SmartMagnet: Proximity-Based Access Control for IoT Devices with Smartphones and Magnets

Yan Zhang, Dianqi Han, Ang Li, Jiawei Li, *Student Member, IEEE,* Tao Li, *Member, IEEE,* and Yanchao Zhang, *Fellow, IEEE*

Abstract— Ubiquitous smartphones can be powerful tools to access IoT devices. Proximity-based access control (PBAC) is needed such that IoT devices only allow data access by legitimate users in close proximity. Traditional smartphone-based authentication techniques do not satisfy the PBAC requirements. This paper presents SmartMagnet, a novel scheme that combines smartphones and cheap magnets to achieve PBAC for IoT devices. SmartMagnet explores a few cheap, tiny commodity magnets which we propose to attach to or embed into IoT devices, as well as the magnetometer and attitude sensor on commodity smartphones. Each legitimate user performs a self-chosen 3D password gesture near the target IoT device with the enrolled smartphone. Then the system server uses the IoT device's confidential magnet configuration parameters to reconstruct the user gesture from the magnetometer and attitude sensor data submitted by the smartphone. If the reconstructed gesture matches the stored template of the purported user, the smartphone user is deemed legitimate and allowed access to the IoT device. Extensive experiments confirm the high usability of SmartMagnet and its strong resilience to lost/stolen smartphones and also remote attacks via signal relaying.

Index Terms—IoT, smartphone, authentication, gesture recognition, magnet, security.

1 Introduction

The Internet of Things (IoT) is quickly reshaping the way people gain digital insights about and interact with the physical world. Cisco predicts that there will be 500 billion IoT devices worldwide by 2030. Many IoT applications may involve various users accessing and quickly acting upon the data at nearby IoT devices. Given strong security concerns, *proximity-based access control* (PBAC) is needed such that IoT devices only allow data access by legitimate users in close proximity [1]–[3]. Due to application contexts, form-factor constraints, and cost considerations, many IoT devices such as those in smart cities and industrial environments may lack a user interface (UI) for inputting a user password. It remains an open challenge to design a secure and usable PBAC scheme for no-UI IoT devices.

Ubiquitous smartphones can be powerful tools to access IoT devices with or without a UI. In particular, a legitimate user, say, Bob, can enroll his smartphone with the IoT system to access nearby IoT devices through WiFi/BLE/NFC. There are three conventional authentication methods for mutual authentication between Bob's smartphone $\mathcal{S}$ and the IoT device (say, $\mathcal{D}$) he aims to access. The first method relies on a pre-distributed shared key between $\mathcal{S}$ and $\mathcal{D}$, but such shared keys are knowingly very difficult to manage in a large IoT system with many devices and legitimate users. The second method explores public-key authentication by assuming that $\mathcal{S}$ and $\mathcal{D}$ both have a certified pair of unique public and private keys, but the involved overhead for computations and public-key certificate revocations may be too demanding for resource-constrained IoT devices. The third method uses the IoT system server as the authentication proxy which maintains a secure connection with $\mathcal{S}$. Each time Bob wishes to access $\mathcal{D}$, he uses $\mathcal{S}$ to request authorization from the server. After authenticating $\mathcal{S}$, the server distributes a session key to $\mathcal{S}$ and $\mathcal{D}$ for securing subsequent communications between them. Note that the third method or its variant has been widely used in practice, e.g., to unlock Internet-connected smart locks [4].

Smartphone-based authentication methods above do not satisfy the PBAC requirements. First, smartphones can be lost/stolen, and many are protected by weak passwords or even not password-protected. For instance, a Kensington reports estimates that more than 70 million smartphones are lost each year with only 7% recovered [5]; Kaspersky Lab reports that 52% of people do not password-protect their mobile devices [6]. Continue with the previous example. If $\mathcal{S}$ is lost/stolen, an adversary may unlock it to impersonate Bob to access $\mathcal{D}$ before Bob informs the server. The root reason is that the authentication of a smartphone is falsely considered equivalent to that of the corresponding user. In reality, smartphones are typically used only as secondary authentication factors, e.g., in a two-factor mobile authentication system. Second, smartphones are vulnerable to the *remote attack* where an adversary stealthily relays wireless signals between smartphones and IoT devices far apart. Therefore, an adversary possessing $\mathcal{S}$ or even Bob himself (e.g., an insider attacker) can attempt to interact with $\mathcal{D}$ from a remote location, e.g., to issue destructive commands while avoid being caught on the spot. The root cause for this attack is lack of proximity verification between $\mathcal{S}$ and $\mathcal{D}$. One may think about secure localization techniques to determine precise smartphone locations, but such techniques often require densely deployed access points which may incur a prohibitive cost in a large-scale IoT system [7].

This paper presents **SmartMagnet**, a novel scheme that

Yan Zhang, D. Han, A. Li, J. Li, and Y. Zhang are with the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ 85287, USA. E-mail: {yanzhangyz, dqhan, anglee, yczhang}@asu.edu.
T. Li is with the Computer and Information Technology Department, Indiana University-Purdue University Indianapolis, Indianapolis, IN 46202, USA. E-mail: tli6@iupui.edu.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3149746, IEEE Transactions on Mobile Computing

combines smartphones and cheap magnets to achieve PBAC for IoT devices. SmartMagnet is motivated by the reed switch widely used in door/window sensors in modern alarm systems, which is paired with a permanent magnet to detect door/window opening and closing according to magnetic field changes. It explores one or a few cheap, tiny commodity magnets which we propose to attach to or embed into current and future IoT devices, as well as the magnetometer and attitude sensor on commodity smartphones. The magnet configuration—including the magnet type, the number of magnets, and their layout—is relatively unique to each IoT device and results in a magnetic field difficult to guess and emulate. To pass authentication, a legitimate user performs a self-chosen 3D password gesture near the target IoT device with the enrolled smartphone. Then the system server uses the IoT device's magnet configuration to reconstruct the user gesture from the magnetometer and attitude sensor data submitted by the smartphone. If the reconstructed gesture matches the stored template of the purported user, the smartphone user is deemed legitimate and allowed access to the IoT device.

SmartMagnet offers strong resilience to lost/stolen smartphones and also remote attacks. In particular, each password gesture is designed to be easily reproduced by the legitimate user but difficult for the attacker to guess or emulate. So even if able to unlock a lost/stolen legitimate smartphone, the adversary can hardly perform the correct password gesture. In addition, if the correct password gesture is performed with the legitimate smartphone at a faraway location from the IoT device, the resulting magnetometer and attitude sensor data only match the surrounding magnetic field and are not compatible with that of the IoT device. Therefore, the system server would not be able to recover the right password gesture with the IoT device's magnet configuration. SmartMagnet thus defeats remote attacks launched by the adversary (not) knowing the password gesture and achieves true PBAC.

The major challenges to implement SmartMagnet include recovering the password gesture from the noisy magnetometer and attitude sensor data induced by the user's natural hand movement and then validating its authenticity. We tackle these challenges in three steps. First, we apply novel transformations to translate the smartphone's moving trajectory into a fixed coordinate system centered. Second, we recognize the password gesture with the help of the MyScript framework [8]. Finally, we validate the legitimacy of the recovered password gesture with both Dynamic Time Warping (DTW), traditional machine learning methods (Naive Bayes and Random Forest), and a three-layer Convolutional Neural Network (CNN).

We conduct comprehensive experiments with commodity magnets and iPhone 6s to verify the security and usability of SmartMagnet. Our evaluations involve 12 volunteers and over 720 samples. We show that SmartMagnet is highly secure with the true-positive rates (TPRs) up to 91.5% and 96.3% and the false-positive rates (FPRs) no larger than 4.8% and 3.5% for the DTW and CNN methods, respectively. We also show that a brute-forth remote attacker needs at least 756,680 trials to guess and emulate the correct magnetic field of the targeted IoT device equipped with only two magnets. Since a practical PBAC system often rate-limits unsuccessful attempts, SmartMagnet can effectively thwart signal-relaying attacks. In addition, the enrollment and authentication time of SmartMagnet with both the DTW and CNN methods are comparable to those of finger or face authentication. Finally, a usability study based on the evaluation framework in [9], [10] confirms that SmartMagnet is quite easy and convenient to use.

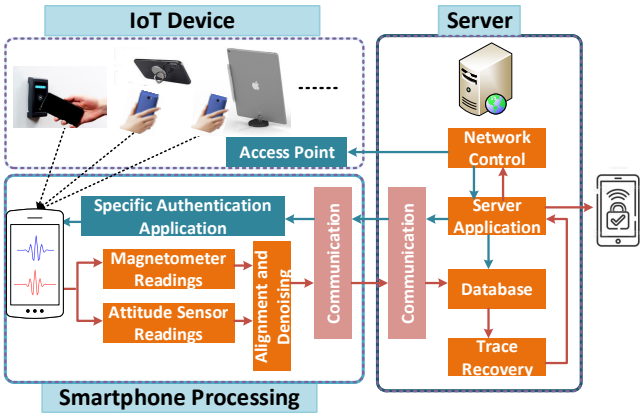


Fig. 1: SmartMagnet system architecture.

The rest of this paper is organized as follows. Section 2 gives the system and adversary models. Section 3 presents our approach to reconstruct the user gesture from the noisy magnetometer and attitude data. Section 4 illustrates the SmartMagnet design. Section 5 presents the experimental evaluation of SmartMagnet. Section 6 reviews the related work. Section 7 concludes this paper.

