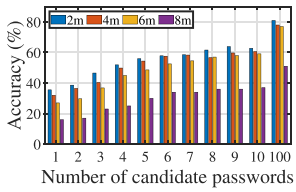
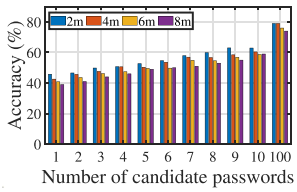


Fig. 15. Comparison for top-N word accuracy.



(a) Distance between Wi-Fi antenna of POS terminal and victim.



(b) Distance between Wi-Fi antenna of POS terminal and AP.

Fig. 16. Impact of distance.

the confusion matrices for BeamThief and WiPOS. Regarding classification accuracy, BeamThief exhibits a rate of 82.7%. Comparatively, WiPOS 80.6%. Then we provide the F1-score comparison with WiPOS in Fig. 14, with BeamThief maintaining at 82.6% while WiPOS maintains around 80.7%. For this phenomenon, we offer an explanation based on our experimental setup, suggesting that deploying the USRP X310 antenna for capturing CSI is farther compared to deploying the Wi-Fi antenna of POS terminal. Consequently, BeamThief exhibits better perceptual performance, which aligns with a realistic scenario assumption.

3) Performance of Keystroke Inference: We proceed to evaluate the overall performance of BeamThief. For each keystroke inference, BeamThief selects the top-n candidates based on all potential passwords. As shown in Fig. 15, BeamThief's accuracy falls below 65% in the top-10 candidates, whereas WINK and WiPOS achieve only 46% and 32% accuracy under the same conditions, respectively. Furthermore, when considering the top-100 attempts, BeamThief achieves an accuracy of 79%, surpassing the accuracies of WINK (69%) and WiPOS (53%).

4) Impact of Distance: Due to the nature of BFI information propagation, the distance between the Wi-Fi antenna of POS terminal and the victim, as well as the distance between the Wi-Fi antenna of POS terminal and the AP, can both have an impact on the performance of BeamThief. Fig. 16 illustrates the top-1 to top-10 accuracy of BeamThief at different distances.

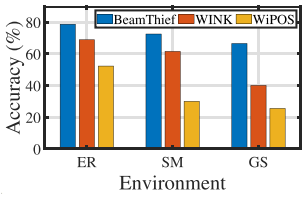


Fig. 17. Impact of environments.

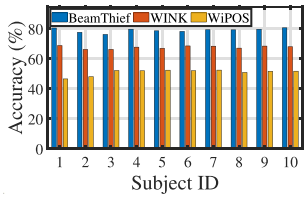


Fig. 18. Impact of subjects.

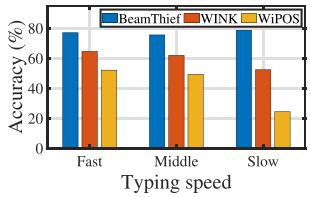


Fig. 19. Impact of typing speed.

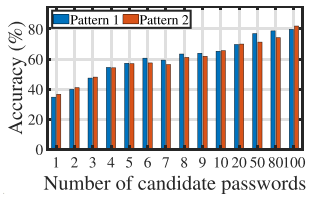


Fig. 20. Impact of typing patterns.

Based on Fig. 16(a), it can be observed that the performance of BeamThief decreases as the distance between the victim and the device increases. Particularly, when the distance reaches 1.5m, the top-10 accuracy drops from 62% to 33% , indicating that BFI's ability to perceive keystrokes diminishes as the distance increases, with a sharp decline occurring at the 150 cm mark. However, at a distance of 100 cm, BeamThief maintains a top-100 accuracy of 79%, demonstrating its competence in most password theft scenarios. Additionally, Fig. 16(b) depicts the influence of the distance between the device and the AP on BeamThief's performance, showing that as the distance increases, the top-10 accuracy of BeamThief decreases by approximately 15% . This is because a greater distance between the device and the AP results in weaker Wi-Fi signals and increased interference. Finally, we will investigate the impact of attack distance on performance. BeamThief initiates attacks by sniffing Wi-Fi packets in the environment, so the theoretical effective attack range is between 20 and 50 meters. Therefore, we conduct experiments in an open outdoor area (SNR $= 24$), with a fixed AP and POS terminal, at attack distances ranging from 10 to 50 meters in 5-meter increments. As shown in Fig. 21, BeamThief's effective attack range is approximately 20 meters.

