MuKI-Fi: Multi-person Keystroke Inference with BFI-enabled Wi-Fi Sensing

Hongbo Wang, *Student Member, IEEE*, Jingyang Hu, *Student Member, IEEE*, Tianyue Zheng, *Student Member, IEEE*, Jingzhi Hu, *Member, IEEE*, Zhe Chen, *Member, IEEE*, Hongbo Jiang, *Senior Member, IEEE*, Yuanjin Zheng, *Senior Member, IEEE* and Jun Luo, *Fellow, IEEE*

Abstract—The contact-free sensing nature of Wi-Fi has been leveraged to achieve privacy breaches such as *keystroke inference* (KI). However, the use of CSI (*channel state information*) in existing attacks is highly questionable due to its signal instability and hardness to acquire. Moreover, such Wi-fi-based attacks are confined to only one victim because Wi-Fi sensing offers insufficient range resolution to physically differentiate multiple victims. To this end, we propose MuKI-Fi to enable, for the first time, *multi-person* KI, leveraging BFI (*beamforming feedback information*), a new feature offered by latest Wi-Fi hardware, transmitted in clear-text by smartphones. BFI's characteristics, clear-text communication and signal stability, make it readily acquirable and usable by any other Wi-Fi devices switching to monitor mode without the need for *low-level* hacking on hardware. Moreover, to improve upon existing KI methods offering very limited generalizability across diversified application scenarios, MuKI-Fi innovates in an adversarial learning scheme to enable its inference generalizable towards unseen scenarios. Finally, we discover that, as a smartphone is in close proximity to a victim, the variations of BFI caused by that victim's keystrokes in such *near-field* substantially outweigh those caused by other distant victims; this phenomenon naturally allows for multi-person KI. Our extensive evaluations clearly demonstrate that MuKI-Fi can effectively eavesdrop on the keystrokes of multiple subjects, achieving 87.1% accuracy for individual keystrokes and up to 81% top-100 accuracy for stealing passwords from mobile applications(e.g., WeChat) on average.

Index Terms—Keystroke inference attack, password-stealing, Wi-Fi sensing, multi-person sensing, beamforming feedback information, wireless security.

1 Introduction

In modern societies, mobile devices like smartphones and tablets, along with their software applications, are commonly adopted to identify users, making password theft almost equivalent to identity theft [1], [2]. Consequently, diversified eavesdropping attacks have emerged, either *direct* [3], [4] or *indirect* [5], [6], [7], [8], [9], [10], [11]. Bearing no need to have a visual on the target screen, the indirect attacks are particularly threatening as they leverage side-channels to infer passwords in a stealthy manner. Typical side-channels include acoustic [6], [7], electromagnetic emission [12], indirect vision [8], [9], [13], [14], and motion sensors [10], [11], [15]. In addition to these, recently, Wi-Fi CSI (*channel state information*) has been exploited for side-channel attacks [16], [17]. Essentially, since keystrokes affect wireless channels as shown in Fig. 1, the "twisted" CSIs can be used to infer individual keys involved in typing a password, without the need for external devices close to the victim device or any compromise of the victim device itself.

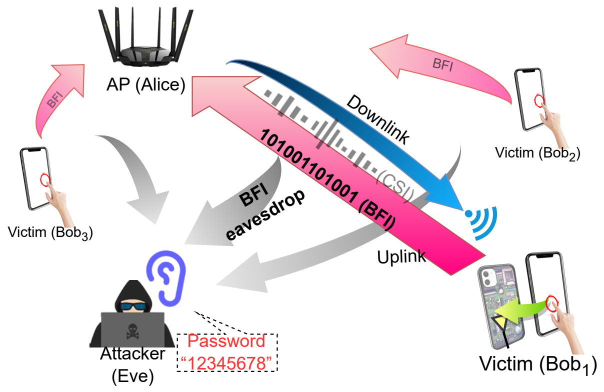


Fig. 1. Vision of MuKI-Fi: in multi-person scenario, eavesdropping on clear-text BFI (representing downlink channel states) transmitted to the AP, Eve can readily infer any Bob's password typing that physically "hits" the Wi-Fi channel.

However, CSI-based attacks suffer from two critical issues. Firstly, though CSI was hacked[1] from Wi-Fi hardware more than a decade ago [20], only a handful of such hardware have been hacked by far while Wi-Fi standards/technologies are constantly getting upgraded every two or three years,[2] which limits the feasibility to obtain

- *H. Wang, T. Zheng, J. Hu and J. Luo are with the School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore. H. Wang is also with the Collaborative Initiative, Interdisciplinary Graduate Programme, NTU Singapore.*
  E-mail: {hongbo001, tianyue002, jingzhi.hu, junluo}@ntu.edu.sg
- *J. Hu and H. Jiang are with the College of Computer Science and Electronic Engineering, Hunan University, China.*
  E-mail: fbhhjy@hnu.edu.cn, hongbojiang2004@gmail.com
- *Z. Chen is with Intelligent Networking and Computing Research Center and School of Computer Science, Fudan University, China.*
  E-mail: zhechen13@fudan.edu.cn
- *Y. Zheng is with the School of Electrical and Electronic Engineering, NTU Singapore. E-mail: yjzheng@ntu.edu.sg*

[1] Instead of high-level software/code hacking, here we refer to a *low-level hacking*, including firmware patching [18] and driver modification [19], on Wi-Fi hardware.

[2] As a matter of fact, most research proposals driven by Wi-Fi CSIs are still leveraging the 15-year-old Wi-Fi 4 hardware [21].

CSI. Secondly, Wi-Fi signal usually suffers severe cross-technology interference [22] in addition to co-channel and adjacent channel interference [23], [24], leading to signal instability of CSI. This instability degrades the performance of CSI-based sensing applications, and we will delve into this phenomenon further in Sec. 2. Therefore, the practicality of CSI-based side-channel attacks is highly questionable, posing a substantial obstacle to the development of a Wi-Fi keystroke inference (KI) system.

Moreover, there is an inherent challenge in password-stealing scenarios: passwords lack linguistic structure in natural languages (e.g., word structure and occurrence frequency of letters) to serve as prior information and features; this has forced existing password inference methods to either rely on independent keystroke features [16] or leverage transition features between two keystrokes [17]. Nonetheless, as these features have strong environment dependency, the resulting inference methods can hardly be generalized to unseen scenarios. Although supervised learning techniques may address this issue with sufficient training data, gathering such a labeled dataset can be prohibitively difficult due to diversified smartphones and human typing habits.

Besides, even if feasible under certain conditions, prior CSI-based attacks are limited to scenarios involving only one single victim, due to an inherent obstacle in Wi-Fi sensing. Specifically, the contention-based multi-access nature of Wi-Fi communication does not provide a sufficiently wide bandwidth to offer a range resolution capable of distinguishing different sensing subjects. To overcome this limitation, two approaches have been attempted. On one hand, many distributed antennas can be used to achieve enhanced spatial resolution for separating subjects [25], at the cost of messing up with the Wi-Fi communication functions. On the other hand, signal processing techniques for separating subjects at the CSI level have been attempted [26], but there is a lack of theoretical guarantee on the success of such signal-level separation [27]. Despite these efforts, the special requirements for Wi-Fi devices and the long processing time have rendered these approaches impractical for keystroke eavesdropping, thus multi-person KI remains a challenge.

To tackle these challenges, we propose MuKI-Fi, the first multi-person KI system, to steal passwords by eavesdropping on keystroke-induced BFI (*beamforming feedback information*) variations. In order to overcome the impracticality of using CSI, we pioneer the use of such a new feature that is piggybacked by new Wi-Fi hardware (starting from Wi-Fi 5 [28]) and trasmitted in clear-text on control frames. Thanks to BFI's clear-text nature, no low-level hacking is needed on Wi-Fi hardware. Basically, in the form of compressed *digital* version of *analog* CSI, BFIs are used to feed downlink channel states back to an access point (AP), for the sake of guiding AP beamforming [29]. Though they only account for part of the downlink CSIs concerning the AP side, the fact that on-screen typing directly impacts the Wi-Fi antennas (hence channels) right behind the screen (see Fig. 1) allows BFIs to contain sufficient information about keystrokes. Consequently, any device capable of overhearing Wi-Fi traffic (under the *monitor* mode [30]) may obtain BFIs for free: in fact, MuKI-Fi may even use a mobile device. As shown in Fig. 1, when the victim types password, our MuKI-Fi eavesdrops on the BFI and makes use of this new vulnerability to realize password inference without the need for hacking the constantly evolving Wi-Fi hardware.

Given the lack of linguistic structure in passwords, we follow the canonical way of identifying individual keystrokes, but we leverage a deep learning model with a natural segmentation as input to get rid of the artifacts introduced by rule-based segmentation and environment interference. Then, we exploit *adversarial learning* [31] to extract features relevant only to individual keystrokes; such a cross-domain training method is capable of generalizing keystroke inference to unseen scenarios with limited amount of training data.

For multi-person scenarios, MuKI-Fi exploits two fundamental factors in such a realistic multi-user communication setting shown in Fig. 1: i) each AP-smartphone link uniquely identifies the victim's keystroke to be sensed, and ii) since the victim's finger is within the near-field (less than 0.1m in range) of his/her own Wi-Fi device, the channel variation caused by the keystroke to the AP-smartphone link could be so strong as to push the interference from other keystrokes down to the noise floor. Thus, MuKI-Fi explores the potential of the default multi-user communication setting of Wi-Fi and naturally applies it to multi-person KI.

In summary, our main contributions are:

- We propose MuKI-Fi as the first WiFi-based *multi-person* keystroke eavesdropping system; leveraging the clear-text BFI, it allows a wide range of Wi-Fi devices to eavesdrop on passwords at ease.
- We innovate in leveraging adversarial learning to remove environment dependencies, rendering MuKI-Fi's inference model generalizable to unseen scenarios.
- We exposes the potential of multi-user communication system for sensing; it is exploited by MuKI-Fi to achieve multi-person KI.
- We implement a prototype of MuKI-Fi and conduct extensive evaluations. The results indicate that MuKI-Fi achieves 87.1% accuracy for identifying single numerical keys, and a top-100 accuracy of 81% for inferring a 6-digit numerical password.

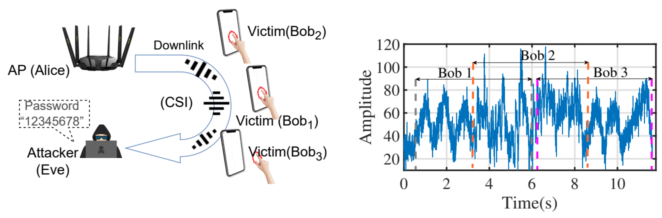
The paper is structured as follows. Sec. 2 introduces the background and motivation of our work. Sec. 3 presents the attack design of MuKI-Fi in detail. Sec.s 4 and 5 respectively explain MuKI-Fi's implementation and report the extensive evaluations on MuKI-Fi, followed by a discussion on extension from numerical to general keystroke inference. Related works are briefly captured in Sec. 6. Finally, we conclude our paper in Sec. 7.

2 Background and Motivation

In this section, considering MuKI-Fi is based on Wi-Fi sensing, we first briefly introduce the basic principles of Wi-Fi human sensing. Next, we present our attack scenario of *multi-person* KI and identify the limitations of existing approaches in addressing this scenario. Finally, we demonstrate the advantages of BFI over CSI for realizing the attack.

2.1 Wi-Fi Human Sensing Basics

We start by introducing a Wi-Fi sensing system with an AP and *user equipment* (UE) pair aiming to sense the physical



(a) Existing method. (b) Mixed CSI of keystrokes.