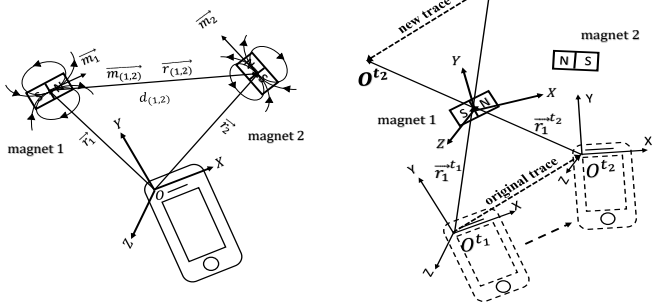
2 System and Adversary Models

System Model. As shown in Fig. 1, SmartMagnet comprises a system server, IoT devices, and smartphones registered by legitimate users. There are $n \geq 1$ cheap, tiny commodity permanent magnets (less than U.S. $1.5 in our experiments) in each IoT device. As shown in Section 5, $n = 2$ can guarantee sufficient security. The magnet configuration—including the magnet type, n, and their layout—is relatively unique to each IoT device and defined by some parameters (Section 3.2.3) stored on the server. Each IoT device communicates with the server through a secure channel.

We use Bob as an exemplary legitimate user to outline the system operations and design requirements. Bob registers his smartphone $\mathcal{S}$ and installs a system app which maintains a secure connection with the server. Bob protects $\mathcal{S}$ with a usual password and also enrolls a self-chosen 3D *password gesture* (Section 4.1) with the server. Assume that Bob wants to interact with a nearby IoT device $\mathcal{D}$ through a WiFi/BLE/NFC channel. Bob passes the PBAC if he is verified to use $\mathcal{S}$ to perform the enrolled password gesture around $\mathcal{D}$. SmartMagnet achieves this in four steps.

1) Bob waves $\mathcal{S}$ to perform his password gesture around $\mathcal{D}$. The system app on $\mathcal{S}$ records and submits the resulting magnetometer and attitude sensor data to the server as a PBAC request to access $\mathcal{D}$.
2) The server retrieves $\mathcal{D}$'s magnet configuration from its database, with which to extract a gesture trace from received sensor data.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3149746, IEEE Transactions on Mobile Computing



(a) Magnetic field illustration (b) Trace reconstruction

Fig. 2: Illustrations of two-magnet magnetic field and reconstructed trace in the smartphone coordinate system.

3) The server inputs the gesture trace into a coarse-grained *gesture recognition* module and then a fine-grained *user identification* module. The former module aims to recognize Bob's password gesture, and the latter seeks to verify that Bob indeed performs the gesture. Bob passes the PBAC if both modules succeed.
4) The server distributes a session key separately to $\mathcal{S}$ and $\mathcal{D}$ for them to establish a secure WiFi/BLE/NFC connection for subsequent communications.

Adversary Model. We assume a powerful attacker who possesses a legitimate smartphone whereby to cheat a target IoT device. Although the smartphone can be password-protected, it remains unlocked either because the password is very weak to guess or because it is lost/stolen while in the unlocked state. This means that the attacker can submit data through the system app on the legitimate smartphone to the server. The attacker is fully aware of how SmartMagnet works and can launch both an *in-situ* attack near the IoT device and a remote attack from a faraway location. For both in-situ and remote attacks, we assume the worst-case scenario that the attacker may have observed the legitimate user's password gesture either as a bystander or through a spy camera. In addition, the remote attacker may also be the legitimate user him/herself, e.g., as a malicious insider to fake presence around the target IoT device in a workplace.

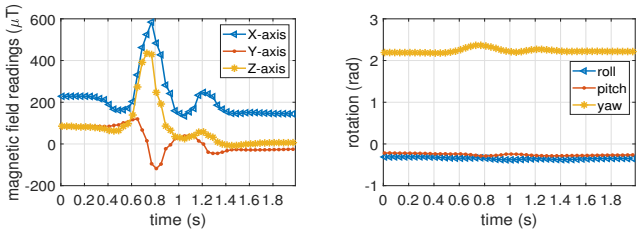
3 Reconstructing Password Gesture

SmartMagnet requires each legitimate user to perform his/her self-chosen password gesture by waving his/her smartphone around the IoT device with embedded or attached magnets. The smartphone then submits the resulting magnetometer and attitude sensor data to the system server which then reconstructs the performed gesture. In this section, we outline our technique to achieve this purpose.

3.1 Magnetic Field Illustration

A magnet induces a 3D magnetic field vector (MFV) [11] measured by a nearby magnetometer, which comprises the magnetic field strength in three axes and is denoted by

$$\vec{H}(\vec{r},\vec{m}) = \frac{\lambda}{|\vec{r}|^3}\left[\frac{3\vec{r}(\vec{m}^\top\vec{r})}{|\vec{r}|^2} - \vec{m}\right] \quad (1)$$



(a) Magnetic field readings (b) Attitude readings

Fig. 3: Magnetic field and attitude sensor readings when one user performs letter "a" by waving his smartphone.

where $\vec{H}(\vec{r},\vec{m}) = (H_x, H_y, H_z)$, λ is a constant relating to the magnetic moment, $\vec{r} = (r_x, r_y, r_z)$ represents the magnet's displacement vector relative to the magnetometer, and $\vec{m} = (m_x, m_y, m_z)$ represents the unit orientation vector pointing from the magnet's south pole to its north pole. All the vectors in Eq. (1) are measured in the magnetometer (smartphone) coordinate system shown in Fig. 3a. The MFVs induced by multiple magnets are simply the addition of individual MFVs. Fig. 2a gives an example with two magnets. The overall x-axis MFV can be derived as

$$H_x = \frac{\lambda_1\left[-3r_{1,x}(-m_{1,x}r_{1,x}-m_{1,y}r_{1,y}-m_{1,z}r_{1,z})-m_{1,x}(r_{1,x}^2+r_{1,y}^2+r_{1,z}^2)\right]}{(r_{1,x}^2+r_{1,y}^2+r_{1,z}^2)^{5/2}} + \frac{\lambda_2\left[-3r_{2,x}(-m_{2,x}r_{2,x}-m_{2,y}r_{2,y}-m_{2,z}r_{2,z})-m_{2,x}(r_{2,x}^2+r_{2,y}^2+r_{2,z}^2)\right]}{(r_{2,x}^2+r_{2,y}^2+r_{2,z}^2)^{5/2}}. \quad (2)$$

The expressions of H_y and H_z are similar and omitted here due to the space limit.

3.2 Password Gesture Reconstruction

The password gesture is equivalent to the trajectory of the smartphone magnetometer and can be derived using $\vec{r}$ in Eq. (1). We tackle this challenge in two steps.

3.2.1 Step 1: deriving unit orientation vector $\vec{m}$

The unit orientation vector $\vec{m}$ in Eq. (1) is measured in the smartphone coordinate system which dynamically changes with the smartphone movement. In contrast, the magnet orientations are relatively fixed with regard to the IoT device itself. So we need proper coordinate transformations to map the magnet orientations to the varying smartphone coordinate system to obtain $\vec{m}$. We use the attitude sensor which has become a standard IMU sensor in mobile and IoT devices to achieve this goal. The attitude sensor uses the earth coordinate system, while the smartphone uses a 3-axis coordinate system defined relative to the phone screen when the smartphone is held in its default orientation. We assume that the IoT device uses its local coordinate system defined in the same way as the smartphone coordinate system.

Consider magnet 1 in Fig. 2a as an example. Let $\vec{m}_{1,0}$ denote a unit orientation vector pointing to the north magnet pole in the IoT device's coordinate system. Assume that the attitude sensor in the IoT device outputs the yaw, roll, and pitch angles denoted by α, β, and γ, respectively. The transformation matrix from the IoT device's coordinate

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3149746, IEEE Transactions on Mobile Computing

system to the earth coordinate system can be defined as R_1^T where R_1 is expressed as

$$R_1 = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix}. \quad (3)$$

With the smartphone's attitude sensor, we can define a similar matrix R_2 from the earth coordinate system to the smartphone coordinate system. Finally, we can derive the unit orientation vector for magnet 1 in the smartphone coordinate system as $\vec{m_1} = R_2 R_1^T \vec{m}_{1,0}$ and for other magnets similarly, where R_1^T denotes the transposition of R_1.

