MagAuth: Secure and Usable Two-Factor Authentication with Magnetic Wrist Wearables

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

Abstract— Secure and usable user authentication is the first line of defense against cyber attacks on smart end-user devices. Advanced hacking techniques pose severe threats to the traditional authentication systems based on the password/PIN/fingerprint. We propose MagAuth, a secure and usable two-factor authentication scheme with commercial off-the-shelf (COTS) wrist wearables with magnetic strap bands to enhance the security and usability of password-based authentication for mobile touchscreen devices. In MagAuth, a user enrolls a self-chosen unlock pattern or touch gesture into his touchscreen device by performing it with the same hand the magnetic wrist wearable is on. The chosen unlock pattern or touch gesture serves as the first authentication factor, and the user's behavioral features manifested in the magnetic field changes during his finger movement correspond to the second factor. The user can unlock his touchscreen device only when both authentication factors can be validated. Comprehensive user experiments confirm the high security and usability of MagAuth. In particular, MagAuth achieves an average true-positive rate up to 96.3% and a false-positive rate no larger than 8.4%. Moreover, we show that MagAuth is highly resilient to various attacks.

Index Terms—Mobile authentication, security, usability.

1 Introduction

Secure and usable user authentication is the first line of defense again cyber attacks on smart end-user devices like smartphones, tablets, wearables, and smart home devices. Most such devices store private user information and are also entry points to home/enterprise networks and the Internet. A secure authentication scheme can prevent illegitimate users from logging into a protected device to access sensitive information therein or even launch more severe attacks on the network with the hacked device as the stepping stone. In contrast, a usable authentication scheme means that it should be very easy to use by the legitimate user. There is often a natural conflict between the security and usability requirements. Popular authentication schemes can be classified into the following categories with each having its merits and drawbacks.

Password-based authentication is the most widely used. Familiar examples include PINs, alphanumeric passwords, Android pattern locks, etc. Complex passwords relate to stronger security but lower usability, as they are more difficult to remember and input, especially by senior citizens, children, and people with cognitive disability.

Biometric authentication is also seeing popular usage. Examples are fingerprint and face authentication functions on the latest mobile devices. Although very easy to use, biometric authentication has a known disadvantage that a user cannot alter his[1] biometric identifier once cloned by an attacker. There has been a lot of news on successful hackings of fingerprint and face scanners on both Android and iOS devices [1]–[4]. For instance, FaceID released by Apple can be cracked with a composite mask [3] and cannot distinguish family members who are not even alike [4].

Hardware token-based authentication explores an extra miniature device each legitimate user should own and carry. When the user tries to log into his device, the token transmits a user-chosen password to the device on behalf of the user. Such devices [5], [6] have to be specially built and are not available on the market. In addition, these techniques actually authenticate the hardware token rather than the user to the device, so an attacker stealing the token can log into the device.

Touch-based authentication harnesses a user's behavioral biometrics exhibited in his random or predefined finger-touch gestures performed on touchscreen devices. Most existing research on touch-based authentication uses behaviorial features such as stroke time, touching pressure, device motion, the shape of the swiping trajectories, and tapping rhythms passively recorded by the device [7]–[11]. These features are relatively simple and vulnerable to mimicry attacks [12] because they are only related to the fingertip. Another category of systems actively generate acoustic or vibration signals and explore a user's unique response to the signals for authentication [13]–[15]. The systems using acoustic signals [13] are highly susceptible to environmental interference, and those based on device vibration [14], [15] may generate audible noise which is not appropriate for many naturally quiet environments such as the meeting room and classroom.

In this paper, we propose **MagAuth**, a secure and usable two-factor authentication scheme with commercial off-the-shelf (COTS) wrist wearables with magnetic strap bands/clasps as hardware tokens to enhance the security

This work was supported in part by the U.S. National Science Foundation under grants CNS-1619251, CNS-1824355, CNS-1933069, CNS-2055751.

Y. Zhang, D. Han, A. Li, L. Zhang 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, lilizhang, tli, yczhang}@asu.edu).

T. Li is with the Computer and Information Technology Department, Indiana University-Purdue University Indianapolis, Indianapolis, IN 46202 (E-mail: tli6@iupui.edu).