Fig. 2. Existing CSI-based KI method: (a) system architecture and (b) entangled CSI signals of keystrokes from different users.

motion of a victim denoted by V. Additionally, there are $N-1$ other individuals denoted as I_j $(\forall j \in 1, ...N-1)$ within the system, serving as interference to V. At a certain time t, we can model the wireless channel gain between the AP and the UE as:

$$\tilde{h}_{A,E}(t) = h_{A,V,E}(t) + \sum_{j=1}^{N-1} h_{A,I_j,E}(t) + h^S_{A,E} + h^D_{A,E}(t), \quad (1)$$

where $h^S_{A,E}$ and $h^D_{A,E}(t)$ represent the static and dynamic channel gains between the AP and UE due to, respectively, the direct communication path and interfering motions along it, and $h_{A,V,E}(t)$ and $h_{A,I_j,E}(t)$ indicates the channel gain from the AP to UE via the reflection of V and I_j respectively. $h_{A,V,E}(t)$ can be expressed as:

$$h_{A,V,E}(t) = \frac{\lambda^2 \sqrt{G_{A,V,E}} e^{-i2\pi(d_{A,V}(t)+d_{V,E}(t))/\lambda}}{(4\pi)^2 (d_{A,V}(t) d_{V,E}(t))^{\alpha/2}}, \quad (2)$$

where λ is the carrier wavelength, $G_{A,V,E}$ represents the product of Tx and Rx antenna gains and the reflection coefficient of V, $d_{A,V}(t)$ denotes the distance between the AP, $d_{V,E}(t)$ denotes the distance between V and the UE and α is the path loss exponent [32]. Typically, $\alpha \in [2,4]$ with $\alpha \approx 4$ for indoor environments [33]. $h_{A,I_j,E}(t)$ can also be modeled in a similar manner as Eqn. (2). To this end, Wi-Fi human sensing can be described as follows: the physical motion of a human subject results in changes of $d_{A,V}(t)$ and $d_{V,E}(t)$, which in turn lead to the changes of channel gain $h_{A,V,E}(t)$ over time. Therefore, if other interference items can be reasonably handled, by analyzing the time series of $\tilde{h}_{A,E}(t)$ obtained from the Wi-Fi frames, both AP and UE are able to sense the motion of V, which can be subsequently utilized for KI purposes.

2.2 Attack Scenarios and Existing Methods' Failure

We consider a scenario where some potential victims, Bobs (as shown in Fig. 1, there are 3 Bobs), use their mobile devices (smartphones or tablets) to connect to a nearby Wi-Fi assess point (AP) with a shared password or even no password protection; this is a reasonable assumption in public places such as shopping malls, office buildings, airports, and restaurants, because such a Wi-Fi access is often provided for the convenience to customers or visitors. After connecting to the AP for accessing the Internet, one Bob happens to have the need to access a sensitive account (e.g., online payment) protected by a *password*[3]. The keystroke of typing the password by the Bob induces changes in the channel gain $h_{A,V,E}(t)$ over time, which makes him become the victim, a target of attack launched by the attacker Eve (see Fig. 1). It is important to note that in this multi-person attack scenarios, any of the Bobs can become a target at any given time. This significantly enhances both the practicality and the degree of risk associated with MuKI-Fi.

However, existing methods proposed in [17] is not suitable for multi-person attack scenarios. This method employs a system architecture depicted in Fig. 2(a), which requires Eve to create a separate channel irrelevant to Bob, using Eve's Wi-Fi NIC and another device (e.g., an AP). Eve then infers Bob's keystrokes by observing the CSIs of this channel. Fig. 2(b) illustrates the CSI waveforms collected using the existing system architecture, where in a three-person scenario, each individual inputs a six-digit password. It is evident that the CSI waveforms of each individual's password are entangled and indistinguishable, which demonstrates the failure of such methods in multi-person scenarios. Another endeavor, as described in [16], imposed more stringent prerequisites; it demands hacking the AP in Fig. 1 in the hope of tricking users into connecting to this rogue AP and then extracting the keystroke-introduced CSI. This approach may not be feasible in practice, because while the feasibility of hacking the continuous evolving Wi-Fi NICs cannot be guaranteed (see Sec. 1), effectively deploying rogue AP has been made extremely challenging due to the increasing alerts raised by individuals and companies on such attacks [34], [35], [36].

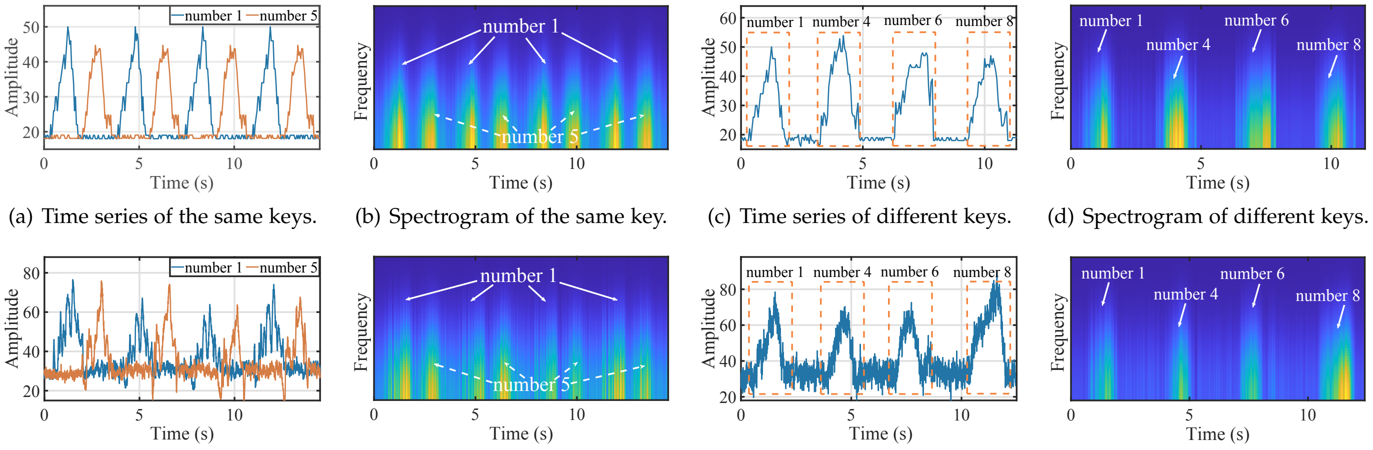
2.3 Why BFI instead of CSI?

BFI actually offers other advantages over CSI in terms of KI attack, apart from its easy acquirement explained earlier. To be specific, BFI behaves *less sensitive* to channel variation than CSI, rendering the sensing outcome more *stable* especially upon the close impact (from on-screen keystrokes) on Wi-Fi channels. This stability stems from the way BFI is generated. Given the downlink CSI represented as $H = Y/X$, where X and Y respectively denote the transmitted (Tx) and received (Rx) signals [16], BFI is generated by partitioning H (hence the channels it represents) into separated Tx and Rx components; only the Tx component is fed back to the AP for guiding AP beamforming [29].

To showcase the superiority of BFI over CSI in KI, Fig.s 3(a) and 3(b) respectively depict the BFI time series and spectrograms for clicking numerical keys '1' and '5' four times. One may readily observe that the BFI patterns remain consistent for clicking the same keys at different times, while the distinctions between two keys are also pronounced. Additionally, Fig.s 3(c) and 3(d), presenting BFI time series and spectrograms for clicking four different keys, again confirm the remarkable distinctions across these keys. In short, BFI is well-suited for KI with minimal preprocessing.

As a comparison, Fig.s 3(e) and 3(f), with contents similar to 3(a) and 3(b) but for CSIs collected simultaneously with the aforementioned BFIs, fail to indicate either remarkable consistence for the same key or pronounced distinctions between two different keys. Meanwhile, the four-key tests shown in Fig.s 3(g) and 3(h) also suggest the need for some heavy denoising before using CSIs for KI, as the distinctions between certain keys (e.g., '4' and '6') appear to be overwhelmed by noises. We suspect that

[3] We follow the convention [16], [17] to mainly focus on numerical passwords, but we also consider an extension to general KI in Sec. 5.4.2.



(a) Time series of the same keys. (b) Spectrogram of the same key. (c) Time series of different keys. (d) Spectrogram of different keys.

(e) Time series of the same keys. (f) Spectrogram of the same key. (g) Time series of different keys. (h) Spectrogram of different keys.

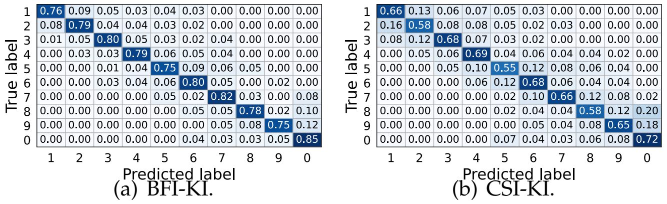
Fig. 3. BFI-KI (a)-(d) vs. CSI-KI (e)-(h): whereas BFIs exhibit both consistency for the same key and distinction for different keys, CSI's irregular patterns may cause ambiguities for keystroke inference.

such noises cannot be easily eliminated using conventional signal processing techniques, since their wide spectrum may confuse themselves with CSI features, as confirmed by the following KI test with denoised CSI and raw BFI.

We collect both BFI and CSI samples from 20 subjects typing numerical keys '0' to '9'. The same denoising technique in [16] is applied to the CSI samples, while the BFI samples are kept raw. We then use a one-dimensional convolutional neural network (1-D CNN) [37] to perform classification for the sake of KI and evaluate the KI accuracy by cross-validation. Fig.s 4(a) and 4(b) present the confusion matrices for BFI- and CSI-based KIs, respectively; these results evidently demonstrate that BFI achieves higher accuracy for individual keys, as indicated by the diagonal of the confusion matrix. Overall, the average accuracy achieved by BFI is 78.9%, notably higher than 64.5% achieved by CSI, hence clearly confirming the benefit of BFI's stability over even denoised CSI in terms of realizing KI.

3 The design of MuKI-Fi

In this section, we introduce the attack strategy of MuKI-Fi. As shown in Fig. 5, we present single-person KI workflow which consists of four steps: i) identifying the victim, ii) determining the attack time when the victim accesses the targeted application service, iii) capturing and parsing the victim-associated BFI time series, and finally iv) segmenting the BFI series and performing KI to recover the intended password. Then, we conduct a theoretical analysis of MuKI-Fi in multi-person scenarios in Sec. 3.4.



(a) BFI-KI. (b) CSI-KI.

Fig. 4. Confusion matrices for BFI- vs. CSI-based keystroke inference, demonstrating the superiority of BFI over CSI in completing this task.

3.1 Victim Identification and Attack Timing

Following an implicit assumption of [16], we also allow Eve to have prior knowledge of Bob's device identity (e.g., MAC address). In reality, Eve can acquire this information beforehand by conducting visual and traffic monitoring concurrently: correlating network traffic originating from various MAC addresses with users' behaviors should allow Eve to link Bob's physical device to his digital traffic, thereby identifying Bob's MAC address. It is worth noting that victim identification is not possible in [17] since Bob's device does not communicate with the AP or Eve.

Once locked onto Bob's MAC address, Eve waits for the right time (when Bob is about to enter his password) to launch attack. This timing issue can be readily addressed if visual hints are presented (e.g., Bob scan the WeChat Pay QR code or Bob's screen shows the payment page); otherwise, Eve can inspect the requests made to a payment service. Consider the case of WeChat [38], though most of its traffic is secured via application-layer encryption [39], IP addresses are not encrypted for the public Wi-Fi networks targeted by MuKI-Fi. To exploit this vulnerability, Eve creates a database of IP addresses associated with the payment service: though such IP addresses can be dynamic, our experiments reveal that users from the same region are directed to the same IP address within a certain period. Subsequently, upon detecting an IP address recorded in the database, the attack can be launched; the recording of BFI series will be stopped once no more requests to the IP can be observed.

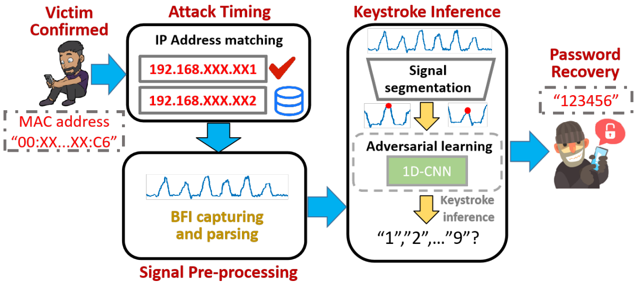


Fig. 5. The workflow of MuKI-Fi's single-person attack strategy.