3.2.2 Step 2: deriving smartphone-displacement vector $\vec{r}^*$

We proceed to derive the magnet's displacement vector $\vec{r} = (r_x, r_y, r_z)$ in the smartphone coordinate system. If there is only one magnet, we can use $\vec{m}$ derived in Step 1 and the magnetometer output $\vec{H}(\vec{r}, \vec{m})$ to obtain a unique closed-form solution $\vec{r}$ to Eq. (1). If there are two magnets such as in Fig. 2a, we obtain two unit orientation vectors $\vec{m_1}$ and $\vec{m_2}$ in Step 1. In this case, Eq. (1) has two unknown vectors $\vec{r_1} = (r_{1,x}, r_{1,y}, r_{1,z})$ and $\vec{r_2} = (r_{2,x}, r_{2,y}, r_{2,z})$, and there are no closed-form solutions. But we can rewrite $\vec{r_2}$ as

$$\vec{r_2} = (r_{1,x}, r_{1,y}, r_{1,z}) + d_{(1,2)} * (m_{(1,2),x}, m_{(1,2),y}, m_{(1,2),z}),$$

where $\vec{m}_{(1,2)} = (m_{(1,2),x}, m_{(1,2),y}, m_{(1,2),z})$ denotes the unit orientation vector from magnet 1 to magnet 2, and $d_{(1,2)}$ is the distance between the magnet centers. Both $\vec{m}'_{1,2}$ (similar to $\vec{m}_{1,0}$ in Eq. (3)) and $d_{(1,2)}$ can be obtained during manufacturing and preloaded to the system server. Then we have $\vec{m}_{(1,2)} = R_2 R_1^T \vec{m}'_{1,2}$. This process can be similarly performed for three or more magnets. Given the knowledge of orientation vectors, the equation set, consisting of Eq. (2) and the corresponding expressions of H_y and H_z, could be solved using the classical Levenberg-Marquardt (LM) algorithm–a popular numerical method for nonlinear least-squares problems–with only three variables $[r_{1,x}, r_{1,y}, r_{1,z}]$.

Finally, we define a magnet coordinate system centered at an arbitrary magnet (e.g., magnet 1 in Fig. 2a) with axes parallel to the earth coordinate system. Then we use its displacement vector $\vec{r_1}$ and the transformation matrix R_2 used in Step 1 to obtain the magnetometer displacement $\vec{r}_1^*$ (i.e., the 3D gesture trajectory) in the magnet coordinate system. In particular, we compute

$$[r^*_{1,x}, r^*_{1,y}, r^*_{1,z}]^T = R_2^T [r_{1,x}, r_{1,y}, r_{1,z}]^T. \quad (4)$$

Fig. 2b gives an example of trace reconstruction. When the smartphone moves from one point O^{t_1} at t_1 to another point O^{t_2} at t_2, we can derive two displacement vectors for magnet 1 as $\vec{r_1}^{t_1}$ and $\vec{r_1}^{t_2}$ in the smartphone coordinate system, which are then mapped into the magnet coordinate system based on the aforementioned process. As we can see, the point trajectory $O^{t_1} \dashrightarrow O^{t_2}$ changes its direction to the opposite. However, this change does not affect the final trace recovery because the new and original trajectories are symmetric about the origin of the magnet coordinate system. The later gesture recognition and user identification phases in SmartMagnet only depend on the shape of the gesture trace without using its direction.

3.2.3 Magnet configuration parameters

The system server needs some critical parameters to reconstruct a gesture trace according to the process above. These include the number n of magnets, the magnetic constant of each magnet (e.g., λ_1 in Eq. (2)), the individual orientation vector of each magnet like $\vec{m}_{1,0}$, the distance (e.g. $d_{(1,2)}$) between the origin magnet and another magnet, and the unit orientation vector (e.g., $\vec{m}'_{1,2}$) from the origin magnet to another magnet. These parameters define the relatively unique magnetic environment of each IoT device and are kept confidential to the system server.

4 SmartMagnet Design

In this section, we illustrate the SmartMagnet framework which consists of an enrollment phase and a verification phase, as shown in Fig. 4.

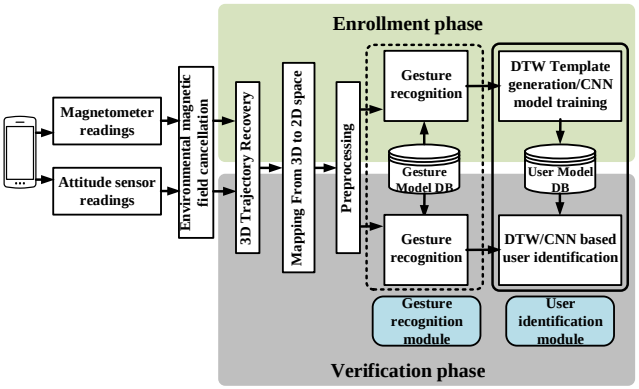


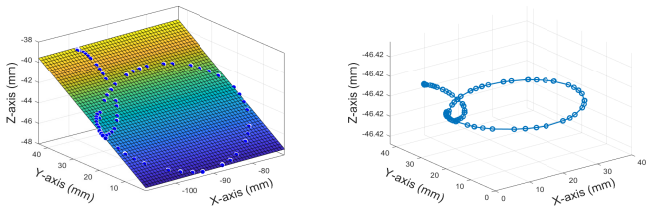
Fig. 4: SmartMagnet Workflow.

4.1 Enrollment Phase

In the enrollment phase, each legitimate user needs to select a proper password gesture easy for him/herself to remember and reproduce (high usability) but difficult for the attacker to emulate (high security). The authors in [12] first discretize a 2D gesture, then use the similar representation of PIN numbers with the n-gram Markov Model, and finally refer to the partial guessing entropy estimation [13]. If the discretized gesture is complex, the original gesture is also considered complex. SmartMagnet could follow a white list of good gestures in [12] which achieve the usability and security at the same time properly.

Then the user opens the SmartMagnet app and starts to perform the chosen gesture as what s/he would do in subsequent verification phases. It is worth noting that in a large system of IoT devices connected to the same server, the user just needs to enroll with an arbitrary IoT device but can use the same gesture with any other. This nice feature also translates into the high usability of SmartMagnet. The SmartMagnet app records and then submits the resulting magnetometer and attitude sensor data to the system server which reconstructs the password gesture (or smartphone trace) with the techniques in Section 3. Next, the recovered 3D trace goes through the following steps in sequence.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3149746, IEEE Transactions on Mobile Computing



(a) Before rotation and translation (b) After rotation and translation

Fig. 5: Mapping from 3D to 2D space.

4.1.1 Cancelling environmental magnetic field

Before trajectory recovery, we first cancel the impact of environmental magnetic field (EMF) [14]. We denote the EMF recordings as $H_{mag'} = (H_{mag',x}, H_{mag',y}, H_{mag',z})$ in the earth coordinate system. Then the real-time EMF values recorded by the magnetometer in the smartphone can be obtained using the following equation,

$$[H_{mag,x}, H_{mag,y}, H_{mag,z}]^T = R_2[H_{mag',x}, H_{mag',y}, H_{mag',z}]^T. \quad (5)$$

Here R_2 is the rotation matrix defined in Section 3.2.1. After receiving the readings from the magnetometer and attitude sensor, the system server simply subtracts the EMF values.

4.1.2 Mapping from 3D to 2D space

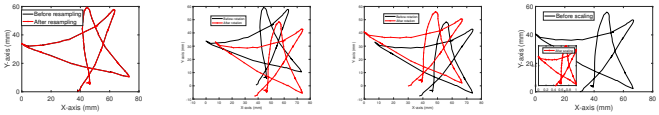
We map the 3D smartphone trace into the 2D space for easier recognition and verification later on. The first step maps the points on the trace into a 2D plane, which can be derived with linear polynomial surface fitting. This mapping is feasible, as simple gestures such as letters performed by most people usually lead to trace points almost in one plane. If the password gesture is complex such as a string of letters, this mapping process could be easily conducted after gesture segmentation. Then we rotate the 2D mapping plane into the X-Y plane of the 3D space because the mapping plane can be different each time the user performs the same gesture. Fig. 5 shows the mapping result.

4.1.3 Preprocessing

Since people cannot perform exactly the same gesture every time, we further preprocess the 2D trace for better recognition and verification performance in four steps.

Resampling. Each gesture trace is sampled at a fixed rate, say 100 Hz in our implementation. But the various movement speeds in different authentication instances result in diverse distributions of the sampling points. So we resample the gesture trace to obtain a new trace with 100 equidistant points. To accommodate the instability at the beginning and end of the gesture input, we remove several points (e.g., three in our experiments) in both ends of the 100-point trace.

Orientation invariance. To offset the bias caused by various gesture-input orientations, we rotate the resampled trace into a fixed orientation with the first method in [15]. We rotate the gesture until the direction vector $[x_n - x_1, y_n - y_1]$, directed from the first point to the last, is parallel with the X-axis using point rotation method, i.e., $[x', y']^T = [\cos\theta \ \ -\sin\theta; \sin\theta \ \ \cos\theta] * [x, y]^T$, where $\theta = -\arctan((y_n - y_1)/(x_n - x_1))$.



(a) Resampling (b) Rotation (c) Translation (d) Scaling

Fig. 6: Illustration of the preprocessing procedures.

Location invariance. It is nearly impossible for the same user to input his password gesture at the same position each time. To provide gesture-location invariance, we further shift the rotated trace to make the minimum X-coordinate and Y-coordinate among the trace points both non-negative.

Scale invariance. Users rarely input their gestures in exactly the same size each time. So we scale the trace by dividing the sequence of X-axis and Y-axis values using their maximum values, respectively [15], [16]. Thus the trace points are bounded by a square box [0,1]. The overall preprocessing procedures are shown in Fig 6.

4.1.4 Gesture recognition

A usable password gesture can comprise standard digits, diagrams, and letters similar to those in handwritten signatures. So the gesture input can be considered similar to signing in the air. We further require that each user's password gesture consist of a unique combination of digits, diagrams, and letters just like a computer password. So we can explore the efficient MyScript framework [8] to recognize individual digits/diagrams/letters from the processed 2D trace. The recognition result is stored at the system server for later authentication phases. Our preliminary evaluations show that the MyScript recognition accuracy for a letter-shape gesture is almost 100% for both one-magnet and two-magnet configurations, as shown in Fig. 7.