1. No gender implication.

and usability of password-based mobile authentication techniques. MagAuth is motivated by three observations. *First*, a cheap magnetometer is available on growing end-user devices such as smartphones and IoT devices. *Second*, there has been a massive production of wrist wearables [16], [17] such as smartwatches and fitness trackers that are equipped with magnetic strap bands and/or clasps with strong magnetism [18], [19], and we refer to such devices as *magnetic wrist wearables* hereafter. *Last*, when a user inputs an unlock pattern or touch gesture on a touchscreen device (say, a smartphone or tablet) with the same hand wearing the magnetic wrist wearable, his finger movement can induce magnetic field changes that can be sensed by the device's inertial magnetometer. Each user's finger movement relative to his wrist may be unique enough due to his palm size, finger lengths, and input habits. For example, when a user inputs an unlock pattern while wearing a smartwatch with a magnetic clasp, his unique finger movement is manifested by the time sequences of orientation and displacement vectors from the varying touch points on the touchscreen to the center of the magnetic clasp. MagAuth explores such natural behavioral features to enhance the security of conventional password-based authentication schemes.

MagAuth is naturally a two-factor authentication scheme. In particular, a MagAuth user buys a magnetic strap band/band for his wrist wearable and then enrolls his self-chosen unlock pattern or touch gesture by performing it with the same hand the wrist wearable is on. The chosen unlock pattern or touch gesture serves as the first authentication factor, and the user's behavioral features manifested in the magnetic field changes during his finger movement correspond to the second factor. The user can unlock his touchscreen device only when both authentication factors can be validated. Since there is no need for any specially-built hardware token or additional user effort, MagAuth can be highly usable.

MagAuth aims to complement rather than completely replace password-based authentication techniques for touchscreen devices. For example, a user needs to input a password each time he wants to unlock his password-protected smartphone. A complex alphanumerical password can obviously enhance his device security. But average iPhone and Android users unlock their phones 80 and 110 times per day, respectively [20]. It is quite inconvenient for the user to input the complex password for each unlocking attempt. With MagAuth in place, the user can input the complex password only when powering on his smartphone and then use MagAuth along with a simpler unlock pattern or touch gesture for subsequent smartphone unlockings. MagAuth can basically apply to any end-user device with an internal magnetometer and a touchscreen.

Our contributions can be summarized as follows.

We propose MagAuth, a new two-factor authentication method to enhance the security and usability of password-based authentication techniques for smart touchscreen devices. MagAuth requires a user to input a self-chosen unlock pattern or touch gesture on the device's touchscreen with the same hand wearing a magnetic wearable. The user is authenticated if the unlock pattern or touch gesture he inputs and his behavioral characteristics exhibited in the inputting process both match those stored in the mobile device. To the best of our knowledge, we are the first to explore the movement features of a user's finger relative to his wrist along with magnetic wrist wearables to achieve user authentication.

We conduct comprehensive experiments to evaluate the security and usability of MagAuth. Our experimental studies involved 23 volunteers and over 1,800 samples. We show that MagAuth is highly secure with the true-positive rate up to 96.3% and false-positive rate no larger than 8.4% in all the testing scenarios. In addition, the enrollment time and authentication time of MagAuth are comparable to those of face or finger authentication on COTS mobile devices. In addition, we use the volunteers to verify the performance of MagAuth in different application scenarios with various magnetic bands/clasps and smartphones.

The rest of this paper is organized as follows. Section 2 introduces the background knowledge about magnetic field theory and magnetometers. Section 3 presents the MagAuth design. Section 4 evaluates the performance of MagAuth. Section 5 discusses the related work. Section 6 concludes this paper.

2 Preliminaries

For ease of illustration, we assume a smartphone as the target device to protect, though MagAuth can apply to virtually any end-user device with an internal magnetometer and a touchscreen. The smartphone user protects his device with a strong system password such as an alphanumeric password or a complex unlock pattern. The user needs to input his system password when turning on his smartphone or performing critical operations such as system updates, which is the current practice even if face or fingerprint authentication is available. As mentioned, average iPhone and Android users unlock their phones 80 and 110 times per day, respectively [20]. MagAuth aims to help the user unlock his smartphone in a more secure and usable fashion after the user powers on his smartphone. In this section, we first introduce some background on magnetic field and magnetometer to help illustrate the MagAuth design.

2.1 Briefs of Magnetic Field and Magnetometer

A magnet is an object that produces a magnetic field. A permanent magnet is made from some magnetized material and maintains its own magnetic field. In contrast, an electromagnet acts as a magnet only when an electric current runs through it. They have found tremendous household applications and are available in various shapes and sizes like rings, bracelets, and watch bands. MagAuth uses COTS magnetic strap bands with some small magnets embedded inside them [18] or COTS stainless steel bands with magnetic closure [19]. This paper uses the latter ones. The extension of our studies to other types of magnetic products is left as future work.