5) Impact of Environments: In order to investigate the robustness of BeamThief in different environmental settings, we conducted further tests in three distinct scenarios. Empty rooms can be considered as an ideal environment with minimal interference. On the other hand, shopping malls and grocery stores are more representative of real-life settings, characterized by the presence of moving objects in the surroundings. The average SNRs at these three locations are 27 dB, 24 dB, and 20 dB, respectively. Fig. 17 illustrates the performance of BeamThief across three scenarios, showcasing its optimal performance in an empty room, while experiencing varying degrees of degradation in a shopping mall and grocery store. This demonstrates how environmental factors affect BeamThief's keystroke inference performance by influencing the channel. In contrast, WINK and WiPOS exhibit more pronounced instability in shopping mall and grocery store scenarios, with average accuracies decreasing by 21% and 32% , respectively. The reason for this phenomenon can be attributed to the information provided in Section V-C4, which explains that the distance between moving objects and Wi-Fi antennas of POS terminals is typically greater than 150 cm. BeamThief's performance benefits from distance, as it is less affected by such interferences compared to the other two methods.

6) Impact of Input Manner: In our experiment, subjects are instructed to enter passwords according to the methods provided in the experimental setup. Therefore, we evaluate the impact of input manner on BeamThief from three dimensions based on the experimental setup. These dimensions include the impact of different subjects, typing speed, typing patterns, and typing fingers. Fig. 18 presents the impact of different volunteers on performance, where we provide the top-100 accuracy for ten volunteers under three methods. It can be observed that all three methods exhibit robustness across different subjects. WINK and WiPOS benefit from their neural network's generalization capability, while BeamThief maintains robustness across different subjects due to its well-described finger movement modeling and parameter estimation. Fig. 19 explains the effect of different typing speeds, we provide the comparison accuracy of top-100. We categorize cases where a six-digit password is typed within 5 seconds as fast speed, within 5-7 seconds as middle speed, and above 7 seconds as slow speed. Clearly, due to the predicted keystroke speed parameters in advance, BeamThief demonstrates well stability. In contrast, WINK and WiPOS experience decreased accuracy by 19% and 27% respectively, as they struggle to cope with faster keystroke speeds. Finally, Fig. 20 illustrates the top-n performance of BeamThief under two typing modes. BeamThief performs better under Pattern 1 (POS machine placed on the desktop) compared to Pattern 2 (POS machine held in one hand), as placing the POS machine on the desktop provides more stable keystroke actions for the victims. Finally, we investigate the impact of using different fingers for keystrokes on performance. As shown in Fig. 22, there is little difference in performance between using the middle

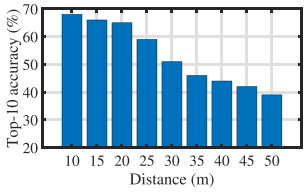


Fig. 21. Impact of attack distance.

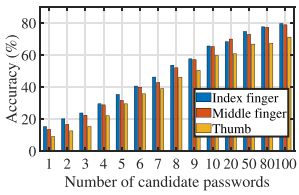
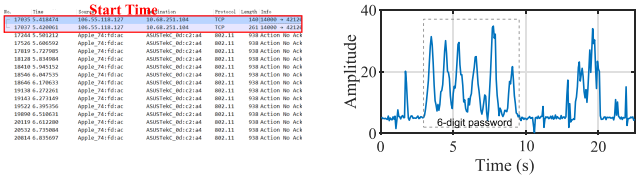


Fig. 22. Impact of different fingers.



(a) Confirming the moment of initiating the attack.

(b) Verifying the BFI series waveforms corresponding to the 6-digit password.

Fig. 23. Determine the time to attack.

finger and the index finger for keystrokes, so switching to the middle finger does not have a significant effect. However, since the keystroke motion of the thumb differs noticeably from that of the index finger, it results in different changes in channel state. Consequently, this leads to an approximate 8% decrease in accuracy.