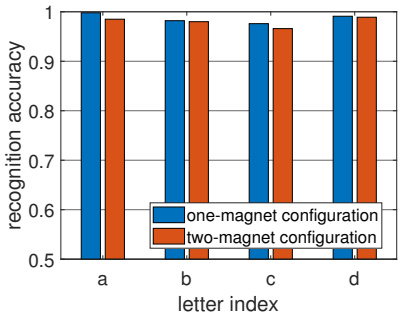


Fig. 7: Accuracy of gesture recognition using MyScript.

4.1.5 User identification

Gesture recognition relates to the traditional something-you-know authentication paradigm and can be enough for IoT applications without strong security requirements. But it can only provide coarse-grained user identification and is insufficient for security-critical IoT applications. So we further design the fine-grained user-identification module executed solely or after a successful gesture recognition. This module corresponds to the conventional someone-you-are authentication paradigm and trades off higher computational overhead for strong resilience to the adversary who

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3149746, IEEE Transactions on Mobile Computing

observes the user's password gesture through shoulder-surfing attacks or a spy camera. We consider and compare the DTW, traditional machine learning methods, and CNN methods for user identification in this paper.

DTW. Each legitimate user performs the password gesture a few times during the enrollment phase, leading to multiple processed 2D traces (or samples). We select the reference sample for the user as the one that has the minimum average DTW distance between itself and all the others. The system server is preloaded with the gesture traces of many random users as the training data. Then we compute the DTW distance between the reference sample and each other sample in the train data, where the samples from this user and other users are considered ground-truth positive and negative instances, respectively. Next, we obtain the optimal DTW classification threshold by maximizing the statistical F-1 score which is the harmonic mean of precision and recall.

Traditional machine learning. We tested the performance of SmartMagnet with popular classifiers including SVM, Naive Bayes (NB), KNN, and Random Forest (RF), and found that NB and Random Forest (RF) led to comparably better results. For both NB and RF, we first initialize 15 features in Table 1 to characterize the trajectory data. Then we apply the features in [17] (slope angle, path angle, and curvature) to represent the geometrical characteristics of the 2D movement trajectory.

TABLE 1: List of initialization features.

Feature	Description
Distance	$d_{xy}(i) = \sqrt{(x_{i+1}-x_i)^2 + (y_{i+1}-y_i)^2}$
Velocity	$\{v_{x,i} = \frac{x_{i+1}-x_i}{t_{i+1}-t_i}\}_{i=1}^{N-1}$ $\{v_{y,i} = \frac{y_{i+1}-y_i}{t_{i+1}-t_i}\}_{i=1}^{N-1}$
Acceleration	$\{acce_{x,i} = \frac{v_{x,i+1}-v_{x,i}}{t_{i+1}-t_i}\}_{i=1}^{N-1}$ $\{acce_{y,i} = \frac{v_{y,i+1}-v_{y,i}}{t_{i+1}-t_i}\}_{i=1}^{N-1}$
Slope Angle	$\{acce_{x,i} = \frac{v_{x,i+1}-v_{x,i}}{t_{i+1}-t_i}\}_{i=1}^{N-1}$
Path Angle	$alpha_{xy}(i) = \arccos \frac{\vec{p}(t)\cdot\vec{p}(t+1)}{\|\|\vec{p}(t)\|\|*\|\|\vec{p}(t+1)\|\|}$, $p(i) = [x_{i-1}-x_i; y_{i-1}-y_i]$
Curvature	$k_{xy}(i) = \frac{\|v_{x,i}*acce_{y,i} - v_{y,i}*acce_{x,i}\|}{(v_{x,i}^2+v_{y,i}^2)^{(3/2)}}$, $log_{k_{xy}(i)} = log(1/k_{xy}(i))$
Hu Invariant Moments	$M_i(\mu_{j,k}), i = 1, 2, ..., 7$, $\mu_{j,k} = \sum_x \sum_y (x-\bar{x})^j (y-\bar{y})^k$, $j, k \in \{0, 1, 2, 3\}$

In the next step, we convert each feature vector into a scalar using the root-mean-square (RMS) metric. Consider the velocity feature vector $\{v_{x,i}\}_{i=1}^{N-1}$ for the X-axis as an example. Let $\{\overline{v_{x,i}}\}_{i=1}^{N-1}$ denote the averages of multiple legitimate training samples for the same gesture. The RMS value of $\{v_{x,i}\}_{i=1}^{N-1}$ is computed as $\sqrt{\frac{1}{N-1}\sum_{j=1}^{N-1}(v_{x,j} - \overline{vx,j})^2}$. Finally, we obtain a scalar vector of 8 RMS values and 7 Hu-invariant moment values, one for each feature.

CNN. We also use a standard three-layer CNN. Each convolutional layer consists of a batch normalization layer, a Rectified Linear Unit (ReLU) activation layer, and a dropout layer. These layers extract features from the processed 2D traces, which are fed to a max-pooling module and then a fully connected (FC) layer. Finally, we use the sigmoid function to make a decision.

There are two remarks to make. First, all the models can be dynamically retrained on the availability of new training data. Second, any combination of the three models can be used, which depends on the system's security need.

4.2 Verification Phase

Suppose that someone interacts with an IoT device with the legitimate smartphone of a purported user. The user performs the password gesture with the smartphone which then submits the resulting magnetometer and attitude data to the server. The same workflow in Fig. 4 is then executed. The user is considered illegitimate if failing to pass the gesture-recognition module. Otherwise, the user-identification module is also invoked, in which any combination of the three models can be used. The user is considered legitimate if all the invoked modules yield successful results, in which case the last step of the system workflow outlined in Section 2 is executed to assign a session key between the IoT device and the smartphone. To tolerate inputting errors, the system may allow a threshold number of failed attempts in a short time window.

5 Performance Evaluation

5.1 Experimental Setup

We prototype SmartMagnet with the following hardware. The smartphone is an iPhone 6s, and the system server is a Dell desktop with 3.19 GHz CPU, 16 GB RAM, and Windows 10 64-bit Professional. We emulate both one-magnet and two-magnet IoT devices. Each magnet is NdFeB, Grade N40, 12.7mm*12.7mm*3.2mm (block shape), costing U.S. $1.5 at Amazon.com. The magnets are placed on a table and fully covered by standard printer paper.

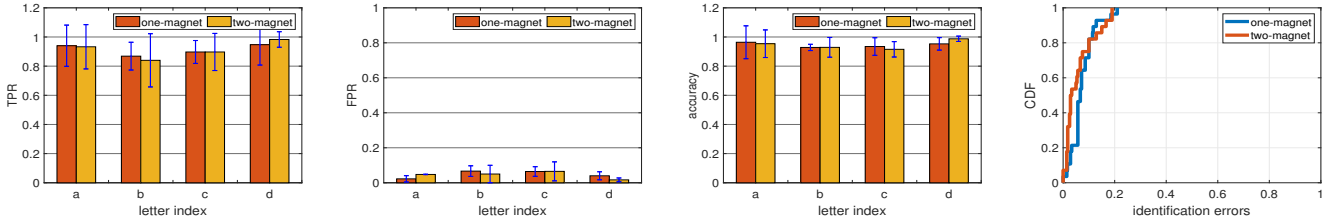
Our experiments involve 12 volunteers, all of whom are college students aged above 18. Each volunteer is required to perform his/her gestures within a given 20cm*20cm*20cm space centered around the magnet(s), emulating the practical scenario. Although there are plenty of options for a password gesture, we report the results with four low-case letters—a, b, c, and d—which have different complexity and shapes. Experiments of other gesture types are omitted for lack of space. Every volunteer inputs each letter gesture 15 times, leading to 720 samples in total. Each sample contains a timed sequence of 3-axis magnetometer readings and a sequence of 3-angle attitude readings. Since all our volunteers have the same choices of gestures, we actually evaluate the worst-case performance of SmartMagnet.

We evaluate the resilience of SmartMagnet to both signal-relaying and in-situ attacks per the adversary model in Section 2. Performance metrics include the true-positive rate (TPR), false-positive rate (FPR). Denote the number of true-positive, false-positive, true-negative, and false-negative samples by #TP, #FP, #TN, and #FN. We have TPR=#TP/(#TP+#FN), FPR=#FP/(#FP+#TN), and accuracy=(#TP+#TN)/(#TP+#FP+#FN+#TN).

5.2 Resilience to In-Situ Attacks

To emulate in-situ attacks, for each of the four letters, we randomly choose 10 samples from each volunteer and 10 from each other volunteer for the training, and the rest

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3149746, IEEE Transactions on Mobile Computing



(a) TPR (b) FPR (c) Accuracy (d) Identification errors

Fig. 8: SmartMagnet's resilience to in-situ attacks with the DTW method.

other 160 samples are used for testing. We use 5-fold cross-validation and compute the average results which are quite similar for different one-magnet and two-magnet configurations. Below we report the results for a random one-magnet configuration and the 4*th* two-magnet configuration in Table 5, unless specified otherwise.