The magnetic moment of a magnet is a vector that characterizes its overall magnetic properties. Consider a bar magnet as an example. The direction of its magnetic moment points from its south pole to its north pole, and the corresponding magnitude relates to how strong and how far apart these poles are. The magnetic field vector (MFV)

produced by a magnet at any given point is proportional to the magnitude of its magnetic moment [21]. It can be made from many types of materials such as composites, alloys, rare-earth elements, and nano-structured materials [22], [23].

According to the magnetic field theory [24], the magnet-induced three-dimensional (3D) MFV measured by a nearby magnetometer can be approximated by

$$H(\vec{r}) = \frac{\lambda}{4\pi|\vec{r}|^3}\left[\frac{3\vec{r}(\vec{m}^\top\vec{r})}{|\vec{r}|^2} - \vec{m}\right], \tag{1}$$

$$\lambda = \frac{B_r V}{4\pi\mu_0} * \mu_0, \tag{2}$$

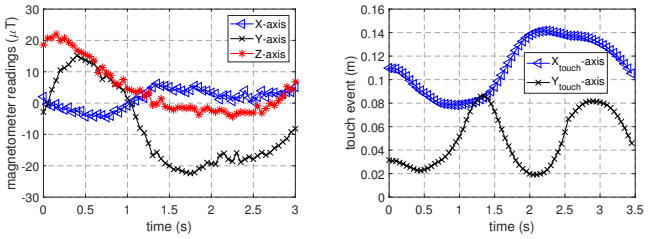
where $\lambda = B_r V/(4\pi\mu_0) * \mu_0$. Here λ is a constant related with magnetic moment, $\vec{r} := \langle r_x, r_y, r_z\rangle$ and $\vec{m} := \langle m_x, m_y, m_z\rangle$ represent the 3D distance of the magnet relative to the magnetometer and directional unit vectors of magnetic moment, and all the variables take values in the magnetometer's coordinate system as is shown in Fig. 2. μ_0 is the permeability of a vacuum (unit: N/A^2), B_r is the Residual Flux Density (unit: Tesla), V is the volume of the magnet (unit: m^3). Since λ is approximately a constant for each given magnet, the MFV changes relate with the 3D relative position and direction between the magnetometer and magnet.

A magnetometer is a standard miniature Hall-effect sensor in smartphones, tablets, and other smart end-user devices, which detects the Earth's magnetic field along three perpendicular axes. The magnetometer is often used as a compass and crucial for detecting the orientation of the smartphone relative to the Earth's magnetic north. The price of three-axis magnetometers is well below US $1 per device. The magnetometer outputs a time series of MFV measurements in the format of $\langle t, H_x(t), H_y(t), H_z(t)\rangle$, where t is a timestamp, and $H_x(t)$, $H_y(t)$, and $H_z(t)$ denote the MFV in micro-Tesla (μT) at time t along the X-axis, Y-axis, and Z-axis, respectively. Because there exists environmental magnetic field such as the geomagnetic field, to measure the magnetic field induced only by the tiny magnet, we need to cancel the impact of environmental magnetic field before further data processing which is demonstrated in Section 3. We can greatly boost the security strength of mobile devices by combining magnetic properties with pattern/gesture-based authentication techniques, as shown in Section 3.

2.2 Swipe Events on Touchscreen Devices

MagAuth collects data about a swipe event on the device touchscreen as a time series $\langle t, r_x^{touch}, r_y^{touch}\rangle$, where t is the time stamp of the event, and r_x^{touch} and r_y^{touch} are the x and y coordinates of the touch point, respectively. It is worth mentioning that one finger touch may generate multiple data samples which have little difference because of imperceptible finger movement. Therefore, we downsample the collected data to facilitate subsequent data processing.

Fig. 1 shows the magnetometer data when a user inputs the unlock pattern, Pattern 2, in Fig. 6 on the touchscreen with the upper-left corner as the start point. This experiment used a COTS magnetic **clasp** [19] and a Samsung Galaxy S5 with a magnetometer on the upper-right corner of the phone



(a) magnetometer data (b) swipe event data

Fig. 1: Illustration of magnetometer and swipe event data.

front. The phone was placed on the table, and the sampling rates of the touch sensor and the magnetometer were both 100 Hz. We derived the touch gesture and MFV data sequences for the same trajectory after proper time alignment. The figure clearly demonstrates the nonlinear relationship between the MFV and relative magnet-magnetometer position. In addition, elaborate magnetic motions can lead to varying MFV data that can be explored for user authentication.