D. Determine the Time to Attack

In addition to visual observation by attackers, the attack can also be executed through the identification of specific packets. During the experiments, a particular store initiates transactions by directing the device to connect to a designated IP address server for interaction with the payment service provider or bank, typically lasting for a certain period (usually around one week). As illustrated in Fig. 23(a), the appearance of the IP address "10, 68, xx, xx" indicates imminent payment initiation, allowing us to prepare for the attack. Moreover, in real-world scenarios, the victims may engage in additional actions such as card swiping and button confirmation. To address this situation, BeamThief merely needs to identify a consecutive sequence of six waveforms within a specific time interval to infer the corresponding password keystrokes shown as Fig. 23(b).

VI. Discussion

A. Limitation

The results in Section V-C4 indicate that when the attack distance exceeds 150 centimeters, the PIN recovery rate of the top-10 candidates drops to 33%. If successful remote execution of the attack could be achieved, the threat could become even more severe. The primary reason for the limited distance here is the attenuation of Wi-Fi signal strength. As part of our future work, we plan to construct a more robust prototype equipped with dedicated components capable of extracting useful signals from noise and weak BFI measurements, thus extending the attack distance. Additionally, while BeamThief does not require a large amount of training data, estimating keystroke finger sliding speed, as mentioned in Section II, requires the victim to input the password at least twice, posing a challenge to the attack. We suggest conducting some prior data collection before the attack to refine the speed estimation. Finally, it only applies when the victim keeps the mobile device relatively stable. To partially address this issue, conducting prior analysis on the victim or employing relevant motion models for targeted attacks may be helpful.

B. Defending Strategies

The most direct defense strategy is to consciously disrupt typing habits. As described in Section V-C6, altering keystroke posture, changing finger gliding time, or introducing slight pauses between keystrokes can effectively interfere with BeamThief's estimation of finger gliding speed, thereby thwarting its attack.

Furthermore, users can undermine BeamThief's directional estimation by moving their fingers to other positions after each keystroke before sliding to the next key. In this scenario, attackers would find it challenging to accurately obtain individual key pairs, leading to significant errors when attempting to guess the entire password.

Another approach involves using randomized POS terminal keyboards. If different POS terminals have unique, randomized keyboard layouts, attackers would struggle to map observed finger movements to specific keys without prior knowledge of the layout.

While these methods can effectively prevent password inference, they all reduce user convenience to some extent. Whether requiring users to change typing habits or search for keys on a disorganized keyboard, these approaches may lead to frequent input errors. Preventing such third-party attacks at the protocol level could offer a balanced solution between user convenience and privacy protection. The method outlined in [55] blocks Wi-Fi eavesdropping attacks by encrypting source-defined channels. In our future work, we plan to inject fake BFI into the channel to confuse attackers and defeat eavesdropping attempts. This approach aims to enhance system security without compromising the normal user experience.

C. Potential Risks

The ability to obtain passwords input through POS machine keypads can have significant and concerning implications. One of the primary risks is the potential for illegal transactions.

Attackers who gain access to these passwords can use them to perform unauthorized purchases, leading to financial losses for both individuals and businesses. Moreover, this stolen information can be sold on the dark web, further perpetuating criminal activities.

Another critical risk is identity theft. By capturing passwords and other sensitive information entered into POS systems, attackers can potentially gain access to additional personal data linked to the user's financial accounts. This can lead to a range of fraudulent activities, from opening new credit accounts in the victim's name to draining existing accounts. The consequences of such breaches extend beyond immediate financial loss, often causing long-term damage to the victims' credit scores and financial stability.

Additionally, the compromise of POS machine passwords can undermine consumer trust in the security of electronic transactions. If customers believe that their financial information is at risk when using POS machines, they may be reluctant to use these systems, affecting businesses that rely on electronic payments.

D. The Advantages and Disadvantages of Training-Based Versus Non-Training-Based Methods

In the conference version of this paper, we employ a training-based approach to accomplish keystroke inference tasks. However, in this version, we have switched to a non-training-based approach. Experimental results indicate that the training-based method exhibits superior robustness and accuracy across different environments, subjects, and typing patterns. Conversely, the non-training-based method sacrifices some robustness and accuracy in exchange for a more relaxed attack scenario, where the arduous task of data collection is avoided, allowing for attacks to be conducted without a large dataset of targets.