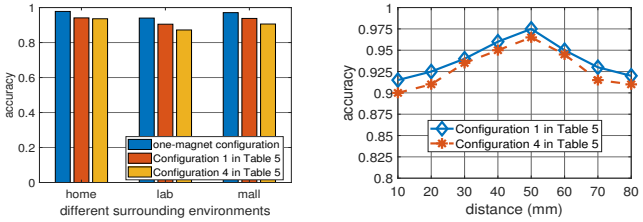
Fig. 8 shows the average result with the DTW method. From Fig. 8a, the TPRs for "a" and "d" are above 90%, while those for "b" and "c" are lower but still near 85%. After a further inspection, the reason is that the starting part of "b" and the whole "c" are both unclosed shapes, resulting in the large variation of the same user's gestures. Fig. 8b shows that the FPRs for all four letters are below 10%, confirming the resilience of SmartMagnet to in-situ attacks. Fig. 8c and Fig. 8d demonstrate very high accuracy of all letters (greater than 95%) and also low identification-error rates (mostly less than 10%) covering both false negatives and positives.

We also evaluate the authentication performance in three environments (lab, mall, and home). Each user is required to input Letter d and the results are shown in Fig. 9a. In most scenarios, the authentication accuracies are above 90%. Besides, the performance in the lab is worse than that in the other two environments due to the magnetic field interference induced by various electromagnetic devices. In addition, one-magnet configuration and two-magnet with orderly layouts perform better than complex magnet layouts because the former could generate a relatively simple and symmetric magnetic field which benefits the derivation of the gesture trace. Moreover, the result also verifies that the user can use the same gesture as any other IoT device.

Fig. 9b shows the impact of the distance $d_{(1,2)}$ between magnets 1 and 2. If $d_{(1,2)}$ is too large, the magnetic field generated by one magnet dominates the whole field when the user inputs the gesture near it, while the other magnet almost plays no role on magnetometer readings. If $d_{(1,2)}$ is too small, the resulting magnetic saturation would affect the performance as well. Therefore, there may exist an optimal $d_{(1,2)}$ distance value which is to be studied in future work.

TABLE 2: Performance of DTW, NB, RF, and CNN.

	one-magnet			two-magnet		
	TPR	**FPR**	**accuracy**	**TPR**	**FPR**	**accuracy**
DTW	0.915	0.048	0.946	0.913	0.045	0.947
NB	0.926	0.015	0.949	0.927	0.011	0.958
RF	0.942	0.030	0.955	0.931	0.026	0.961
CNN	0.963	0.035	0.978	0.960	0.024	0.981



(a) Different environments (b) Different distances

Fig. 9: Impact of environments and magnet distances.

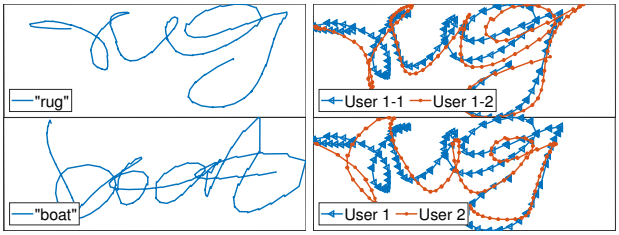
TABLE 3: Accuracy comparison of DTW, NB, RF, and CNN with various training samples.

training samples	**5**	**6**	**7**	**8**	**9**	**10**
DTW (one-mag)	0.887	0.904	0.921	0.925	0.935	0.946
NB (one-mag)	0.912	0.927	0.931	0.938	0.944	0.949
RF (one-mag)	0.924	0.939	0.942	0.947	0.952	0.955
CNN (one-mag)	0.953	0.968	0.970	0.973	0.977	0.978
DTW (two-mag)	0.906	0.918	0.922	0.923	0.931	0.947
NB (two-mag)	0.917	0.932	0.938	0.943	0.949	0.958
RF (two-mag)	0.926	0.949	0.953	0.956	0.959	0.961
CNN (two-mag)	0.955	0.971	0.974	0.976	0.980	0.981

Table 2 compares the classification performance of DTW, Naive Bayes (NB), Random Forest (RF), and CNN. The TPRs and accuracies of all methods are above 90% with the FPRs below 5%. CNN has the best overall performance at the cost of slightly larger computational overhead shown in Section 5.5. In addition, the performance of all methods improves as the number of training samples increases, as shown in Table 3. We also do not observe obvious performance differences with various password gestures and thus omit the results for simplicity.

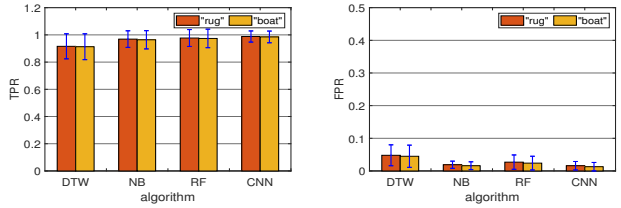
We further test the SmartMagnet performance with more complicated password gestures by using the words "rug" and "boat" in the white list of gestures in [12]) as examples. Fig. 10 illustrates their 2D trajectories after preprocessing in SmartMagnet. From Fig. 10b, we can see that the gesture trajectories are similar for the same user and quite diverse for different users. Fig. 11a shows the TPRs for "rug" and "boat" both above 90%. In particular, NB, RF, and CNN achieve significantly higher TPRs than DTW because DTW utilizes only one feature (i.e., the DTW distance of two samples) that incurs more errors when the gestures are complex. In addition, Fig. 11b shows that the FPRs are below 5% for all four classification methods and below 3% except DTW, confirming the higher resilience of SmartMagnet to in-situ attacks with more complex gesture passwords.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3149746, IEEE Transactions on Mobile Computing



(a) "rug" and "boat" (b) Trajectories of "rug" of the same user and different users

Fig. 10: 2D trajectories of "rug" and "boat" gestures.



(a) TPR (b) FPR

Fig. 11: Resilience to in-situ attacks with complex gestures.

5.3 Resilience to Remote Attacks

A remote attacker seeks to use the legitimate smartphone (lost or stolen) to access the target IoT device from a faraway location through stealthy wireless signal relaying. To pass the authentication, the attacker holds the smartphone to perform the victim's password gesture around an emulated magnetic field of the target IoT device. The magnetometer and attitude sensor data submitted to the system server relate to the emulated magnetic field. So the remote attack is successful only if two conditions are both satisfied: (1) the performed gesture closely match the legitimate one, and (2) the emulated magnetic field resembles the authentic one of the target IoT device. The remote attacker may guess the legitimate gesture to satisfy the first condition, but the results in Section 5.2 have confirmed that a random gesture has a little chance to be classified as legitimate by the server. In this section, we assume the worst-case scenario that the first condition is satisfied, which happens if the remote attacker can observe the victim's gesture-input process to do extensive practices or can be the legitimate user him/herself as a malicious insider (e.g., attempting to fake the presence at the work site). So we focus on evaluating the (in)feasibility of satisfying the second condition.

We emulate the remote attack as follows. According to Section 3.2.3, the reconstruction of the gesture trace needs some critical parameters that define the relatively unique magnetic environment of each IoT device. If the attacker can use his/her own magnets to generate these parameters and thus successfully emulate the authentic magnetic field, the remote attack succeeds. Our subsequent evaluations report the impact of each individual parameter by fixing other parameters for one-magnet and two-magnet scenarios.

Impact of unit orientation vectors. In this experiment, we assume that the remote attacker has the same magnet(s) in the target IoT device but does not know the magnet layout hidden inside the IoT device. The attacker can freely rotate all the magnets to eventually change their unit orientation vectors and relative orientations in the smartphone coordinate system, e.g., $\vec{m_1}$ and $m_{\vec{(1,2)}}$ in Fig. 2. Below we only report the DTW-based results due to space constraints, as other classification methods yield similar results.

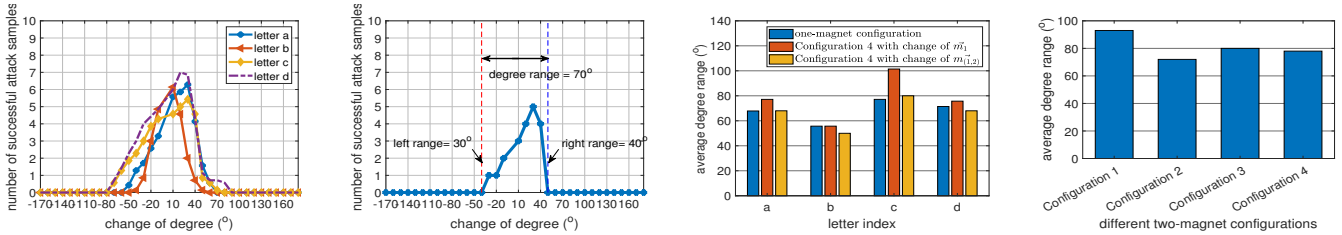
We first assume that the attacker rotates magnet 1 to only change the yaw angle of $\vec{m_1}$ from $-170°$ to $180°$ with an interval of $10°$. The rotation also changes the yaw angle of $m_{(1,2)}$ in the two-magnet setting. Each volunteer mimics a remote attacker and performs each of the four letter gestures under each raw angle, leading to 36 gesture samples per letter. We further assume that the correct $\vec{m_1}$ corresponded to $0°$, so each volunteer generates 35 attack samples per letter gesture for both one-magnet and two-magnet settings.