2.3 Association of Magnetometer and Swipe Events

When the user performs an unlock pattern or touch gesture with the wrist wearable on the same hand, the relative 3D magnetometer-magnet position changes with time. The distance vector $\vec{r}$ and direction vector $\vec{m}$ in Eq. (1) thus both become a function of time t. So does the MFV measurement $H(\vec{r})$. For simplicity, we abuse Eq. (1) without introducing the variable t and expand it into three equations corresponding to the three axes:

$$H_x = \frac{\lambda[-3r_x(-m_x r_x - m_y r_y - m_z r_z) - m_x(r_x^2+r_y^2+r_z^2)]}{4\pi(r_x^2+r_y^2+r_z^2)^{5/2}}, \tag{3}$$

$$H_y = \frac{\lambda[-3r_y(-m_x r_x - m_y r_y - m_z r_z) - m_y(r_x^2+r_y^2+r_z^2)]}{4\pi(r_x^2+r_y^2+r_z^2)^{5/2}}, \tag{4}$$

$$H_z = \frac{\lambda[-3r_z(-m_x r_x - m_y r_y - m_z r_z) - m_z(r_x^2+r_y^2+r_z^2)]}{4\pi(r_x^2+r_y^2+r_z^2)^{5/2}}. \tag{5}$$

Here r_x, r_y, r_z could be expressed as,

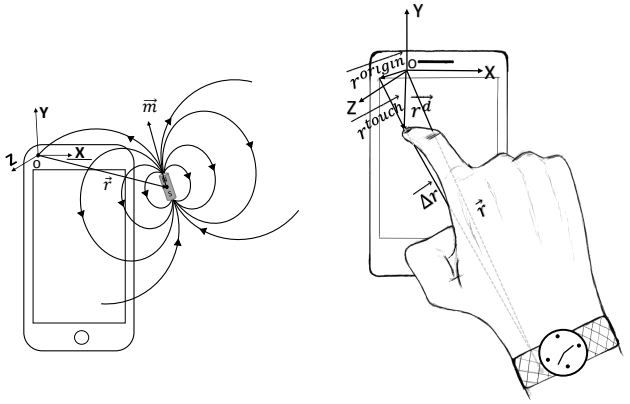
$$r_x = r_x^d + \Delta x, r_y = r_y^d + \Delta y, r_z = r_z^d + \Delta z, \tag{6}$$

where

$$r_x^d = r_x^{origin} + r_x^{touch}, r_y^d = r_y^{origin} + r_y^{touch}, r_z^d = r_z^{origin} + r_z^{touch}. \tag{7}$$

Here r_x^d, Δx, and r_x^{origin} are the X-axis values of displacement vectors from the origin of the magnetometer to the touch point, from the touch point to the magnet, and from the origin of the magnetometer to that of touch screen, respectively. The illustration is shown in 2b. In addition, dividing Eq. (3) and Eq. (4) by Eq. (4) and Eq. (5), respectively, we obtain

$$\frac{H_x}{H_y} = \frac{3r_x(m_x r_x + m_y r_y + m_z r_z) - m_x(r_x^2+r_y^2+r_z^2)}{3r_y(m_x r_x + m_y r_y + m_z r_z) - m_y(r_x^2+r_y^2+r_z^2)}, \tag{8}$$



(a) magnet-induced 3D MFV (b) finger movement

Fig. 2: (a) Illustration of magnet clasp-induced 3D MFV; (b) when a user inputs an unlock pattern or touch gesture, the inertial magnetometer on the smartphone generates an MFV sequence.

and

$$\frac{H_y}{H_z} = \frac{3r_y(m_x r_x + m_y r_y + m_z r_z) - m_y(r_x^2 + r_y^2 + r_z^2)}{3r_z(m_x r_x + m_y r_y + m_z r_z) - m_z(r_x^2 + r_y^2 + r_z^2)]}. \quad (9)$$

According to the analysis and derivation process above, we have three remarks to make for Eq. (3), Eq. (4), and Eq. (5).