3.2 BFI Analysis and Parsing

We hereby provide more details on how password typing can manifest in BFI to facilitate later developments. As explained in Sec. 2.3, BFI is the Tx component of CSI H and is fed back to guide AP beamforming. This is accomplished by SVD (singular value decomposition) [40] that decomposes the channel as $H = USV$. Among these components, only the right matrix V is chosen as BFI, while the other two matrices U and S (representing Rx beamforming and channel gains, respectively) are not. As illustrated in Fig. 6, Bob's password typing affects the diffraction pattern of the Wi-Fi signals around the phone body. This altered pattern is then reflected in downlink CSI that is in turn decomposed with SVD to obtain BFI V.

As BFI is transmitted in clear text, Eve can easily intercept it using a Wi-Fi device in monitor mode, along with Wireshark [41]. The frame structure of 802.11ac can be followed to locate BFI in the "VHT beamforming report" field within the Wi-Fi Action frames [29]. To completely extract BFI, the length of the field can be calculated based on the number of Tx and Rx antennas, as outlined in [28]. By continuously recording the BFIs in the Wi-Fi frames from Bob during the time window of Bob's password typing, Eve can obtain a time series of BFI samples correlated with the password.

3.3 Keystroke Inference

In this section, we elaborate on how MuKI-Fi conducts BFI-KI. We first discuss the drawbacks of previous proposals and explain possible improvements upon them. After that, we specify the signal segmentation on BFI series to kick off KI, which is then followed by the design of the KI neural model and its adversarial learning framework to generalize KI towards unseen scenarios.

3.3.1 What's Wrong with Prior Art?

Two previous works have exploited Wi-Fi side-channels to steal passwords. The seminal proposal of WindTalker [16] performs classification upon individual keystrokes with rule-based CSI series segmentation. Intuitively, such segmentation should not perform well because it can result in information loss or introduce artifacts. To confirm this suspicion, we ask two subjects to type passwords on their respective smartphones, and Fig. 7(a) shows their corresponding CSI series. Apparently, the duration of the keystrokes and the amount of overlap between them vary significantly due to the subjects' distinct typing habits. While rule-based segmentation may be effective for Subject A who types more steadily, it most likely fails for Subject B whose inter-keystroke patterns appear rather messy. In attempting to forcibly assign different sections of the BFI series to individual keys, the segmentation process introduces artifacts (e.g., clipping) to each keystroke, potentially harming the KI performance.

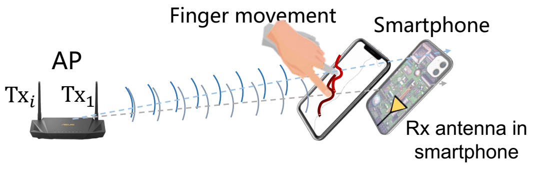
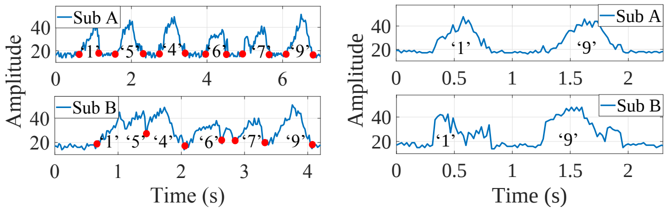


Fig. 6. Finger movements cause diffraction on the downlink path, which is manifested in BFI variations.



(a) Segmentation ambiguity. (b) Variance in transition features.

Fig. 7. Two cases where previous methods fail.

A recent proposal WINK [17] claims to improve the KI performance via series learning. However, it inherits the rule-based segmentation adopted by [16] and hence the same weakness too. Additionally, as linguistic structure cannot be exploited for series learning, WINK argues that transition features between keystrokes may serve as replacements for improving KI accuracy. Unfortunately, factors such as typing habits and smartphone types can affect CSI during the transition period, resulting in different features for the same password. To illustrate this, we ask two subjects to type two keys '1' and then '9' on their phones, and Fig. 7(b) shows significant morphological and temporal differences in these two transitions. Therefore, it is very questionable if transition features can ever replace linguistic structure.

To overcome the disadvantages in previous proposals, the rule-based segmentation needs to be replaced with a more intelligent method, preferably a data-driven one. Also, as using transition features to replace linguistic structure cannot be reliable, MuKI-Fi falls back to the canonical approach of inferring individual keystrokes as executed by WindTalker. To prevent information loss in segmentation. MuKI-Fi deems the environment-dependent transition periods as different "domains" of the same numerical keystroke. Consequently, an adversarial learning is exploited to train the KI model, aiming to remove domain interference (i.e., environment dependency) and hence generalize KI to unseen scenarios. Note that the data-driven nature of MuKI-Fi also prevents it from taking a series learning perspective, as it would otherwise demand a prohibitively large training dataset whose size grows exponentially with the password length.

3.3.2 Signal Segmentation

In reality, BFI series may not show distinct boundaries between consecutive keystrokes, significantly complicating signal segmentation. Fig. 8 provides an example for such a case, where the BFI series displays prominent peaks corresponding to Bob's finger hitting the screen, as well as fluctuations between two peaks representing the transition movement of his fingers. Since the transitions carry information about both the preceding and succeeding keystrokes, segments of neighboring keystrokes should contain the transition. Therefore, we propose to employ an overlapping segmentation method that incorporates all data samples located between two consecutive peaks, instead of the non-

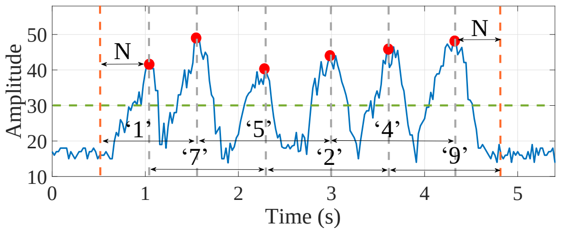


Fig. 8. Signal segmentation with overlaps.

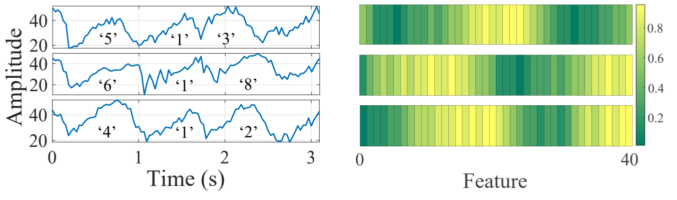
overlapping segmentation achieved by windows of rule-defined sizes [5], [17].

Our segmentation method starts with utilizing the Constant False Alarm Rate (CFAR) algorithm [42] to identify peaks in a BFI series. Suppose Bob typing a K-digit numerical password to produce a BFI series of length L, the CFAR algorithm conducts statistical analysis on the series to determine an adaptive threshold and selects the peaks exceeding this threshold as the targets. Among these target peaks, we further select the top-K peaks corresponding to the K numbers in the password, assisted by an inter-peak distance of W sampling points, where $W = \alpha \times \frac{L}{K}$. For each peak, we include all the data samples between itself and its two neighboring peaks into the segment corresponding to a single keystroke; since the first and last numbers in the password have no preceding and succeeding numbers, we choose to extend N points before and after as the segment boundaries, where $N = \beta \times \frac{L}{K}$. We shall empirically determine the values of α and β in Sec. 4. As demonstrated in Fig. 8, this approach effectively partitions a BFI series (for password "175249") into segments corresponding to individual keystrokes, while preserving the feature-rich transitions between keystrokes caused by finger movements.

3.3.3 Adversarial Learning Framework

This section explains how adversarial learning is employed to generalize KI to unseen domains. Prior to that, we briefly describe the basic design of KI network. The classification of time series is a well-established task that can be effectively addressed using a 1-D CNN. However, as discussed in Sec. 3.3.2, the BFI segments may differ in length, posing a challenge to conventional 1-D CNNs. To overcome this issue, we employ an adaptive average pooling layer [43] to enhance the flexibility of 1-D CNNs. To be specific, this layer automatically calculates the appropriate kernel size required to yield a fixed-size output feature map, thus enabling 1-D CNNs to accommodate inputs of varying lengths.

In fact, the direct deep learning approach mentioned above overlooks the impact of the domain on each keystroke. Here *domain* refers to the context arising from the diversified transitions from the preceding and to the succeeding keystrokes; it includes the distinctions caused by, for example, typing speed, inter-typing irregularities, and the adjacent keystrokes. To illustrate this, we consider the numerical key '1' in three different domains: '5-1-3', '6-1-8', and '4-1-2', and present their segments and corresponding feature maps in Fig. 9. Although the segments of key '1' under different domains, in Fig. 9(a), exhibit a high degree of similarity near the peak, the '1' in '6-1-8' displays drastic fluctuations during transitions between neighboring keystrokes, while those in '5-1-3' and '4-1-2' have rather smooth transitions. Such differences can be attributed to larger channel variations induced by finger movements over greater distances between the keys in '6-1-8 '. Additionally, we show the feature heatmaps for different '1's after the adaptive average pooling layer in Fig. 9(b): the same key '1' in different domains exhibit distinct feature maps, thus posing significant challenges to the subsequent keystroke classifier.



(a) BFI segments.

(b) Feature maps.

Fig. 9. Difference in BFI segments and features maps of key '1' indicates the domain dependency of KI.

The aforementioned domain interference entails the need for a method ensuring KI's invariance to such interference, so we employ the idea of domain adaptation [44] to learn keystroke representations invariant across different domains. Given the complexity of BFI segment features due to the diversity of inter-keystroke transition patterns, employing an explicit feature space transformation as in [45] could be challenging. Instead, MuKI-Fi aims to achieve a consistent feature space representation in different domains, by harnessing the power of *adversarial learning* [31] to integrate domain adaptation with KI in a unified training process. To incorporate adversarial learning, we revamp the training strategy of 1-D CNN as illustrated in Fig. 10, whose training and inference processes are introduced as follows.

During the *training* phase, we first prepare a dataset consisting of randomly *paired* BFI segments corresponding to the same key (e.g., '1') but under different domains, e.g., two '6-1-8' from different passwords or a pair of '4-1-2' and '5-1-3'. We concatenate the pair as input $\boldsymbol{x}$ and process them through the feature extractor G_{f}. The resulting features are then fed into both the keystroke classifier G_{c} and domain discriminator G_{d}: G_{c} infers the key y shared by both segments within the pair, and G_{d} predicts the *domain discrepancy* $\Delta \in \{0, 1\}$, with 0 and 1 denoting the keys from the two segments *are* and *are not* from the same domain, respectively. While G_{d} aims to improve the accuracy of predicting Δ, the adversarial learning strategy

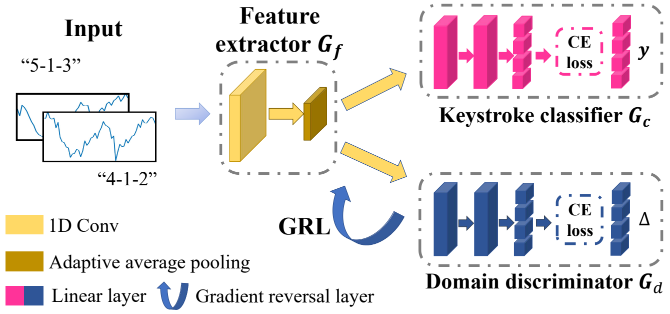


Fig. 10. The training strategy enabled by adversarial learning removes domain-specific information.