Fig. 12 shows the impact of the remote attack. As we can see in Fig. 12a, the number of successful attack samples decreases from the center $0°$ to both the left and right because a larger degree change induces more significant deviations from the authentic gesture trace. To quantitatively measure the remote-attack strength, we define a new metric—called *degree range*–to denote the degree interval centered at $0°$ in which there exist one or more successful attack samples. Fig. 12b plots the degree range for a randomly chosen volunteer. When the degree change reaches $-40°$ to the left and $50°$ to the right, the number of successful attacks samples is zero. So the degree range of this user is from $[30°, 40°]$. We also show the average degree range for each letter for both one-magnet and two-magnet configurations in Fig. 12c. The average degree ranges for "a", "b", and "d" are all below $80°$, while that for "c" is higher because the trajectory of "c" is too simple to have significant deviation with the degree change. Therefore, legitimate users are encouraged to choose more complex gestures. In addition, Table 5 shows that the average degree ranges with one magnet are smaller than those for two magnets, which is expected.

Based on the degree-range statistics above, we can estimate the number of brute-force trials for a successful remote attack. According to Fig. 2 and Section 3.2, the trace reconstruction is affected by $\vec{m_1}$, $\vec{m_2}$, and $m_{\vec{(1,2)}}$ for two magnets. The impacts of $\vec{m_1}$ and $\vec{m_2}$ are similar for the same degree change, while that of $m_{\vec{(1,2)}}$ is larger because it appears on the denominator to derive $\vec{r_2}$. For simplicity, we assume that $\vec{m_1}$, $\vec{m_2}$, and $m_{\vec{(1,2)}}$ have the same impact. As reported above, the average degree range for the yaw angle is $80°$, which divides $360°$ into equal-width ranges where the reconstructed traces are considered similar. Since the yaw, roll, and pitch angles of each orientation vector can be freely adjusted, a brute-forth remote attacker needs to try at least $(360/80)^9 \approx 756,680$ times for two magnets and $(360/80)^3 \approx 91$ times for one magnet. These results verify SmartMagnet's high resilience to the remote attack, especially when a practical IoT system rate-limits unsuccessful authentication attempts. Intuitively speaking, more magnets can construct a more complex magnetic field involving more parameters and much harder to crack.

Fig. 13 shows the number of successful attack samples for Letter "a" when the remote attacker could change only the yaw angle, yaw and roll angles, and all three angles of magnet 1. A smaller degree range corresponds to a lower probability of success. It is clear that multi-angle changes

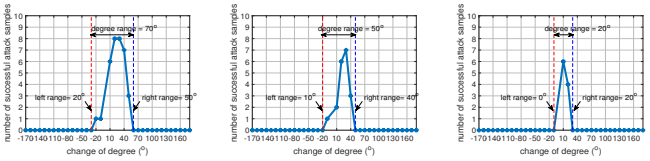
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3149746, IEEE Transactions on Mobile Computing



(a) # of successful attack samples (b) Degree range of a random user (c) Average degree range (d) Impact of $\vec{m_1}$, $\vec{m_2}$, and $m_{(\vec{1},2)}$

Fig. 12: Resilience to the remote attack.

would weaken the attack strength due to accumulated errors, which is consistent with our conjecture. So it is in the best interest of the remote attacker to only change one angle, say the yaw angle for Fig. 12. Equivalently speaking, we have emulated the most powerful remote attack above.



(a) Yaw (b) Yaw and roll (c) Yaw, roll and pitch

Fig. 13: Number of successful attack samples for letter "a" as the remote attacker changes different angles.

We further evaluate the impact of different two-magnet configurations in Table 5 and show the results in Fig. 12d. It is clear that orderly magnet layouts would result in worse resilience to the remote attack. The reason is that such orderly magnet layouts generate a relatively simple and symmetric magnetic field, which tends to smooth the variation of magnetometer readings in Eq. (2) caused by the degree change.

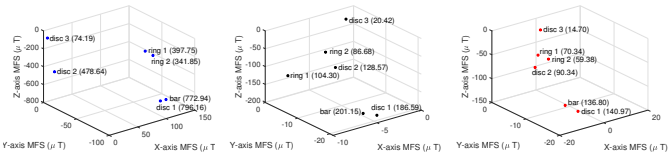
Impact of magnet material, size, and shape. The individual magnetic constant for each magnet (say, λ_1 and λ_2 in Eq. (2)) depends on the material, size, and shape of each magnet. Such constants affect the induced magnetic field strength (MFS) and thus the field distribution, especially for multiple-magnet configurations. Fig. 14 shows the average 3-axis MFS of the chosen magnets in Table 4, where the enclosed number following each magnet type indicates the magnitude of the 3D MFS vector. The larger the MFS magnitude, the stronger the corresponding magnet. So the MFS indeed relates to different materials, grades, and volumes under the same environmental conditions like heat, humidity, and radiation.

We have four particular observations which are quite consistent for each testing position. First, the magnet material is critical to the induced MFS. In our experiment, neodymium magnets always induce stronger MFS than ceramic ones with a similar volume. This result coincides with the common viewpoint that neodymium magnets are much stronger than ceramic ones. In addition, ring-1 and ring-2 ceramic magnets have the same volume, but the ring-1 magnet has a higher grade and thus always induces stronger MFS. Second, the volume of a magnet is proportional to its induced MFS, which can be easily observed, e.g., by comparing the MFS data of disc-1 and disc-2 magnets or those of disc-3 and ring-1 magnets. Third, the farther a magnet from the magnetometer at the upper-left corner of the used iPhone 6+, the weaker the induced MFS. Last, the magnet shape is less important for the induced MFS. In particular, disk-1 and bar magnets have the same material and similar volumes, leading to similar MFS despite their different shapes. So the magnets of the same material and similar volumes cannot be well distinguished by their induced MFS. To further validate the last observation, we test four rectangular magnets in the same batch and observe almost identical MFS at the same position.

TABLE 4: Magnets of various materials, grades, volumes, and costs.

magnet	material	grade	vol. (mm^3)	cost (US $)
disc 1	NdFeB	N45	7236	3.50
disc 2	NdFeB	N45	4824	2.35
bar	NdFeB	N45	7200	3.50
disc 3	ceramic	C5	4824	1.50
ring 1	ceramic	C8	6839	2.00
ring 2	ceramic	C5	6839	3.00

To sum up, the magnets of different materials and volumes are quite distinguishable by their induced MFS, but those of the same material and similar volume have less unique magnetic properties. Therefore, we can greatly boost SmartMagnet's security strength by considering the magnetic properties of various magnets when selecting internal magnets for IoT devices.



(a) MFS at the upper-right corner (μT) (b) MFS at the lower-left corner (μT) (c) MFS at the lower-right corner (μT)

Fig. 14: MFS of different magnets at the same position relative to the smartphone magnetometer.

5.4 Usability-Deployment-Security (UDS) Studies

We use the Usability-Deployability-Security (UDS) framework proposed in [9], [10] to evaluate the usability of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3149746, IEEE Transactions on Mobile Computing

TABLE 5: Different two-magnet configurations for Fig. 2a.

Configurations	Description
Configuration 1	$\vec{m_1}$, $\vec{m_2}$, and $m_{(\vec{1},2)}$ are aligned.
Configuration 2	$\vec{m_1}$ and $\vec{m_2}$ are aligned, but $m_{(\vec{1},2)}$ has one angle deviated by 90°.
Configuration 3	$\vec{m_1}$ and $m_{(\vec{1},2)}$ are aligned; $\vec{m_2}$ has one angle deviated by 90°.
Configuration 4	$\vec{m_1}$, $\vec{m_2}$, and $m_{(\vec{1},2)}$ are not aligned in pairs.

TABLE 6: Usability-deployment-security (UDS) evaluation.

	Usability						Dep.		Security								
	Memorywise-Effortless	*Nothing-to-Carry*	*Physically-Effortless*	*Infrequent-Errors*	*Easy-Recovery-from-Loss*	*No-Requirement-for-Interface*	*Accessible*	*Negligible-Cost-per-User*	*Resilient-to-Physical-Observation*	*Resilient-to-Targeted-Impersonation*	*Resilient-to-Internal-Observation*	*Resilient-to-Leaks-from-Other-Verifiers*	*Resilient-to-Phishing*	*Resilient-to-Theft*	*Requiring-Explicit-Consent*	*Insider-Identification*	*Resilient-to-Insider-Threat*
Password	○	○	○	●	●	○	●	●	○	○	○	○	○	○	●	○	○
Face	●	●	◐	○	○	○	◐	◐	●	◐	○	●	○	●	◐	●	●
Fingerprint	●	●	◐	○	○	◐	◐	○	●	◐	○	●	○	●	◐	●	●
Keystroke [18]	●	●	◐	◐	○	●	●	●	◐	○	●	○	●	○	●	○	●
Touch gesture [19]	○	●	○	◐	●	○	●	●	◐	○	●	○	●	○	●	○	●
Proximity [20]	●	◐	◐	◐	◐	●	●	◐	◐	○	●	○	○	○	○	○	○
SmartMagnet	○	◐	○	◐	●	●	●	◐	●	●	●	●	●	●	●	●	●

● = "offer" the benefit, ◐ = "somewhat offer" the benefit, and no circle = "does not offer" the benefit.