VII. Conclusion

In this paper, we present a novel BFI-based side-channel attack design and evaluation that leverages Wi-Fi signals to infer passwords entered by victims on POS machines. Analysis and experimentation demonstrate that BFI possesses superior potential as a eavesdropping side-channel compared to CSI. Consequently, BeamThief requires lower device requirements, allowing attacks to be conducted using conventional laptop computers without the need for intrusion or pre-deployment of additional equipment. We theoretically analyze how BeamThief's observation of password entry actions affects channel states and map them to BFI, establishing a mathematical model. By applying this model, BeamThief can infer keystrokes without requiring training, thus avoiding the extensive prior data collection needed by previous training-based methods. Extensive real-world experiments with a prototype of BeamThief have substantiated its effectiveness in password theft.

References

[1] N. Report, "Global POS market in 2023," 2022. [Online]. Available: https://nilsonreport.com/

[2] A. Maiti, O. Armbruster, M. Jadliwala, and J. He, "Smartwatch-based keystroke inference attacks and context-aware protection mechanisms," in *Proc. 11th ACM Asia Conf. Comput. Commun. Secur.*, 2016, pp. 795–806.

[3] H. Cao et al., "Data augmentation-enabled continuous user authentication via passive vibration response," *IEEE Internet Things J.*, vol. 10, no. 16, pp. 14137–14151, Aug. 2023.

[4] J. Hu et al., "Combining IMU with acoustics for head motion tracking leveraging wireless earphone," *IEEE Trans. Mobile Comput.*, vol. 23, no. 6, pp. 6835–6847, Jun. 2024.

[5] L. Lu et al., "KeyListener: Inferring keystrokes on qwerty keyboard of touch screen through acoustic signals," in *Proc. 34th IEEE Conf. Comput. Commun.*, 2019, pp. 775–783.

[6] M. Zhou et al., "PatternListener: Cracking android pattern lock using acoustic signals," in *Proc. 25th ACM Conf. Comput. Commun. Secur.*, 2018, pp. 1775–1787.

[7] W. Huang, W. Tang, H. Chen, H. Jiang, and Y. Zhang, "Unauthorized microphone access restraint based on user behavior perception in mobile devices," *IEEE Trans. Mobile Comput.*, vol. 23, no. 1, pp. 955–970, Jan. 2024.

[8] J. Hu et al., "EarSonar: An acoustic signal-based middle-ear effusion detection using earphones," in *Proc. 43rd IEEE Int. Conf. Distrib. Comput. Syst.*, 2023, pp. 225–235.

[9] H. Jiang, J. Li, P. Zhao, F. Zeng, Z. Xiao, and A. Iyengar, "Location privacy-preserving mechanisms in location-based services: A comprehensive survey," *ACM Comput. Surv.*, vol. 54, no. 1, pp. 1–36, 2021.

[10] Q. Zhang et al., "Toward predicting stay time for private car users: A RNN-NALU approach," *IEEE Trans. Veh. Technol*, vol. 71, no. 6, pp. 6007–6018, Jun. 2022.

[11] J. Hu, H. Jiang, Z. Xiao, S. Chen, S. Dustdar, and J. Liu, "HeadTrack: Real-time human-computer interaction via wireless earphones," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 4, pp. 990–1002, Apr. 2024.

[12] C. Cai, R. Zheng, and J. Luo, "Ubiquitous acoustic sensing on commodity IoT devices: A survey," *IEEE Commun. Surv. Tut.*, vol. 24, no. 1, pp. 432–454, First Quarter, 2022.

[13] Y. Chen, T. Li, R. Zhang, Y. Zhang, and T. Hedgpeth, "Eyetell: Video-assisted touchscreen keystroke inference from eye movements," in *Proc. 39th IEEE Symp. Secur. Privacy*, 2018, pp. 144–160.

[14] S. Fang, I. Markwood, Y. Liu, S. Zhao, Z. Lu, and H. Zhu, "No training hurdles: Fast training-agnostic attacks to infer your typing," in *Proc. 25th ACM Conf. Comput. Commun. Secur.*, 2018, pp. 1747–1760.

[15] K. Ling, Y. Liu, K. Sun, W. Wang, L. Xie, and Q. Gu, "SpiderMon: Towards using cell towers as illuminating sources for keystroke monitoring," in *Proc. 39th IEEE Conf. Comput. Commun.*, 2020, pp. 666–675.