"cheats" G_d by inverting its loss via reversing the gradient during backpropagation using the Gradient Reversal Layer (GRL) [46]; this procedure tends to suppress domain-specific features from the output of G_f and thus allows the 1-D CNN to learn keystroke representations invariant across domains. Denoting the parameters of G_f, G_c, and G_d as θ_f, θ_c, and θ_d, respectively, the training procedure can be formulated as:

$$(\hat{\theta}_\mathrm{f}, \hat{\theta}_\mathrm{c}) = \arg\min\nolimits_{\theta_\mathrm{f},\theta_\mathrm{c}} \mathcal{L}(y, \Delta, \boldsymbol{x}), \quad \hat{\theta}_\mathrm{d} = \arg\max\nolimits_{\theta_\mathrm{d}} \mathcal{L}(y, \Delta, \boldsymbol{x}),$$
$$\mathcal{L}(y, \Delta, \boldsymbol{x}) = \mathcal{L}_\mathrm{c}\left(y, G_\mathrm{c}(G_\mathrm{f}(\boldsymbol{x}))\right) - \lambda \mathcal{L}_\mathrm{d}\left(\Delta, G_\mathrm{d}(G_\mathrm{f}(\boldsymbol{x}))\right)$$

where $\mathcal{L}_\mathrm{c}$ and $\mathcal{L}_\mathrm{d}$ are respectively the cross-entropy losses for G_c and G_d, and λ, a balance factor controlling the trade-off between $\mathcal{L}_\mathrm{c}$ and $\mathcal{L}_\mathrm{d}$, should have its value empirically determined in Sec. 4. G_d is discarded during the *inference* phase, and the input of segment pair $\boldsymbol{x}$ is emulated by replicating the original BFI segment.

3.4 How MuKI-Fi works in multi-person scenarios?

In the context of multi-person scenarios, each victim is equipped with a mobile device (i.e.,UE). Consequently, for each victim, there exists a distinct Access Point (AP)-UE pair, which corresponds to its specific channel gain $\tilde{h}_{A,E}(t)$. For illustrative purposes, we select one of the victims along with their associated UE for further discussion and the corresponding $\tilde{h}_{A,E}(t)$ follows Eqn. (1). Considering V is close to or in the near-field of his own UE (i.e., distance below 0.2 m, empirically), the variation of the channel gain is dominated by the V's physical motion; in other words, $\partial|h_{A,V,E}(t)|/\partial t \gg \partial|h_{A,I_j,E}(t)|/\partial t$. We term this phenomenon **near-field domination** effect, and we provide its theoretical analysis as follows.

To quantify the variation of $h_{A,V,E}(t)$, we evaluate it by the squared amplitude of the partial derivative of $h_{A,V,E}(t)$ w.r.t. t, which is referred to as the *power of channel variation*. To simplify the analysis, we assume $\partial d_{A,V}(t)/\partial t = \partial d_{V,E}(t)/\partial t = v_V$. The value of v_V can be interpreted as the *intensity* of V's motion in terms of speed. The power of channel variation of V can then be calculated as:

$$\begin{aligned} P_V &= \left|\frac{\partial h_{A,V,E}(t)}{\partial t}\right|^2 \\ &\approx \frac{G_{A,V,E}\lambda^4 v_V^2}{(4\pi)^4 (d_{A,V} d_{V,E})^\alpha} \left[\frac{\alpha^2}{4}\Big(\frac{d_{A,V} + d_{V,E}}{d_{A,V} d_{V,E}}\Big)^2 + \frac{16\pi^2}{\lambda^2}\right] \\ &\overset{(\star)}{\approx} \tilde{G}_{A,V,E} \cdot v_V^2 \cdot (d_{A,V} d_{V,E})^{-\alpha}, \end{aligned} \tag{3}$$

where we omit symbol t in the distance notations and let $\tilde{G}_{A,V,E} = (\lambda/4\pi)^2 G_{A,V,E}$ for the sake of brevity. $(\star)$ holds because, in typical 5 GHz Wi-Fi sensing systems with V in the near-field of UE (e.g., $d_{A,V} \sim 3\,\mathrm{m}$, $d_{V,E} \sim 0.1\,\mathrm{m}$, and $\lambda \sim 0.06\mathrm{m}$), the first term inside the bracket is much smaller than the second term and thus can be omitted.

In order to quantify the influence of the number and location of interferers on the domination of P_V at the UE, we propose an novel metric **variation to interference ratio** (VIR); it evaluates the variation power ratio between $h_{A,V,E}(t)$ and the sum of $h_{A,I,E}(t)$ and dynamic channel $h^D_{A,E}(t)$. Based on [47], $h^D_{A,E}(t)$ can be also treated as an interference, whose power P_d is in linear proportion to that of the static channel gain. Therefore, assuming a LoS path between the AP and UE, we have $P_d = \eta\lambda^2 d_{A,E}{}^{-\alpha} + b$, where η and b are fixed for a given pair of AP and UE. Then, we have:

$$\mathrm{VIR}_V = \frac{P_V}{P_I + P_d} = \frac{v_V^2 \tilde{G}_{A,V,E} (d_{A,V} d_{V,E})^{-\alpha}}{\eta\lambda^2 d_{A,E}^{-\alpha} + b + v_I^2 \tilde{G}_{A,I,E} (d_{A,I} d_{I,E})^{-\alpha}}. \tag{4}$$

Intuitively, the number and location of interferers are indicated by VIR_V value being greater than a threshold γ_th. To this end, we consider two symmetric distribution cases, i) how many people can multiperson MuKI-Fi support? and ii) how close can adjacent people be? To simplify the presentation, we no longer distinguish between the positions of the victim and its UE.

To answer question i), we analyze the case where victim V and $N-1$ other victims stay at distance r from the AP and are uniformly spaced, as shown in Fig. 11(a). After extending Eqn. (4) and normalizing v_V, v_{I_j} to 1, we have:

$$\mathrm{VIR}_V^{(\mathrm{i})} = \frac{\tilde{G}(\Delta r)^{-\alpha}}{\eta\lambda^2 + br^\alpha + (2r)^{-\alpha}\tilde{G} \cdot \sum_{j=1}^{N-1} \sin^{-\alpha}(j \cdot \pi/N)}, \tag{5}$$

where $\tilde{G}$ represents the identical values of $\tilde{G}_{A,V,E}$ and $\tilde{G}_{A,I_j,E}$, and Δr denotes the short distance from V to its UE. Generally, the series summation in Eqn. (5), $\sum_{j=1}^{N-1} \sin^{-\alpha}(j \cdot \pi/N)$, has no closed-form expression w.r.t. N. Fortunately, given $N \in [3, 60]$ and $\alpha \in [2, 4]$, the series summation can be numerically fitted by a function in the form of $p_1 N^{p_2} + p_3$ with R-square ≈ 1, where parameters p_1, p_2, and p_3 are dependent on α: $p_1 = 0.0230$, $p_2 = 3.99$, $p_3 = 38.0$ for $\alpha = 4$. Now given γ_th, the **upper bound** on the number of subjects that can be accommodated becomes:

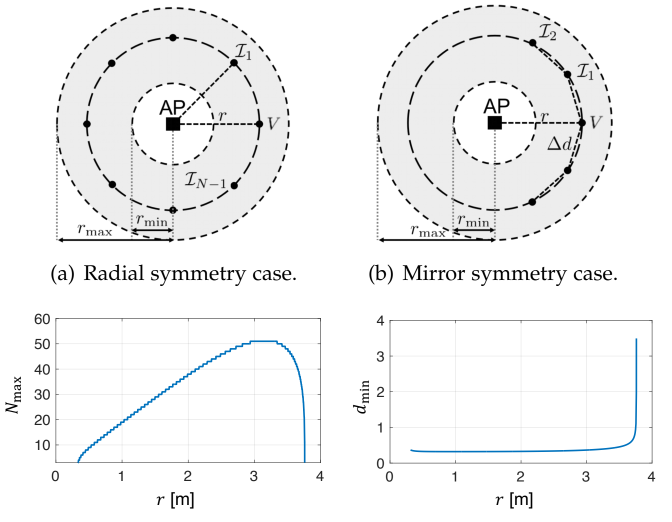
$$N_\mathrm{max} = \left\lfloor \left(\frac{(2r)^\alpha}{p_1} \cdot \frac{\tilde{G}(\Delta r)^{-\alpha} - \eta\lambda^2\gamma_\mathrm{th} - br^\alpha\gamma_\mathrm{th}}{\tilde{G}\gamma_\mathrm{th}} - \frac{p_3}{p_1}\right)^{\frac{1}{q_2}} \right\rfloor. \tag{6}$$

Moreover, based on (6), given v_V, v_{I_j}, η, b, $\tilde{G}$ being normalized to 1, $\Delta r = 0.015m$ and $\alpha = 4$, we can also derive the minimum and maximum distances (resp. r_min and r_max) between tar and the AP for the considered case to be feasible by solving the inequality $N_\mathrm{max} \geq 3$ in the field of real number. As shown in Fig. 11(c), N_max first increases and then decreases in r; it reaches its maximum 51 when $r \in [2.94, 3.35]$ m.

To answer question ii), we consider the case as shown in Fig. 11(b), where $2K(K = 1, 2, 3, \ldots)$ interferers are evenly distributed on both sides of the victim(then, $N = 2K + 1$). Denote the distance and angle between each pair of neighboring people by Δd and ϕ, we have $\phi = 2 \cdot \arcsin(\Delta d/(2r))$. It is clear that the middle victim suffers from the worst interference (among all people) when $2\pi - (2K + 1)\phi > 0$, equivalent to $\Delta d < 2r\sin(\pi/(2K + 1))$. Using Eqn. (4) again, we have:

$$\mathrm{VIR}_V^{(\mathrm{ii}} = \frac{\tilde{G}\Delta r^{-\alpha}}{\eta\lambda^2 + br^\alpha + 2\tilde{G}(2r)^{-\alpha}\sum_{j=1}^{K} \sin^{-\alpha}(j\phi/2)}. \tag{7}$$

Again, the series summation $\sum_{j=1}^{K} \sin^{-\alpha}(j\phi/2)$ can be fitted by function $q_1(\sin(\phi/2))^{q_2} + q_3$ given $\alpha \in [2, 4]$, $K \in [1, 10]$, and $\phi \in [\pi/180, \pi/(2K + 1)]$ with R-square ≈ 1, where parameters q_1, q_2, and q_3 depend on K and α: $q_1 = 1.06$,



(a) Radial symmetry case.

(b) Mirror symmetry case.

(c) $N_{\max}$ vs r.

(d) $d_{\min}$ vs r.

Fig. 11. Two symmetric cases considered for multi-person sensing scenarios. (a)&(c) N subjects uniformly spaced and (b)&(d) $2K+1$ subjects closely located.

$q_2 = -4$, and $q_3 = 6.57$ for $\alpha = 4$ and $K = 2$. Consequently, we obtain the **lower bound** of Δd as follows:

$$\Delta d_{\min}= 2r\left(\frac{(2r)^{\alpha}}{q_1}\cdot\frac{\tilde{G}(\Delta r)^{-\alpha}-\eta\lambda^2\gamma_{\text{th}}-br^{\alpha}\gamma_{\text{th}}}{2\tilde{G}\gamma_{\text{th}}}-\frac{q_3}{q_1}\right)^{\frac{1}{q_2}}. \quad (8)$$

Furthermore, by solving $\Delta d_{\min} \leq 2r\sin(\pi/(2K+1))$ in the field of real number, we can obtain the boundary for the distance between the AP and the subjects, i.e., $r_{\min}$ and $r_{\max}$. As shown in Fig. 11(d), $\Delta d_{\min}$ remains around 0.34 m for $r \in [0.32, 3.30]$ m but increases steeply to 3.49 m for $r \in [3.30, 3.76]$ m.

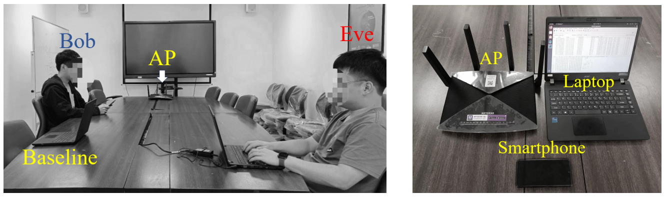
Through the above analysis, we can deduce that in a multi-person scenarios, the near-field channel variation(i.e. the variation of BFI) caused by the victim significantly overwhelms variations caused by other distant Bobs. Thus, this dominating effect of near-field sensing illustrates that MuKI-Fi can implement KI for multiple victims, and where they are close to each other.