SmartMagnet. For comparison with other representative authentication schemes, we only use the relevant terms of the UDS framework. Specifically, Table 6 lists 18 UDS benefits and rates different schemes by determining if they satisfy these benefits.

Similar to other authentication schemes based on passwords or touch gestures, SmartMagnet requires the user to remember a self-chosen secret and perform it during the login process. This feature enables users to dynamically change their secrets, thus making SmartMagnet resilient to phishing attacks. In contrast, traditional face and fingerprint recognition schemes provide more convenience to users while exposing users to both phishing and smudge attacks. For example, there has been a lot of news on successful hackings of fingerprint and face scanners on both Android and iOS devices. Besides, the authentication schemes based on alphanumerical passwords, patterns, PINs, and touch gestures all require a UI which is unavailable on many IoT devices; the UI requirements are not friendly to senior citizens, children, and people with cognitive disabilities either. In contrast, SmartMagnet applies to no-UI IoT devices and explores ubiquitous commodity smartphones. Therefore, we place SmartMagnet into the "somewhat offering" category for the *Nothing-to-Carry* benefit. SmartMagnet also has a lower false reject rate (i.e., *Infrequent-Errors*) when compared with face and fingerprint recognition schemes. The reason is that face or fingerprint recognition needs to explore complex feature metrics to balance the inter-variance of different users and the intra-variance of the same user. By comparison, SmartMagnet could utilize complex gestures to augment the inter-variance, resulting in more reliable and robust authentication performance. Finally, SmartMagnet outperforms other schemes in terms of security benefits. To sum up, the analysis above verifies that SmartMagnet could simultaneously offer good usability and high security.

5.5 Computation Time

We also evaluate the computation time of SmartMagnet. The one-time enrollment time for inputting five gestures is less than $11\,\mathrm{s}$ which is comparable with that of finger or face recognition on smartphones. In addition, we train the classifier on a Dell desktop with 3.19 GHz CPU, 16 GB RAM, and Windows 10 64-bit Professional. The training and authentication time is less than $1\,\mathrm{s}$ for the DTW method and traditional machine learning methods, and less than $2\,\mathrm{s}$ for the CNN method. So the computational performance of SmartMagnet is quite acceptable in practice.

6 Related Work

There is extensive work (e.g., [21]–[24]) on using the magnetometer for text entry or interaction with mobile devices. In [21]–[23], researchers proposed to embed magnets in the writing pen or stylus for input/interaction. They utilized the association of magnets and the magnetometer installed in the mobile device. In addition, the authors in [21], [23] designed a new kind of wearable devices attached to magnetic sensors for 3D interaction and localization in a magnetic field environment. These methods require a large array of magnetic sensors to achieve localization which may not be possible for many applications. In contrast, our SmartMagnet utilizes the attitude sensor in the smartphone to assist the smartphone magnetometer for the localization, which can broaden the application scope. Authors in [24] proposed to monitor the movement of one driver's hands during driving by tracking the magnet attached in one hand with his smartwatch worn on the other hand. Their scheme assumes that the smartwatch is static while the magnets are moving. In contrast, the smartphone moves around the fixed magnets in our context, so we designed novel techniques to reconstruct the smartphone trajectory.

Researchers have also explored the magnetometer for user authentication. The authors in [25] let a user draw 3D signatures around the mobile device with a magnet and then authenticated the user. The authors in [26] investigated the possibility of 3D gesture authentication using multiple magnetic sensors deployed in advance. Besides, the authors in [27] studied the fingerprint hidden in the raw readings of sensors in the smart device such as gyroscope and magnetometer which utilizes the manufacturing imperfection. As far as we know, some device manufacturers have prohibited access to raw sensor data to avoid their illegitimate use. Overall, these papers studied the applicability of magnetometers on user authentication in a totally different application context from ours.

Significant research has been done for gesture-based authentication. In [28] and [29], the authors used the channel state information (CSI) of surrounding RF signals to recognize gestures and authenticate legitimate users. Their schemes require pre-deployed APs which may incur a pro-

TABLE 7: Comparison of different gesture-tracking systems.

Technical basis	DOF of Tracking	Tracking Error	Tracking Objectives	System Deployment
WiFi WiDraw [30]	3	5cm	finger	APs
RF RF-IDraw [38]	3	3.7cm	tags	RFID devices
sound LLAP [39]	1&2	3.5/4.6mm	finger	speakers and microphones
vision Kinect [40]	3	mm-level	finger	Microsoft KINECT
vision and RFID TagVision [41]	3	10.33mm	tags	cameras and RFID devices
IMUs Digits [42]	3	2°-9°	finger	wrist-worn sensor
magnetic field **SmartMagnet**	3	3mm	mobile devices	magnets

hibitive cost in a large-scale IoT system. Besides, the hand-movement trajectory derived from CSI has an error of nearly $5\,\mathrm{cm}$ in [30] that might affect the gesture-distinguishment performance. In [31], the authors localized one IoT device by using the inertial magnetometer to measure its bearing. Compared with our work, their localization accuracy is relatively lower with an average error of $13\,\mathrm{cm}$. The authors in [32] and [33] designed elaborate touch gestures to enhance password-based mobile device authentication. Such techniques can provide probabilistic defenses only against the in-situ attack. In contrast, MagDraw can also defeat the remote attack launched by the most powerful attacker who manages to perform the correct password gesture.

Our work is also related to the rich literature on user authentication in the established input environment. In [34], the authors utilized the hand-surface vibration response for biometric authentication by requiring a user to put his hand on a flat surface. In [35], the authors authenticated a user by the physical vibration of one specific surface the user inputs on. The authors in [36] integrated one small microphone into the earphone and required the user to listen to some sounds. By analyzing the backscattered signal, they used the earphone to authenticates the user. In [37], the authors used a static sound source to transmit an acoustic signal to the mobile device. By analyzing the received signal, they measured the location of the moving device. This line of work is orthogonal and complementary to our work.

Finally, there are some user-authentication schemes based on gesture tracking in the air [30], [38]–[42]. We list some representative gesture-tracking systems in Table 7. Compared with these tracking methods, SmartMagnet targets a totally different application context and does not need complex hardware except some cheap commodity magnets with nearly the smallest tracking error. Moreover, SmartMagnet users just need to perform gestures with ubiquitous smartphones instead of any extra device.

7 Conclusion and Future Work

In this paper, we presented the design and evaluation of SmartMagnet, a PBAC scheme for large IoT systems based on commodity smartphones and cheap magnets. Extensive experiments have confirmed that SmartMagnet is highly resilient to lost/stolen smartphones and also remote attacks via stealthy signal relaying. We have also shown that SmartMagnet is highly usable.

In SmartMagnet, we need to have prior knowledge of some magnet configuration parameters to recover the gesture trajectory, such as each magnetic constant. Such information may be hard to obtain due to the complex electromagnetic structures. Therefore, the extension of SmartMagnet under these scenarios is left as our future work. In addition, we will explore the potential of other mobile devices such as various wearables embedded with the attitude sensor and magnetometer to enhance SmartMagnet.

References

[1] K. Rasmussen, C. Castelluccia, T. Heydt-Benjamin, and S. Capkun, "Proximity-based access control for implantable medical devices," in *CCS*, Chicago, IL, November 2009.

[2] L. Xiao, Q. Yan, W. Lou, G. Chen, and Y. Hou, "Proximity-based security techniques for mobile users in wireless networks," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 12, pp. 2089–2100, 2013.

[3] J. Zhang, Z. Wang, Z. Yang, and Q. Zhang, "Proximity based iot device authentication," in *Infocom*, Atlanta, GA, May 2017.

[4] (2022) smartlock. [Online]. Available: https://www.pcmag.com/picks/the-best-smart-locks

[5] (2020) phonelost. [Online]. Available: https://www.phoneassured.com/blogs/news/phone-theft-is-on-the-rise-heres-what-you-need-to-know

[6] (2014) phonelostk. [Online]. Available: https://www.kaspersky.com/about/press-releases/2014_kaspersky-lab-survey-shows-employees-are-slow-to-report-stolen-mobile-devices

[7] B. Wang, Q. Chen, L. Yang, and H. Chao, "Indoor smartphone localization via fingerprint crowdsourcing: Challenges and approaches," *IEEE Wireless Communications*, vol. 23, no. 3, pp. 82–89, 2016.

[8] (2019) Myscript implementation. [Online]. Available: https://github.com/MyScript/interactive-ink-examples-android

[9] J. Bonneau, C. Herley, P. Oorschot, and F. Stajano, "The quest to replace passwords: A framework for comparative evaluation of web authentication schemes," in *2012 IEEE Symposium on Security and Privacy*, San Francisco, CA, May 2012.

[10] A. Acar, H. Aksu, A. Uluagac, and K. Akkaya, "A usable and robust continuous authentication framework using wearables," *IEEE Transactions on Mobile Computing*, vol. 20, no. 6, pp. 2140–2153, 2020.

[11] S. Song, B. Li, W. Qiao, C. Hu, H. Ren, H. Yu, Q. Zhang, M. Q.-H. Meng, and G. Xu, "6-d magnetic localization and orientation method for an annular magnet based on a closed-form analytical model," *IEEE Transactions on Magnetics*, vol. 50, no. 9, pp. 1–11, April 2014.

[12] E. Cheon, Y. Shin, J. H. Huh, H. Kim, and I. Oakley, "Gesture Authentication for Smartphones: Evaluation of Gesture Password Selection Policies," in *2020 IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA, May 2020.