[16] E. Yang, Q. He, and S. Fang, "WINK: Wireless inference of numerical keystrokes via zero-training spatiotemporal analysis," in *Proc. 29th ACM Conf. Comput. Commun. Secur.*, 2022, pp. 3033–3047.

[17] W. Jin, S. Murali, H. Zhu, and M. Li, "Periscope: A keystroke inference attack using human coupled electromagnetic emanations," in *Proc. 28th ACM Conf. Comput. Commun. Secur.*, 2021, pp. 700–714.

[18] Q. Zhang et al., "Enhancing perception for intelligent vehicles via electromagnetic leakage," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 7, pp. 7029–7043, Jul. 2024.

[19] Z. Zhang et al., "WiPOS: A POS terminal password inference system based on wireless signals," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7506–7516, Aug. 2020.

[20] T. Zheng, Z. Chen, S. Zhang, C. Cai, and J. Luo, "MoRe-Fi: Motion-robust and fine-grained respiration monitoring via deep-learning UWB radar," in *Proc. 19th ACM Conf. Embedded Netw. Sensor Syst.*, 2021, pp. 111–124.

[21] M. Wang et al., "RoPriv: Road network-aware privacy-preserving framework in spatial crowdsourcing," *IEEE Trans. Mobile Comput.*, vol. 23, no. 3, pp. 2351–2366, Mar. 2024.

[22] H. Jiang, S. Chen, Z. Xiao, J. Hu, J. Liu, and S. Dustdar, "Pa-count: Passenger counting in vehicles using Wi-Fi signals," *IEEE Trans. Mobile Comput.*, vol. 23, no. 4, pp. 2684–2697, Apr. 2024.

[23] Z. Chen, C. Cai, T. Zheng, J. Luo, J. Xiong, and X. Wang, "RF-based human activity recognition using signal adapted convolutional neural network," *IEEE Trans. Mobile Comput.*, vol. 22, no. 1, pp. 487–499, Jan. 2023.

[24] Wikipedia, "IEEE_802.11ac-2013," 2023. [Online]. Available: https://en.wikipedia.org/wiki/IEEE_802.11ac-2013

[25] J. Hu et al., "Password-stealing without hacking: Wi-Fi enabled practical keystroke eavesdropping," in *Proc. 30th ACM Conf. Comput. Commun. Secur.*, 2023, pp. 1–14.

[26] F. Zhang et al., "Towards a diffraction-based sensing approach on human activity recognition," in *Proc. 19th ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, New York, NY, USA, 2019, pp. 1–25.

[27] F. Maggi, A. Volpatto, S. Gasparini, G. Boracchi, and S. Zanero, "A fast eavesdropping attack against touchscreens," in *Proc. 7th Int. Conf. Inf. Assurance Secur.*, 2011, pp. 320–325.

[28] Q. Yue, Z. Ling, X. Fu, B. Liu, K. Ren, and W. Zhao, "Blind recognition of touched keys on mobile devices," in *Proc. 21st ACM Conf. Comput. Commun. Secur.*, 2014, pp. 1403–1414.

[29] J. Sun, X. Jin, Y. Chen, J. Zhang, Y. Zhang, and R. Zhang, "VISIBLE: Video-assisted keystroke inference from tablet backside motion," in *Proc. 23rd Netw. Distrib. Syst. Secur. Symp.*, 2016, pp. 1–15.

[30] M. Sabra, A. Maiti, and M. Jadliwala, "Zoom on the keystrokes: Exploiting video calls for keystroke inference attacks," 2020, *arXiv: 2010.12078*.

[31] L. Cai and H. Chen, "TouchLogger: Inferring keystrokes on touch screen from smartphone motion," in *Proc. 6th USENIX Conf. Hot Top. Secur.*, 2016, Art. no. 9.

[32] P. Marquardt, A. Verma, H. Carter, and P. Traynor, "(sp)iPhone: Decoding vibrations from nearby keyboards using mobile phone accelerometers," in *Proc. 18th ACM Conf. Comput. Commun. Secur.*, 2011, pp. 551–562.