4 Implementation and Setup

In this section, we elaborate on MuKI-Fi's implementation, as well as introduce the experiment setup and metrics.

4.1 System Implementation

Though a rooted smartphone under the monitor mode can act as Eve, Android systems offer minimal support in capturing Wi-Fi traffic at application layer. Therefore, we focus on a laptop implementation in our experiments. We use an Acer TravelMate laptop [48] with an Intel AX210 Wi-Fi NIC [49] supporting 802.11b/g/n/ac as the basis; setting the NIC to the monitor mode, we then use WireShark to capture the BFI series contained in Action No-ACK frames. The captured BFIs are analyzed using Matlab and Python, with the neural models built upon PyTorch 1.7.1 [50]. For the segmentation, the two parameters α and β are set to 0.6 and 0.5, respectively. In the adversarial learning framework, the balance factor λ is set to 0.5.



(a) Experiment scene.

(b) Adopted hardware.

Fig. 12. Evaluative MuKI-Fi: (a) experiment scene in a conference room and (b) hardware configurations.

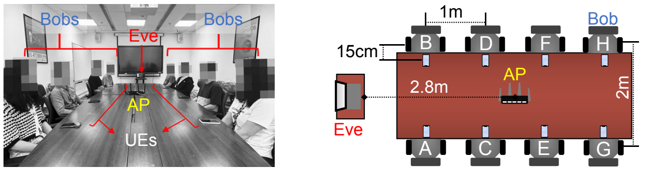
4.2 Experiment Setup

We recruit 20 subjects, of 12 males and 8 females, between the ages of 20 and 53. All subjects are right-handed and use their own smartphones of various models, including iPhone 13 [51], OnePlus 10T [52], Xiaomi 13 Pro [53], Huawei P40 Pro [54], and Samsung Galaxy S20 [55]. The subjects type a total of 1,500 predefined passwords of 4, 6, and 8 digits, with each length having 5,000 passwords. During typing, background apps remain active to emulate daily smartphone usage. The subjects adopt different postures while typing on the smartphones, such as holding it with one or both hands or placing it on a stand or table. The typing speed of the subjects ranges from 0.5 to 2 cps (*characters per second*).These experiments have strictly followed our IRB.

We conduct experiments and collect BFI series in six environments, including a library, bookstore, auditorium, cafeteria, corridor, and conference room. In each environment, a Wi-Fi router working as an AP for the subjects to connect. Besides BFI collection, we simultaneously obtain CSIs from the AP and another laptop to respectively serve as comparison baselines of WINK [17]. The distance between a subject and the AP ranges from 1 to 10 m, and the distance between the attacker and the subject ranges from 3 to 10 m. Fig. 12 shows an example experiment scene and the hardware we use. For multi-person scenarios, Fig. 13(a) illustrates our experiment setup, where the AP, subjects, UEs, and baseline device are all exhibited and annoted in Fig. 13(b). MuKI-Fi segments the BFI series using the overlapping scheme described in Sec. 3.3.2, while the baseline conducts its rule-based segmentation. We use 70% of the collected data for training and the remaining 30% for testing.

4.3 Metrics

We adopt two metrics for our evaluations, namely keystroke *classification accuracy* and top-N password *inference accuracy*. For single keystroke classification, the classification accuracy measures the percentage of correctly classified keystrokes. For password inference, since an attacker may try multiple passwords to increase the success rate, we



(a) Multi-person experiment scene.

(b) Layout.

Fig. 13. Experiment scene (a) and layout of subject arrangement (b) for multi-person scenario

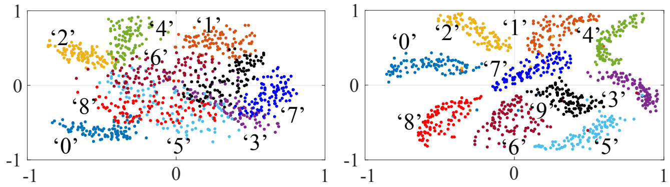
adopt the top-N accuracy as the evaluation metric: the probability of a candidate password is computed as the product of the probability of each key present in the password, then the top-N accuracy is measured by checking if any of the candidates within top-N probability matches the true one.

5 Evaluation

We start with micro-benchmark studies of domain adaptation to demonstrate the effectiveness of MuKI-Fi's adversarial learning framework. It is followed by evaluations on overall performance and the impact of practical factors. Finally, we conduct real-world experiments to showcase how MuKI-Fi steals passwords of WeChat Pay, while also extending it to general KI on QWERTY keyboards.

5.1 Domain Adaptation Micro-benchmark

To demonstrate the effectiveness of MuKI-Fi's adversarial learning framework in Sec. 3.3.3, we use t-SNE (t-Distributed Stochastic Neighbor Embedding) [56] to visualize the feature maps of 10 numerical keys segmented from 100 random passwords in Fig. 14. As shown in Fig. 14(a), the normal feature extractor G_f fails to find a domain-invariant feature map: features of different keys apparently get mixed together due to domain interference. In contrast, Fig. 14(b) demonstrates that, with adversarial learning, the features of the same keys are consistent across domains and form distinct clusters, indicating that domain-invariant representations have been successfully learned.



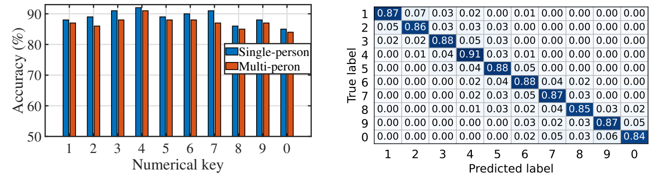
(a) w/o adversarial learning. (b) w/ adversarial learning.

Fig. 14. t-SNEs of the features output by the feature extractor G_f evidently confirm that adversarial learning results in domain-invariant representations.

5.2 Overall Performance

5.2.1 Classification Accuracy

In this section, we present the accuracy of classifying numerical keys of MuKI-Fi in single person scenarios and multi-person scenarios. We do not compare MuKI-Fi with baseline WINK [17] in terms of keystroke classification accuracy because WINK is based on series learning that predicts the password as a whole. As shown in Fig. 15(a), the classification accuracy of MuKI-Fi's two different scenarios for keys '0' to '9' remains steady at around 88.9% and 87.1%, respectively. To further analyze the classification accuracy for each key, we present the confusion matrix of MuKI-Fi in multi-person scenarios in Fig.s 15(b). It is intuitive that each key is most commonly confused with adjacent keys (e.g., the key '5' is most commonly confused with '2', '4', '6', and '8'). Despite the inevitable confusion, the high success rate of classifying individual keys lays a solid foundation for later password inference.

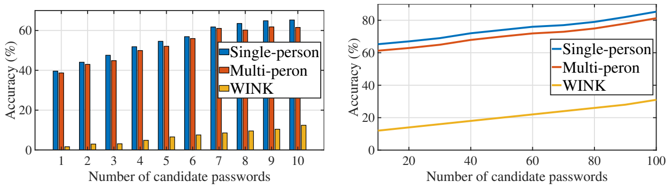


(a) Overall accuracy. (b) Confusion matrix.

Fig. 15. Comparing the classification accuracy of MuKI-Fi in single person scenarios and multi-person scenarios.

5.2.2 Password Inference Accuracy

Let us further evaluate MuKI-Fi's password inference capability, focusing on 6-digit numerical passwords due to their widespread usage in daily scenarios, but leaving the performance assessment for different password lengths to Sec. 5.3.6. Fig. 16(a) compares the top-1 to -10 accuracy of MuKI-Fi in single person scenarios and multi-person scenarios and WINK: while WINK only reach 12% for top-10 accuracy, MuKI-Fi's accuracy in single person scenarios varies from about 40% to 65% for top-1 to -10 candidates, with tiny deduction of its accuracy in multi-person scenarios. Fig. 16(b) indicates that MuKI-Fi's two accuracy can infer passwords with about an 85% success rate and 81% success rate in 100 attempts, yet WINK can only achieve a rate of 31% at the same number of attempts.



(a) Top-1 to top-10. (b) Top-10 to top-100.

Fig. 16. Comparison for password inference accuracy under different numbers of password candidates.

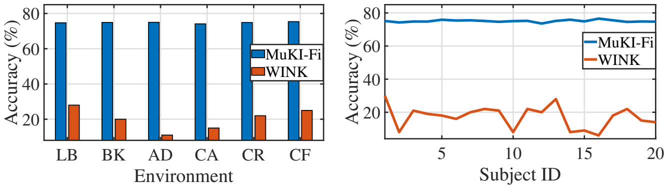
5.2.3 Performance Analysis

The superior performance of MuKI-Fi can be attributed to several reasons. As discussed in Sec. 2.3, BFI dampens the close impact from on-screen keystrokes, making MuKI-Fi more stable than CSI-based approaches. This allows MuKI-Fi to extract consistent features effectively learnable by its neural models. WINK, on the contrary, suffers from CSI noises possibly confused with useful features. Moreover, the overlapping segmentation technique proposed in Sec. 3.3.2 endows MuKI-Fi with richer domain "context" for its adversarial learning framework. In contrast, WINK's rule-based segmentation can not make full use of essential parts of the CSI features already overwhelmed by noises. MuKI-Fi has an edge over WINK because our design has a higher SNR, and the digital nature of BFI prevents fidelity loss of sensing signal. One may notice the performance discrepancy of WINK from that reported in [17]. This may stem from its design failing to properly take into account the influence of domain, thereby limiting its ability to effectively handle diverse data collected from various domains in our experiment setup. At the same time, our experiment of single-person and multi-person scenarios fully demonstrates the effectiveness and practicality of MuKI-Fi's near-field sensing to achieve multi-person KI. It is worth noting that the likelihood of each Bob's password being inferred can be measured by top-N accuracy. In multi-person scenarios, as the number of Bobs increases, the probability of at least one Bob's password being inferred significantly rises. This unequivocally underscores the fact that MuKI-Fi poses a severe threat in KI.

5.3 Impact of Practical Factors

5.3.1 Environments and Subjects

We use the "leave-one-out" strategy [57] to study the impacts of different environments and subjects. This means that the test set consists of all data from one of the 6 environments or one of the 20 subjects, leaving the rest to the training set. Fig. 17(a) and 17(b) respectively show the top-100 password inference accuracy for each environment and each subject. Although the testing environments and subjects are unseen during training, MuKI-Fi's top-100 accuracy across all cases is consistently above 75%, thanks to the generalizability of the adversarial learning. Moreover, MuKI-Fi is robust across environments since our design relies on the diffraction pattern around the phone body that are rarely influenced by environment-specific interference. In contrast, the average top-100 accuracy of WINK drops from that in Fig. 16(b) to less than 18%, due to its limited generalizability to unseen environments and subjects.

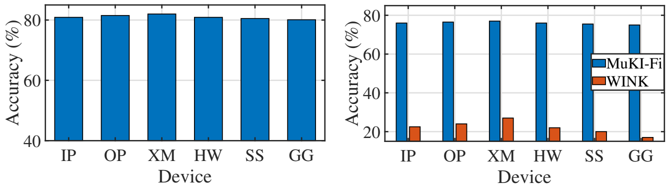


(a) Different environments. (b) Different subjects.

Fig. 17. Top-100 accuracy under the impact of different environments and subjects.

5.3.2 Device Diversity

We again use the "leave-one-out" strategy to evaluate the performance of MuKI-Fi on 6 smartphones specified in Sec. 4.2. Fig. 18(a) shows that MuKI-Fi can reliably identify keystrokes on different devices, with an average keystroke classification accuracy of over 80%. Furthermore, Fig. 18(b) indicates that the top-100 password inference accuracy of MuKI-Fi and WINK is respectively above 76% and below 27%. The consistently high accuracy of MuKI-Fi across different smartphone devices confirms that our adversarial learning framework can generalize to unseen devices. In contrast, the low accuracy of the baseline(evidently worse than the results in Fig. 16(b)) highlights its failure on unseen devices. One may also observe some accuracy variations among smartphones, which we attribute to different screen sizes. Specifically, MuKI-Fi achieves the highest accuracy on Xiaomi 13 Pro having the largest screen size (6.73 inch), while on Google Pixel 6a, with the smallest screen size (6.1 inch), it achieves the lowest accuracy. A possible explanation is smartphones with larger screens tend to have larger key distances that result in longer transition periods, thus making the incurred BFI features more distinguishable.