[13] J. Bonneau, "The science of guessing: analyzing an anonymized corpus of 70 million passwords," in *2012 IEEE Symposium on Security and Privacy*, San Francisco, CA, May 2012.

[14] Y. Zhang, D. Han, A. Li, L. Zhang, T. Li, and Y. Zhang, "Magauth: Secure and usable two-factor authentication with magnetic wrist wearables," *IEEE Transactions on Mobile Computing*, 2021.

[15] C. Liu, G. Clark, and J. Lindqvist, "Guessing attacks on user-generated gesture passwords," in *ACM UbiComp*, New York, NY, September 2017.

[16] M. Sherman, G. Clark, Y. Yang, S. Sugrim, A. Modig, J. Lindqvist, A. Oulasvirta, and T. Roos, "User-generated free-form gestures for authentication: Security and memorability," in *ACM MobiSys*, Bretton Woods, NH, June 2014.

[17] J. Tian, C. Qu, W. Xu, and S. Wang, "Kinwrite: Handwriting-based authentication using kinect," in *NDSS*, San Diego, USA, February 2013.

[18] E. Miluzzo, A. Varshavsky, S. Balakrishnan, and R. Choudhury, "Tapprints: your finger taps have fingerprints," in *ACM MobiSys*, Low Wood Bay, Lake District, UK, June 2012.

[19] L. Li, X. Zhao, and G. Xue, "Unobservable re-authentication for smartphones," in *NDSS*, San Diego, USA, February 2013.

[20] K. Rasmussen, M. Roeschlin, I. Martinovic, and G. Tsudik, "Authentication using pulse-response biometrics," in *Internet Society NDSS*, San Diego, CA, February 2014.

[21] X. Han, H. Seki, Y. Kamiya, and M. Hikizu, "Wearable handwriting input device using magnetic field: Geomagnetism cancellation in position calculation," *Precision Engineering*, vol. 33, no. 1, pp. 37–43, January 2009.

[22] S. Yoon, K. Huo, and K. Ramani, "TMotion: Embedded 3D mobile input using magnetic sensing technique," in *ACM TEI*, Eindhoven, the Netherlands, February 2016.

[23] K.-Y. Chen, K. Lyons, S. White, and S. Patel, "uTrack: 3d input using two magnetic sensors," in *ACM UIST*, Scotland, United Kingdom, October 2013.

[24] H. Huang, H. Chen, and S. Lin, "MagTrack: Enabling safe driving monitoring with wearable magnetics," in *ACM MobiSys*, Seoul, Republic of Korea, June 2019.

[25] H. Ketabdar, K. Yuksel, A. Jahnbekam, M. Roshandel, and D. Skripko, "MagiSign: user identification/authentication based on 3d around device magnetic signatures," in *UBICOMM*, Florence, Italy, October 2010, pp. 31–34.

[26] K.-Y. Chen, S. N. Patel, and S. Keller, "Finexus: Tracking precise motions of multiple fingertips using magnetic sensing," in *ACM CHI*, San Jose, CA, May 2016.

[27] J. Zhang, A. Beresford, and I. Sheret, "SENSORID: sensor calibration fingerprinting for smartphones," in *IEEE SP*, San Francisco, CA, May 2019.

[28] S. Muhammad and S. Zhang, "Augmenting user identification with wifi based gesture recognition," in *ACM Ubicomp*, Singapore, October 2018.

[29] H. Kong, L. Lu, J. Yu, Y. Chen, L. Kong, and M. Li, "FingerPass: Finger Gesture-based Continuous User Authentication for Smart Homes Using Commodity WiFi," in *ACM Mobihoc*, Catania, Italy, July 2019.

[30] L. Sun, S. Sen, D. Koutsonikolas, and K.-H. Kim, "Widraw: Enabling hands-free drawing in the air on commodity wifi devices," in *ACM MobiCom*, Paris, France, September 2015.

[31] P. Appavoo, M. Chan, and A. Bhojan, "MagB: Repurposing the magnetometer for fine-grained localization of iot devices," in *IEEE INFOCOM*, online, July 2020.

[32] J. Sun, X. Chen, J. Zhang, Y. Zhang, and J. Zhang, "TouchIn: Sightless two-factor authentication on multi-touch mobile devices," in *IEEE CNS*, San Francisco, CA, October 2014.

[33] N. Sae-Bae, N. Memon, K. Isbister, and K. Ahmed, "Multitouch gesture-based authentication," *IEEE transactions on information forensics and security*, vol. 9, no. 4, pp. 568–582, 2014.

[34] J. Li, K. Fawaz, and Y. Kim, "Velody: nonlinear vibration challenge-response for resilient user authentication," in *ACM CCS*, London, United Kingdom, November 2019.

[35] J. Liu, C. Wang, Y. Chen, and N. Saxena, "VibWrite: Towards finger-input authentication on ubiquitous surfaces via physical vibration," in *ACM CCS*, Dallas, Texas, October 2017.

[36] Y. Gao, W. Wang, V. Phoha, W. Sun, and Z. Jin, "EarEcho: Using ear canal echo for wearable authentication," in *ACM UbiComp*, London, UK, September 2019.

[37] Y. Zhang, J. Wang, W. Wang, Z. Wang, and Y. Liu, "Vernier: Accurate and fast acoustic motion tracking using mobile devices," in *IEEE INFOCOM*, Honolulu, HI, April 2018.

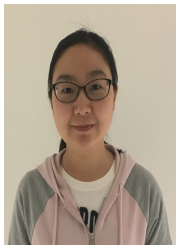
[38] J. Wang, D. Vasisht, and D. Katabi, "RF-IDraw: virtual touch screen in the air using RF signals," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4, pp. 235–246, 2014.

[39] W. Wang, A. X. Liu, and K. Sun, "Device-free gesture tracking using acoustic signals," in *ACM MobiCom*, New York, October 2016.

[40] J. Raheja, A. Chaudhary, and K. Singal, "Tracking of fingertips and centers of palm using kinect," in *IEEE CIMSim*, Langkawi, Malaysia, September 2011.

[41] C. Duan, X. Rao, L. Yang, and Y. Liu, "Fusing rfid and computer vision for fine-grained object tracking," in *IEEE INFOCOM*, Atlanta, GA, May 2017.

[42] D. Kim, O. Hilliges, S. Izadi, A. Butler, J. Chen, I. Oikonomidis, and P. Olivier, "Digits: freehand 3d interactions anywhere using a wrist-worn gloveless sensor," in *ACM UIST*, Cambridge, MA, October 2012.

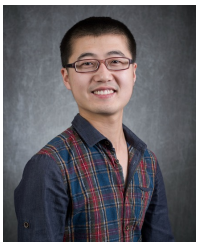


Yan Zhang received the B.S. in Information and Computing Science from Xi'an Jiaotong University, China, in 2014, the M.S. in Communication and Information System from Beijing Normal University, in 2017. She is a Ph.D. student in Computer Engineering at Arizona State University. Her research interest focuses on cyber security and privacy issues in mobile systems.

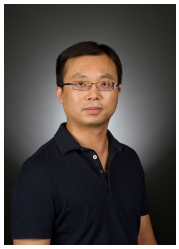
Dianqi Han received the B.S. in Information Security from University of Science and Technology of China, China, in 2010, the M.S. in Electrical and Computer Engineering from University of California, Davis, in 2015. Currently, he is a Ph.D. student in Computer Engineering from Arizona State University. His research interest is about indoor navigation, security and privacy issues in computer and networked systems.

Ang Li received the B.E. in Network Engineering from Guangxi University, China, in 2010, the M.S. in Computer Science from Beihang University, China, in 2014. Currently, he is a Ph.D. student in Computer Engineering from Arizona State University. His research interest is about security and privacy in social networks, machine learning, wireless networks, and mobile computing.

Jiawei Li is a Ph.D. student in Computer Engineering at Arizona State University. He received the B.E. in Telecommunication Engineering from Nanjing University of Posts and Telecommunications at 2013. His research interest is on security and privacy issues in wireless network and wireless sensing.



Tao Li received a Ph.D. in Computer Engineering from Arizona State University in 2020, a M.S. in Computer Science & Technology from Xi'an Jiaotong University in 2015, and a B.E. in Software Engineering from Hangzhou Dianzi University in 2012. His primary research is on security and privacy issues in networked/mobile/distributed systems, smart sensing, and wireless networks. He is an Assistant Professor in the Department of Computer and Information Technology at Indiana University-Purdue University Indianapolis (IUPUI).



Yanchao Zhang received the B.E. in Computer Science and Technology from Nanjing University of Posts and Telecommunications in 1999, the M.E. in Computer Science and Technology from Beijing University of Posts and Telecommunications in 2002, and the Ph.D. in Electrical and Computer Engineering from the University of Florida in 2006. He is an Professor in School of Electrical, Computer and Energy Engineering at Arizona State University. His primary research interests are network and distributed system security, wireless networking, and mobile computing. He is/was on the editorial boards of IEEE/ACM Transactions on Networking, IEEE Transactions on Mobile Computing, IEEE Wireless Communications, IEEE Transactions on Control of Network Systems, and IEEE Transactions on Vehicular Technology. He received the US NSF CAREER Award in 2009 and is an IEEE Fellow for contributions to wireless and mobile security.