[33] C. Wang, J. Liu, X. Guo, Y. Wang, and Y. Chen, "WristSpy: Snooping passcodes in mobile payment using wrist-worn wearables," in *Proc. 38th IEEE Conf. Comput. Commun.*, 2019, pp. 2071–2079.

[34] A. Compagno, M. Conti, D. Lain, and G. Tsudik, "Don't Skype & Type! acoustic eavesdropping in voice-over-IP," in *Proc. 12th ACM Asia Conf. Comput. Commun. Secur.*, 2017, pp. 703–715.

[35] T. Zhu, Q. Ma, S. Zhang, and Y. Liu, "Context-free attacks using keyboard acoustic emanations," in *Proc. 21st ACM Conf. Comput. Commun. Secur.*, 2014, pp. 453–464.

[36] J. Liu, Y. Wang, G. Kar, Y. Chen, J. Yang, and M. Gruteser, "Snooping keystrokes with mm-level audio ranging on a single phone," in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw.*, 2015, pp. 142–154.

[37] J. Hu, T. Zheng, Z. Chen, H. Wang, and J. Luo, "MUSE-Fi: Contactless muti-person sensing exploiting near-field Wi-Fi channel variation," in *Proc. 29th Annu. Int. Conf. Mobile Comput. Netw.*, 2023, pp. 1–15.

[38] T. Zheng, Z. Chen, C. Cai, J. Luo, and X. Zhang, "V2iFi: In-vehicle vital sign monitoring via compact RF sensing," in *Proc. 20th ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, New York, NY, USA, 2020, pp. 1–27.

[39] C. Wu, X. Huang, J. Huang, and G. Xing, "Enabling ubiquitous WiFi sensing with beamforming reports," in *Proc. 37th ACM SIGCOMM Comput. Commun. Rev.*, 2023, pp. 20–32.

[40] H. Cao, D. Liu, H. Jiang, and J. Luo, "MagSign: Harnessing dynamic magnetism for user authentication on IoT devices," *IEEE Trans. Mobile Comput.*, vol. 23, no. 1, pp. 597–611, Jan. 2024.

[41] E. Yi et al., "BFMSense: WiFi sensing using beamforming feedback matrix," in *Proc. 21st USENIX Symp. Netw. Syst. Des. Implementation*, 2024, pp. 1697–1712.

[42] Q. Zhang et al., "Eye of sauron: Long-range hidden spy camera detection and positioning with inbuilt memory EM radiation," in *Proc. 33th USENIX Secur.*, 2024, pp. 1–18.

[43] X. Li, H. Wang, Z. Chen, Z. Jiang, and J. Luo, "UWB-Fi: Pushing Wi-Fi towards Ultra-wideband for Fine-Granularity Sensing," in *Proc. 22nd Annu. Int. Conf. Mobile Syst. Appl. Serv.*, 2024, pp. 42–55.

[44] Power your entire business, 2023. [Online]. Available: https://squareup.com/

[45] Everything you need to run your business smarter, faster, easier, 2023. [Online]. Available: https://www.clover.com/

[46] M. Li et al., "When CSI meets public WiFi: Inferring your mobile phone password via WiFi signals," in *Proc. 23rd ACM Conf. Comput. Commun. Secur.*, 2016, pp. 1068–1079.

[47] E. Perahia and R. Stacey, *Next Generation Wireless LANs: 802.11n and 802.11ac*. Cambridge, U.K.: Cambridge Univ. Press, 2013.

[48] W. Givens, "Computation of plain unitary rotations transforming a general matrix to triangular form," *J. Soc. Ind. Appl. Math.*, vol. 6, no. 1, pp. 26–50, 1958.

[49] N. Otsu et al., "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285–296, pp. 23–27, 1975.

[50] T. Feng et al., "Continuous mobile authentication using touchscreen gestures," in *Proc. IEEE Conf. Technol. Homeland Secur.*, 2012, pp. 451–456.