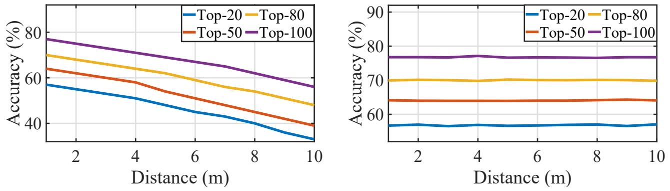
(a) Keystroke classification. (b) Password inference.

Fig. 18. Impact of device diversity.

Due to the consistently worse performance of the baseline, we do not compare MuKI-Fi with WINK in subsequent experiments.

5.3.3 Distance

We evaluate the effect of distances on MuKI-Fi, i.e., the distances from Bob to the AP and from Eve to Bob. Fig. 19 presents the top-20, 50, 80, and 100 password inference accuracy at various distances. Fig. 19(a) shows that the average accuracy decreases by about 23% as the distance between Bob and the AP increases from 1 m to 10 m, because a longer distance from Bob to the AP weakens the Wi-Fi signal and takes in more interference. On the contrary, Fig. 19(b) confirms that the distance between Eve and Bob barely affects the performance of MuKI-Fi, as the digital nature of BFI makes it robust to long-range transmission. Consequently, Eve can eavesdrop stealthily from a long distance without compromising inference accuracy, clearly demonstrating the advantage of BFI's digital nature of MuKI-Fi.

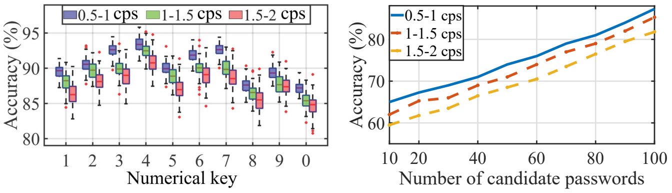


(a) Bob to the AP. (b) Eve to Bob.

Fig. 19. Impact of different distances.

5.3.4 Typing Speed

In this section, we examine how MuKI-Fi's performance varies with typing speeds. Fig.s 20(a) and 20(b) respectively show the keystroke classification and top-$[1, 100]$ accuracy for tying speed ranges of $[0.5, 1.0]$, $[1.0, 1.5]$, and $[1.5, 2.0]$ cps. As expected, both metric values decrease with higher typing speeds, probably due to stronger inter-typing irregularities. Nevertheless, MuKI-Fi still achieves sufficiently good performance in fast typing case with speed from $[1.5, 2.0]$ cps, with only a minor decrease of around 3%



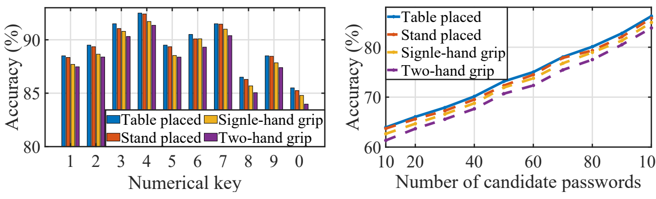
(a) Keystroke classification. (b) Password inference.

Fig. 20. Impact of typing speed.

in keystroke classification and less than 7% in password inference accuracy when compared with those in slow typing case with speed from $[0.5, 1.0]$ cps. The relatively consistent performance of MuKI-Fi across different typing speeds is also the consequence of adopting adversarial learning.

5.3.5 Typing Scenarios

We further investigate the performance of MuKI-Fi across different typing scenarios, including holding the phone with one or both hands and placing the phone on a stand or a table. Fig.s 21(a) and 21(b) show that when the smartphone is placed on a stand or a table, MuKI-Fi achieves higher keystroke classification and password inference accuracy, likely due to the stability inherent to these scenarios. Despite the accuracy differences across scenarios, keystroke classification and password inference accuracy variations are less than 2.5%. These consistent results demonstrate that MuKI-Fi is robust to various occlusions and different typing scenarios, further validating the effectiveness of our adversarial learning framework.

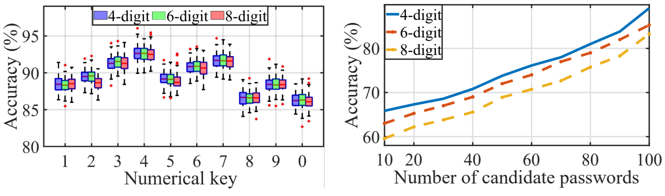


(a) Keystroke classification. (b) Password inference.

Fig. 21. Comparison of 4 different typing scenarios.

5.3.6 Password Length

We finally examine how password length affects MuKI-Fi's performance. Fig. 22(a) demonstrates that the password length does not affect the accuracy of keystroke classification because MuKI-Fi treats each keystroke independently regardless of how many keys are typed. However, it significantly impacts password inference accuracy, as shown in Fig. 22(b). For instance, the top-20 and top-100 accuracy for 4-digit numerical passwords is 69% and 89%, respectively, yet it becomes 64% and 83%, respectively, for 8-digit numerical passwords. The accuracy loss is attributed to the increased uncertainty caused by involving more keys. Nevertheless, even for an 8-digit numerical password, the remarkable success rate of 64% after 20 attempts still poses a severe threat to smartphone users.



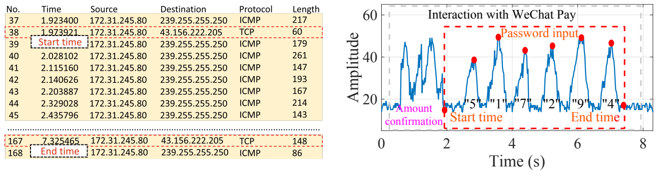
(a) Keystroke classification. (b) Password inference.

Fig. 22. Impact of 3 different password lengths.

5.4 Real-World Experiment

5.4.1 WeChat Pay Password Inference

To showcase the practicality of MuKI-Fi, we conduct a real-world experiment by acting as Eve to steal password



No.	Time	Source	Destination	Protocol	Length
37	1.923400	172.31.245.80	239.255.255.250	ICMP	217
38	1.973921	172.31.245.80	43.156.222.205	TCP	60
39	Start time	172.31.245.80	239.255.255.250	ICMP	179
40	2.028102	172.31.245.80	239.255.255.250	ICMP	261
41	2.115160	172.31.245.80	239.255.255.250	ICMP	147
42	2.140626	172.31.245.80	239.255.255.250	ICMP	193
43	2.203887	172.31.245.80	239.255.255.250	ICMP	167
44	2.329028	172.31.245.80	239.255.255.250	ICMP	214
45	2.435796	172.31.245.80	239.255.255.250	ICMP	143
......					
167	7.325465	172.31.245.80	43.156.222.205	TCP	148
168	End time	172.31.245.80	239.255.255.250	ICMP	86

(a) Attack timing identification. (b) Targeted keystroke extraction.

Fig. 23. Real-world experiment with WeChat Pay.

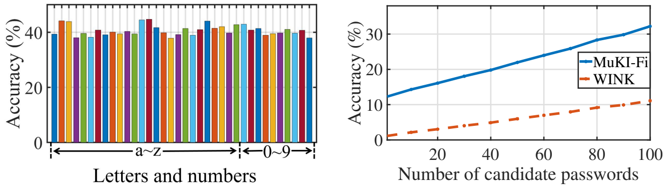
from WeChat Pay, a digital payment service integrated into WeChat [38]. The victim Bob uses an iPhone 13 for his daily activities, typically including WeChat usage, and he is supposed to make a mobile payment transaction with WeChat Pay, for which a numerical password is required, in a conference room of size 5 m × 8 m. The AP is placed on a table and the distance between Bob and the AP ranges from 1.5 to 5 m, as confined by the room layout. Meanwhile, Eve leverages MuKI-Fi to achieve a stealthy eavesdropping at a distance of 3 m from Bob.

Following the method in Sec. 3.1, MuKI-Fi first identifies Bob's Wi-Fi traffic; this is followed by detecting an IP address "43.156.222.205" coinciding with an entry in a pre-recorded IP database, as shown in Fig. 23(a), which in turns starts BFI recording. The recording is stopped once no more requests to that address are made. Subsequently, MuKI-Fi performs SRA on the BFI time series, and the resulting non-sparse BFI series is shown in Fig. 23(b). It appears that the BFI series includes not only the 6-digit numerical password but also other keys entered beforehand (e.g., the transfer amount and confirmation), so we extract the last six peaks corresponding to the password specifically for WeChat Pay, as highlighted by the red box.

After segmenting the signal, MuKI-Fi initiates the password inference. Since WeChat Pay freezes after five incorrect password inputs, we focus on identifying correct passwords among the top 5 candidates. In the experiment shown in Fig. 23(b), the actual password entered by Bob is "517294", and the top 5 candidates are "547294", "517204", "**517294**", "517594", and "517394", indicating a successful password stealing. We conduct 40 such experiments of 8 different people in total, each with a different password. The results indicate that, out of these 40 input passwords, MuKI-Fi achieves a top-5 accuracy of 47.5%, which is quite close to that shown in Fig. 16(a), albeit with a potentially biased statistics given only a small amount of trials. These experiments evidently demonstrate the practicality of MuKI-Fi in real-world scenarios.

5.4.2 Extending to Virtual QWERTY Keyboard

Many applications need more diversified characters than what a numerical keyboard can offer. Typically, banking applications (e.g., the popular Chase Mobile [58]) handling sensitive financial transactions and identity information demand using a virtual (on-screen) QWERTY keyboard for users to create more secure *alphanumeric* passwords. To test the applicability of MuKI-Fi in such scenario, we conduct keystroke classification experiments on the QWERTY keyboard of Chase Mobile. We collect a dataset of 4,000 pre-defined passwords with varying lengths: 1,500 with 6 characters, 1,500 with 8 characters, and 1,000 with



(a) Keystroke classification.

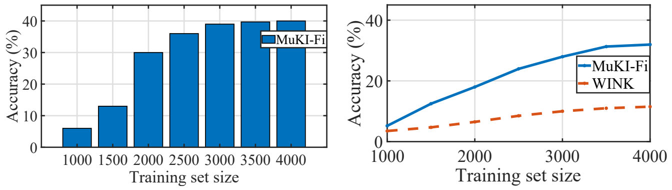
(b) Password inference.

Fig. 24. Performance on QWERTY keyboards.

10 characters. The passwords consist of lowercase letters from ‘a’ to ‘z’ and numbers from ‘0’ to ‘9’. Except for the larger dataset size, we adopt the same experiment settings in Sec. 4. Fig. 24(a) shows that MuKI-Fi achieves an average keystroke classification accuracy of 40%. Additionally, Fig. 24(b) indicates that MuKI-Fi’s top-$[1, 100]$ accuracy of 6-character alphanumeric password ranges from 12% to 32%, surpassing WINK whose top-100 accuracy is only 11%, respectively. Although the accuracy is lower than those in Sec. 5.2.2, it still poses a severe threat to smartphone users. The performance drop on QWERTY keyboards can be attributed to these keyboards having approximately four times more keys than numerical keyboards within the same area. Consequently, the BFI features of clicking different keys are less distinguishable due to their proximity. Additionally, shorter distances (hence shorter transition periods) among keys increase inter-keystroke interference, thereby decreasing KI accuracy.

We also find that KI on a QWERTY keyboard demands a much larger training dataset than on a numerical keyboard. According to Fig. 25, MuKI-Fi performs similarly to the baselines when the training set is small. Fortunately, as the training set size increases from 1,000 to 4,000, MuKI-Fi begins to show its strengths: it improves the keystroke classification accuracy from 6% to 40%, and the top-100 accuracy from 6% to 32%, outperforming the baseline by large margins. The need for a large training set can be explained (again) by the drastic increase in the number of keys on QWERTY keyboards, along with the corresponding increase in the number of domains. By employing the adversarial learning framework, MuKI-Fi can fully utilize the training data and perform adequate KI under domain interference. In contrast, WINK struggles with interference and artifacts caused by a large number of domains, barely improving KI performance.

This experiment also reveals a few challenges to be tackled in future for general KI on QWERTY keyboards. First, more diversified password length should be considered, as over 20% of user may have passwords longer than 10 characters [59]. Second, handling more general passwords containing special characters and uppercase letters is also a crucial aspect: typing these characters may require combinations of multiple keys (e.g., “shift” and its paired keys) and thus complicating the BFI series. Third, certain applications have separate keyboard layouts for distinct groups of keys, requiring users to switch between layouts while entering passwords. Performing KI for such applications requires accurate detection of the layout switching, as well as training two separate neural models for each layout, potentially increasing system complexity. Instead of increasing training data in a brute-force manner, other side-channel attacks and social engineering techniques [60] may be combined with MuKI-Fi to enhance its KI capability in tackling these challenges.



(a) Keystroke classification.

(b) Password inference.

Fig. 25. Extending MuKI-Fi to QWERTY keyboards requires more training data.

6 Related works

We classify existing KI proposals related to MuKI-Fi into the following five different categories:

6.1 Radio-Frequency

WiKey [5] pioneers in leveraging Wi-Fi CSI distortions induced by keystrokes to conduct KI, but the design used by WiKey is soon exceeded (in SNR) by WindTalker [16] for password inference, WINK [17] also leverages simliar design of WiKey but claims that spatiotemporal analysis could enhance the performance of password inference. Recently, WiKI-Eve [61] pioneers the use of Wi-Fi BFI to eavesdrop keystrokes on smartphones without the need for hardware hacking; it serves as the basis of MuKI-Fi, but it targets only attacks on a single Bob.

6.2 Acoustic

Liu et al. [6] propose to classify keys on a keyboard based on the time difference of arrival of the acoustic signals (generated by pressing and releasing a key) at the two microphones on a smartphone. Similarly, KeyListener [62] performs KI on touchscreen based on different attenuation of the signals (generated by phone speaker) at the two microphones. PatternListener [7] compromises pattern locks by using acoustic signals reflected from fingertips to measure their relative movement and infer the pattern lines. These methods can be deemed as acoustic version of method proposed in [17].

6.3 Vision

Early vision-based KI attacks depend on directly observing the contents displayed on a screen [3], [4]. To make it more practical, later works explore side-channel visual cues. KI can be achieved by analyzing changes in the device’s physical appearance, such as shadows and deformations on the screen [9], as well as backside motions of tablet computers [13]. Moreover, capturing videos of the victim’s biometric features during typing, such as finger [8] and eye [14] movements, may also enable KI. Recent work [63] claims to achieve KI even when victims cover the typing hand with the other hand. Although vision-based side-channel attacks have shown a high success rate, the corresponding defense strategies [64], [65] have also grown mature and effective. Compared with the action features required by vision-based KI attacks, MuKI-Fi only requires visual hints (e.g., actions before starting input) rather than the complete input process, as explained in Sec. 3.1.

6.4 Motion Sensors

TouchLogger [15] uses the accelerometer and gyroscope on smartphones to capture phone body movement and infer numerical keys typed on its touchscreen. (sp)iPhone [11] leverages the accelerometer on a nearby phone to detect vibrations from a physical keyboard for enabling KI. Liu et al. [10] further exploit the accelerometer on a smartwatch to capture hand movement and infer keystrokes on POS terminals or QWERTY keyboards.

6.5 Electromagnetic Emission

Vuagnoux et al. [66] propose to eavesdrop on keystrokes from wired and wireless keyboards by capturing electromagnetic emissions during their communications. A later work Periscope [12] extends this idea to a broader range of mobile devices by exploiting human-coupled emission from touchscreens to estimate finger movement trajectories and infer numerical passwords. Vulnerabilities in USB data transfers have also been exploited for password-stealing [67] and malicious command execution [68]. Charger-Surfing [69] further demonstrates that, even without any data transfer over USB, variations of consumed power can be exploited to extract private information such as user passwords.

7 Conclusion

In this paper, we have introduced MuKI-Fi as the pioneering multi-person KI system. By harnessing the new feature BFI, MuKI-Fi accomplishes KI without resorting to low-level hacking. Furthermore, MuKI-Fi's innovative adversarial learning framework empowers KI to generalize effectively across unseen domains, thereby enhancing its practical significance. Additionally, through the utilization of near-field domination effect, MuKI-Fi successfully manages multi-person KI scenarios. Our extensive evaluations validate MuKI-Fi's capability to achieve notably high inference accuracy for both individual keystrokes and numerical passwords. We also explore potential extensions to general keyboards, encompassing both single-person and multi-person scenarios. The outcomes of our research reveal critical vulnerabilities in widely-used applications, such as WeChat, underscoring the urgent need for enhanced security measures to mitigate these risks.

References

[1] R. Cover, *Digital Identities: Creating and Communicating the Online Self*. Academic Press, 2015.

[2] T. W. Bank, "Mobile ID," https://id4d.worldbank.org/guide/mobile-id, 2023, online; accessed 25 March 2023.

[3] F. Maggi, A. Volpatto, S. Gasparini, G. Boracchi, and S. Zanero, "A Fast Eavesdropping Attack against Touchscreens," in *Prof. of the 7th IAS*. IEEE, 2011, pp. 320–325.

[4] Q. Yue, Z. Ling, X. Fu, B. Liu, W. Yu, and W. Zhao, "My Google Glass Sees Your Passwords," *Proceedings of the Black Hat USA*, 2014.

[5] K. Ali, A. X. Liu, W. Wang, and M. Shahzad, "Keystroke Recognition using WiFi Signals," in *Proc. of the 21st ACM MobiCom*, 2015, pp. 90–102.

[6] 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. of the 21st ACM MobiCom*, 2015, pp. 142–154.

[7] M. Zhou, Q. Wang, J. Yang, Q. Li, F. Xiao, Z. Wang, and X. Chen, "PatternListener: Cracking Android Pattern Lock using Acoustic Signals," in *Proc. of the 25th ACM CCS*, 2018, pp. 1775–1787.

[8] D. Shukla, R. Kumar, A. Serwadda, and V. V. Phoha, "Beware, Your Hands Reveal Your Secrets!" in *Proc. of the 21st ACM CCS*, 2014, pp. 904–917.

[9] Q. Yue, Z. Ling, X. Fu, B. Liu, K. Ren, and W. Zhao, "Blind Recognition of Touched Keys on Mobile Devices," in *Proc. of the 21st ACM CCS*, 2014, pp. 1403–1414.

[10] X. Liu, Z. Zhou, W. Diao, Z. Li, and K. Zhang, "When Good Becomes Evil: Keystroke Inference with Smartwatch," in *Proc. of the 22nd ACM CCS*, 2015, pp. 1273–1285.

[11] P. Marquardt, A. Verma, H. Carter, and P. Traynor, "(sp)iPhone: Decoding Vibrations from Nearby Keyboards using Mobile Phone Accelerometers," in *Proc. of the 18th ACM CCS*, 2011, pp. 551–562.

[12] W. Jin, S. Murali, H. Zhu, and M. Li, "Periscope: A Keystroke Inference Attack Using Human Coupled Electromagnetic Emanations," in *Proc. of the 28th ACM CCS*, 2021, pp. 700–714.

[13] J. Sun, X. Jin, Y. Chen, J. Zhang, Y. Zhang, and R. Zhang, "Visible: Video-assisted Keystroke Inference From Tablet Backside Motion," in *Proc. of the IEEE NDSS*, 2016.

[14] Y. Chen, T. Li, R. Zhang, Y. Zhang, and T. Hedgpeth, "EyeTell: Video-assisted Touchscreen Keystroke Inference from Eye Movements," in *Proc. of the 39th IEEE S & P*, 2018, pp. 144–160.

[15] L. Cai and H. Chen, "TouchLogger: Inferring Keystrokes on Touch Screen from Smartphone Motion," in *Proc. of the 6th USENIX Hot Topics*, 2011, pp. 1–9.

[16] M. Li, Y. Meng, J. Liu, H. Zhu, X. Liang, Y. Liu, and N. Ruan, "When CSI Meets Public WiFi: Inferring Your Mobile Phone Password via WiFi Signals," in *Proc. of the 23rd ACM CCS*, 2016, pp. 1068–1079.

[17] E. Yang, Q. He, and S. Fang, "WINK: Wireless Inference of Numerical Keystrokes via Zero-Training Spatiotemporal Analysis," in *Proc. of the 29th ACM CCS*, 2022, pp. 3033–3047.

[18] M. Schulz, D. Wegemer, and M. Hollick. (2017) Nexmon: The C-based Firmware Patching Framework. [Online]. Available: https://nexmon.org

[19] Z. Jiang, T. H. Luan, X. Ren, D. Lv, H. Hao, J. Wang, K. Zhao, W. Xi, Y. Xu, and R. Li, "Eliminating the Barriers: Demystifying wi-fi Baseband Design and Introducing the Picoscenes Wi-Fi sensing Platform," *IEEE Internet of Things Journal*, vol. 9, no. 6, pp. 4476–4496, 2021.

[20] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool Release: Gathering 802.11n Traces with Channel State Information," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 1, p. 53, 2011.

[21] I. Corporation, "Intel Ultimate N WiFi Link 5300," https://www.intel.com/content/dam/www/public/us/en/documents/product-briefs/ultimate-n-wifi-link-5300-brief.pdf, 2008, online; accessed 28 March 2023.

[22] S. Gollakota, F. Adib, D. Katabi, and S. Seshan, "Clearing the Rf Smog: Making 802.11 N Robust to Cross-Technology Interference," in *Proc. of the 25th ACM SIGCOMM*, 2011, pp. 170–181.

[23] J. Huang, G. Xing, G. Zhou, and R. Zhou, "Beyond Co-existence: Exploiting WiFi White Space for Zigbee Performance Assurance," in *Proc. of the 18th IEEE ICNP*, 2010, pp. 305–314.

[24] K. Kosek-Szott, J. Gozdecki, K. Loziak, M. Natkaniec, L. Prasnal, S. Szott, and M. Wagrowski, "Coexistence Issues in Future WiFi Networks," *IEEE Network*, vol. 31, no. 4, pp. 86–95, 2017.

[25] K. Qian, C. Wu, Y. Zhang, G. Zhang, Z. Yang, and Y. Liu, "Widar2.0: Passive Human Tracking with a Single Wi-Fi Link," in *Proc. of the 16th ACM MobiSys*, 2018, p. 350–361.

[26] Y. Zeng, D. Wu, J. Xiong, J. Liu, Z. Liu, and D. Zhang, "Multi-Sense: Enabling Multi-Person Respiration Sensing with Commodity WiFi," in *Proc. of the 22nd UbiComp*, 2020, pp. 102:1–29.

[27] S. Zhang, T. Zheng, Z. Chen, and J. Luo, "Can We Obtain Fine-grained Heartbeat Waveform via Contact-free RF-sensing?" in *Proc. of the 41st IEEE INFOCOM*, 2022, pp. 1759–1768.

[28] M. S. Gast, *802.11ac A Survival Guide: Wi-Fi at Gigabit and Beyond*. O'Reilly Media, Inc., 2013.

[29] "IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems–Local and Metropolitan Area Networks–Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 10: Mesh Networking," *IEEE P802.11s/D8.0, December 2010*, pp. 1–350, 2010.

[30] J. Bullock and J. T. Parker, *Wireshark for Security Professionals: Using Wireshark and the Metasploit Framework*. John Wiley & Sons, 2017.

[31] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in *Proc. of NeurIPS*, 2014, pp. 2672–2680.