[51] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," in *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[52] The world's most popular network protocol analyzer, 2016. [Online]. Available: https://www.wireshark.org/

[53] 2023. [Online]. Available: https://www.shengfutong.com.cn/pos/mpos.html

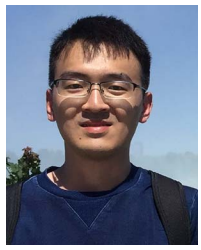
[54] iPerf - the ultimate speed test tool for TCP, UDP and SCTP, 2023. [Online]. Available: https://iperf.fr/

[55] J. Luo, H. Cao, H. Jiang, Y. Yang, and Z. Chen, "MIMOCrypt: Multi-user privacy-preserving Wi-Fi sensing via MIMO encryption," in *Proc. 45th IEEE Symp. Secur. Privacy*, 2024, pp. 1–19.

Siyu Chen (Student Member, IEEE) received the BS degree in communication engineering from Hunan University, Changsha, China, in 2021, where he is currently working toward the PhD degree with the College of Computer Science and Electronic Engineer, Hunan University. He has published papers in IEEE INFOCOM, *IEEE Internet of Things Journal*, *IEEE Transactions on Mobile Computing* and *IEEE Journal on Selected Areas in Communications*. His research interests lie in the area of wireless sensing and Internet of Things security.

Hongbo Jiang (Senior Member, IEEE) received the PhD degree from Case Western Reserve University in 2008. He is now a full professor with the College of Computer Science and Electronic Engineering, Hunan University. He was a professor with the Huazhong University of Science and Technology. His research concerns computer networking, especially algorithms and protocols for wireless and mobile networks. He is serving as the editor for *IEEE/ACM Transactions on Networking*, the associate editor for *IEEE Transactions on Mobile Computing*, and the associate technical editor for *IEEE Communications Magazine*.

Jingyang Hu (Student Member, IEEE) is currently working toward the PhD degree with the College of Computer Science and Electronic Engineering, Hunan University, China. From 2022 to 2023, he works as a joint PhD student with the School of Computer Science and Engineering at Nanyang Technological University (NTU), Singapore. He has published papers in ACM Ubicomp, ACM CCS, IEEE INFOCOM, IEEE ICDCS, *IEEE Transactions on Mobile Computing*, *IEEE Journal on Selected Areas in Communications*, *IEEE Transactions on Intelligent Transportation Systems*, *IEEE Internet of Things Journal*, etc. His research interests include mobile and pervasive computing, the Internet of Things, and machine learning.



Tianyue Zheng (Member, IEEE) received the BEng degree from Harbin Institute of Technology, China, the MEng degree from the University of Toronto, Canada, and the PhD degree from Nanyang Technological University, Singapore. He is currently an assistant professor with the Department of Computer Science and Engineering, Southern University of Science and Technology, China. His research interests include mobile and pervasive computing, the Internet of Things, and machine learning. More information can be found at https://tianyuez.github.io.

Mengyuan Wang (Member, IEEE) received the BS degree from Hunan University, Changsha, China, in 2019. He is currently working toward the PhD degree with the College of Computer Science and Electronic Engineering, Hunan University. His research interests include information security and mobile computing.

Zhu Xiao (Senior Member, IEEE) received the MS and PhD degrees in communication and information system from Xidian University, China, in 2007 and 2009, respectively. From 2010 to 2012, he was a research fellow with the Department of Computer Science and Technology, University of Bedfordshire, U.K. He is currently a full professor with the College of Computer Science and Electronic Engineering, Hunan University, China. His research interests include mobile communications, wireless localization, Internet of Vehicles, and trajectory data mining.

Daibo Liu (Member, IEEE) received the PhD degree in computer science and engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2018. He was a visiting researcher with School of Software, Tsinghua University from 2014-2016, and Department of Electrical and Computer Engineering, University of Wisconsin–Madison from 2016-2017. He is currently an assistant professor with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China. His research interests cover the broad areas of low power wireless networks, mobile and pervasive computing, and system security. He is a member of the ACM.

Jun Luo (Fellow, IEEE) received the BS and MS degrees in electrical engineering from Tsinghua University, China, and the PhD degree in computer science from EPFL (Swiss Federal Institute of Technology in Lausanne), Lausanne, Switzerland. From 2006 to 2008, he has worked as a postdoctoral research fellow with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada. In 2008, he joined the faculty of the School Of Computer Science and Engineering, Nanyang Technological University in Singapore, where he is currently an associate professor. His research interests include mobile and pervasive computing, wireless networking, machine learning and computer vision, applied operations research, as well as security. More information can be found at http://www.ntu.edu.sg/home/junluo.