[32] A. Goldsmith, *Wireless Communications*. Cambridge, U.K.: Cambridge University Press, 2005.

[33] T. S. Rappaport, *Wireless Communications: Principles and practice*. Pearson Education India, 2010.

[34] R. Beyah and A. Venkataraman, “Rogue-access-point Detection: Challenges, Solutions, and Future Directions,” *IEEE Security & Privacy*, vol. 9, no. 5, pp. 56–61, 2011.

[35] C. Wang, X. Zheng, Y. Chen, and J. Yang, “Locating Rogue Access Point using Fine-grained Channel Information,” *IEEE Transactions on Mobile Computing*, vol. 16, no. 9, pp. 2560–2573, 2016.

[36] I. Cisco Systems, “Cisco Wireless Controller Configuration Guide, Release 8.4,” https://www.cisco.com/c/en/us/td/docs/wireless/controller/8-4/configguide/b_cg84/wireless_intrusion_detection_system.html#rogue-ap-classification, 2023, online; accessed 25 March 2023.

[37] S. Kiranyaz, T. Ince, O. Abdeljaber, O. Avci, and M. Gabbouj, “1-D Convolutional Neural Networks for Signal Processing Applications,” in *Proc. of IEEE ICASSP*, 2019, pp. 8360–8364.

[38] WeChat, “WeChat - Free Messaging and Chatting App,” https://www.wechat.com/, 2023, online; accessed 28 March 2023.

[39] S. Wu, Y. Zhang, X. Wang, X. Xiong, and L. Du, “Forensic Analysis of WeChat on Android Smartphones,” *Digital Investigation*, vol. 21, pp. 3–10, 2017.

[40] G. W. Stewart, “On the Early History of the Singular Value Decomposition,” *SIAM Review*, vol. 35, no. 4, pp. 551–566, 1993.

[41] A. Orebaugh, G. Ramirez, and J. Beale, *Wireshark & Ethereal Network Protocol Analyzer Toolkit*. Elsevier, 2006.

[42] R. Nitzberg, “Constant-false-alarm-rate Signal Processors for Several Types of Interference,” *IEEE Transactions on Aerospace and Electronic Systems*, no. 1, pp. 27–34, 1972.

[43] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[44] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, “A Theory of Learning from Different Domains,” *Machine Learning*, vol. 79, pp. 151–175, 2010.

[45] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, “Domain Adaptation via Transfer Component Analysis,” *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, 2010.

[46] Y. Ganin and V. Lempitsky, “Unsupervised Domain Adaptation by Backpropagation,” in *Proc. of the 32nd ICML*, 2015, pp. 1180–1189.

[47] X. Wang, K. Niu, J. Xiong, B. Qian, Z. Yao, T. Lou, and D. Zhang, “Placement Matters: Understanding the Effects of Device Placement for WiFi Sensing,” *Proc. of the ACM IMWUT*, vol. 6, no. 1, pp. 32:1–25, 2022.

[48] A. Inc., “Acer TravelMate Laptops for Business,” https://www.acer.com/sg-en/laptops/travelmate, 2023, online; accessed 25 March 2023.

[49] I. Corporation, “Intel® Wi-Fi 6 AX201,” https://www.intel.sg/content/www/xa/en/products/sku/130293/intel-wifi-6-ax201-gig/specifications.html, 2023, online; accessed 25 March 2023.

[50] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” *arXiv preprint arXiv:1912.01703*, 2019.

[51] Apple Inc., “Buy iPhone 13,” https://www.apple.com/sg/shop/buy-iphone/iphone-13, 2023, online; accessed 12 February 2023.

[52] OnePlus, “OnePlus 10T 5G,” https://www.oneplus.com/sg/10t, 2023, online; accessed 10 April 2023.

[53] Xiaomi, “Xiaomi 13 Pro,” https://www.mi.com/sg/product/xiaomi-13-pro/, 2023, online; accessed 10 April 2023.

[54] Huawei Device Co., Ltd., “HUAWEI P40 Pro,” https://consumer.huawei.com/en/phones/p40-pro/, 2023, online; accessed 10 April 2023.

[55] Samsung, “Samsung Galaxy S20 Series,” https://www.samsung.com/sg/news/local/galaxy-s20-launch/, 2023, online; accessed 10 April 2023.

[56] L. v. d. Maaten and G. Hinton, “Visualizing Data Using t-SNE,” *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[57] T.-T. Wong, “Performance Evaluation of Classification Algorithms by k-fold and Leave-one-out Cross Validation,” *Pattern Recognition*, vol. 48, no. 9, pp. 2839–2846, 2015.

[58] J. C. . Co., “Mobile Banking Features with Chase Mobile App,” https://www.chase.com/digital/mobile-banking, 2023, online; accessed 25 March 2023.

[59] statista, “Average Number of Characters for a Password in the United States in 2021,” https://www.statista.com/statistics/1305713/average-character-length-of-a-password-us/, 2023, online; accessed 25 March 2023.

[60] C. Hadnagy, *Social Engineering: The Art of Human Hacking*. John Wiley & Sons, 2010.

[61] J. Hu, H. Wang, T. Zheng, J. Hu, Z. Chen, H. Jiang, and J. Luo, “Password-Stealing without Hacking: Wi-Fi Enabled Practical Keystroke Eavesdropping,” 2023, to appear.

[62] L. Lu, J. Yu, Y. Chen, Y. Zhu, X. Xu, G. Xue, and M. Li, “Keylistener: Inferring keystrokes on qwerty keyboard of touch screen through acoustic signals,” in *Proc. of the 38th IEEE INFOCOM*, 2019, pp. 775–783.

[63] M. Cardaioli, S. Cecconello, M. Conti, S. Milani, S. Picek, and E. Saraci, “Hand Me Your PIN! Inferring ATM PINs of Users Typing with a Covered Hand,” in *Proc. of the 31st USENIX Security*, 2022, pp. 1687–1704.

[64] Z. Liu, F. Lin, C. Wang, Y. Shen, Z. Ba, L. Lu, W. Xu, and K. Ren, “CamRadar: Hidden Camera Detection Leveraging Amplitude-modulated Sensor Images Embedded in Electromagnetic Emanations,” *Proc. of the 23rd ACM UbiComp*, vol. 6, no. 4, pp. 1–25, 2023.

[65] S. Sami, S. R. X. Tan, B. Sun, and J. Han, “LAPD: Hidden Spy Camera Detection Using Smartphone Time-of-flight Sensors,” in *Proc. of the 19th ACM SenSys*, 2021, pp. 288–301.

[66] M. Vuagnoux and S. Pasini, “Compromising Electromagnetic Emanations of Wired and Wireless Keyboards,” in *Proc. of the 18th USENIX Security Symposium*, vol. 8, 2009, pp. 1–16.

[67] J. V. Monaco, “SoK: Keylogging Side Channels,” in *Proc. of the 39th IEEE S&P*, 2018, pp. 211–228.

[68] D. J. Tian, G. Hernandez, J. I. Choi, V. Frost, C. Raules, P. Traynor, H. Vijayakumar, L. Harrison, A. Rahmati, M. Grace *et al.*, “Attention spanned: Comprehensive vulnerability analysis of at commands within the android ecosystem,” in *Proc. of the 27th USENIX Security*, 2018, pp. 273–290.

[69] P. Cronin, X. Gao, C. Yang, and H. Wang, “Charger-surfing: Exploiting a power line side-channel for smartphone information leakage,” in *Proc. of the 30th USENIX Security*, 2021, pp. 681–698.

PLACE
PHOTO
HERE

Hongbo Wang is a currently pursuiting Ph.D. student with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. He received the MS degree in Comunications Engineering from Nanyang Technological University in 2021 and the BS degree in Electrical Engineering from University of Electronic Science and Technology of China in 2020. He has published papers in ACM Sensys, ACM MobiCom, ACM CCS, etc. His research interests include Integrated Sensing and Communication (ISAC) and deep learning.

PLACE
PHOTO
HERE

Jingyang Hu is currently pursuiting Ph.D. student with the College of Computer Science and Electronic Engineering, Hunan University, China. From 2022 to 2023, he works as a joint Ph.D. student at the School of Computer Science and Engineering at Nanyang Technological University (NTU), Singapore. He has published papers in ACM Ubicomp, ACM CCS, IEEE ICDCS, IEEE TMC, IEEE JSAC, IEEE IoT-J, etc. His research interests include wireless sensing and deep learning.

PLACE
PHOTO
HERE

Tianyue Zheng (zhengty@sustech.edu.cn) is currently an assistant professor at the School of Computer Science and Engineering, Southern University of Science and Technology, China. He received his Ph.D. degree from Nanyang Technological University, Singapore, M.Eng. degree from the University of Toronto, Canada, and B.Eng. degree from Harbin Institute of 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.

PLACE
PHOTO
HERE

Jingzhi Hu (Member, IEEE) received the B.S. degree at the School of Electrical Engineering and Computer Science and the Ph.D. degree at the School of Electronics at Peking University, in 2017 and 2022, respectively. He is currently a research fellow at School of Computer Science and Engineering, Nanyang Technological University. His main research interests are machine learning, Wi-Fi sensing systems, and reconfigurable intelligent surface-aided RF sensing techniques for the Internet of Things, and has published papers in prestigious venues such as IEEE Journal on Selected Areas in Communications, Transactions on Wireless Communications, Wireless Communications, and ACM MobiCom. He served as a TPC Member for IEEE/CIC ICCC in 2017 and 2018.

PLACE
PHOTO
HERE

Zhe Chen is an associate professor within the School of Computer Science at Fudan University, and the Co-Founder of AIWiSe Ltd. Inc. He obtained his Ph.D. degree in Computer Science from Fudan University, China, with a 2019 ACM SIGCOMM China Doctoral Dissertation Award. Before joining Fudan University, he worked as a research fellow in NTU for several years, and his research achievements, along with his efforts in launching products based on them, have thus earned him 2021 ACM SIGMOBILE China Rising Star Award recently.

PLACE
PHOTO
HERE

Hongbo Jiang is now a full professor in the College of Computer Science and Electronic Engineering, Hunan University. He was a professor at Huazhong University of Science and Technology. He received a Ph.D. from Case Western Reserve University in 2008. He has been serving on the editorial board of IEEE/ACM ToN, IEEE TMC, ACM ToSN, IEEE TNSE, IEEE TITS, IEEE IoT-J, etc. He was also invited to serve on the TPC of IEEE INFOCOM, ACM WWW, ACM/IEEE MobiHoc, IEEE ICDCS, IEEE ICNP, etc. He is an elected Fellow of IET (The Institution of Engineering and Technology), Fellow of BCS (The British Computer Society), Senior Member of ACM, Senior Member of IEEE, and Full Member of IFIP TC6 WG6.2. Now his research focuses on computer networking, especially, wireless networks, data science in Internet of Things, and mobile computing.

PLACE
PHOTO
HERE

Yuanjin Zheng received the B.Eng. and M.Eng. degrees from Xi'an Jiaotong University, Xi'an, China, in 1993 and 1996, respectively, and the Ph.D. degree from Nanyang Technological University, Singapore, in 2001. From 1996 to 1998, he was with the National Key Laboratory of Optical Communication Technology, University of Electronic Science and Technology of China. In 2001, he joined the Institute of Microelectronics (IME), Agency for Science, Technology and Research (A$*$STAR), and had been a Principle Investigator and Group Leader. With the IME, he has led and developed various projects like CMOS RF transceivers, baseband system-on-a-chip (SoC) for wireless systems, ultra-wideband, and lowpower biomedical ICs. In 2009, he joined Nanyang Technological University, as an Assistant Professor and Program Director. He has authored or coauthored over 300 international journal and conference papers, 26 patents filed/granted, and several book chapters. His research interests are gigahertz RFIC and SoC design, biosensors and imaging, and SAW/BAW/MEMS sensors.

PLACE
PHOTO
HERE

Jun Luo received his BS and MS degrees in Electrical Engineering from Tsinghua University, China, and the Ph.D. 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 in 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.