- Remark 1: Given the value of $\langle r_x^{origin}, r_y^{origin}, r_z^{origin}\rangle$ and $\langle r_x^{touch}, r_y^{touch}, r_z^{touch}\rangle$, Eq. (3), Eq. (4), and Eq. (5) form an equation set with six variables: Index $= \langle m_x, m_y, m_z, \Delta x, \Delta y, \Delta z\rangle$. Table 1 lists the parameter values used in deriving displacement and orientation vectors which are stored on the user device.
- Remark 2: Using two pairs of the equation set mentioned in Remark 1, we can derive Index with a known constant λ. Here we define $\langle m_x, m_y, m_z\rangle$ and $\langle \Delta x, \Delta y, \Delta z\rangle$ as the *orientation* and *displacement* vectors of finger movement, respectively. We assume that the displacement vectors of adjacent two sampling points from the finger tip to the small magnetic clasp (i.e. the values of $\Delta x, \Delta y, \Delta z$) remain unchanged during finger movement. The assumption is reasonable because the interval between two adjacent sampling points is too short for the change of the finger's bending. We can use the optimization methods to solve the equation set to obtain the values of the two vectors [25]. Note that because of measurement errors of r^{origin}, r^{touch} and $\vec{H}$, there may be non-negligible accumulated errors in the final solution such that the derived orientation vector does not satisfy $m_x^2 + m_y^2 + m_z^2 = 1$. Thus, we use the Levenberg-Marquardt (LM) algorithm because of its computational efficiency with the constraint mentioned above. In addition, we take the orientation equation as a constraint of the optimization problem. Note that Index contains the behavioral features of the finger movement which could be used to uniquely identify the user.

TABLE 1: parameters for deriving displacement and orientation vectors.

parameters	r_x^{origin}	r_y^{origin}	r_z^{origin}	r_z^{touch}
approximate value (mm)	-62.50	2.00	3.00	0.00

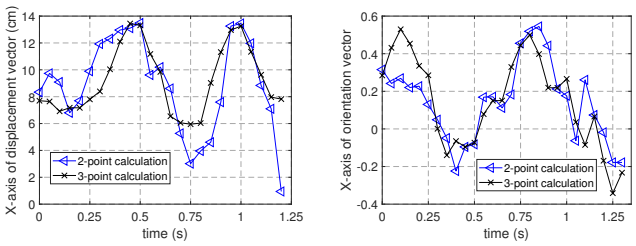
- Remark 3: Using three pairs of Eq. (8) and Eq. (9), we can obtain Index. Because Eq. (8) and Eq. (9) have canceled the impact of the magnetic moment λ, user authentication could succeed for any magnetic strap band which could induce significant changes in the magnetometer data. This observation can inspire common users to utilize our authentication scheme because there is no specific requirement for selecting a strap band. Here we also assume that the displacement vectors of three adjacent sampling points from the finger tip to the small magnetic clasp remain unchanged during finger movement. Again, we use the LM algorithm to solve the equation set in our experiments and evaluation.

The above remarks drive the choice of the unlock pattern or touch gesture in MagAuth. In particular, the first remark indicates that elaborating unlock patterns or touch gestures with large position changes can induce totally different values of $H(\vec{r})$ due to signal attenuation and the nonlinear relationship of $\vec{r}$ and $H(\vec{r})$. In reality, the distinctiveness of $H(\vec{r})$s benefits solving the equation set. Furthermore, more complex unlock patterns or touch gestures always induce more elaborate finger movement which can make the user's behavioral features more representative.

To further illustrate these three remarks, we let a user input the unlock pattern, Pattern 2, in Fig. 6 on the touchscreen of a smartphone placed on the table. This experiment used a COTS tiny rectangle magnet (12.7mm*3.2mm*3.2mm) whose magnetic moment is known and a Samsung Galaxy S5 with a magnetometer on the upper-right corner of the phone front. The touch sensor's and the magnetometer's sampling rates were both 100 Hz. Then we derived the displacement and orientation vectors from the touch-event and MFV data sequences. We utilize the two methods proposed in Remark 2 and 3 and compare the curves after resampling. The results are shown in Fig. 3. The figure clearly verifies the feasibility of our schemes. In addition, the displacement and orientation readings are similar for 2-point and 3-point methods as expected. In Section 2.4, we discuss how the derived displacement and orientation readings can be used as the second factor during the authentication process based on the pattern lock or touch gesture.

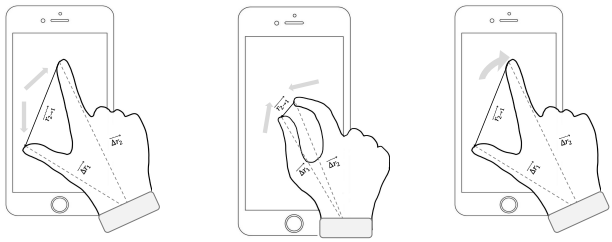
2.4 Pattern Lock and Touch Gesture

In this paper, we focus on two authentication modes for MagAuth: pattern lock and touch gesture. For both modes, the user should use the same hand wearing the magnetic wearable for authentication to unlock his smartphone. For the pattern-lock mode, the user performs a carefully chosen unlock pattern such as those in Fig. 6. For the touch-gesture mode, the user uses one or two fingers to perform a self-chosen touch gesture at arbitrary positions on the smartphone's touchscreen such as pinch and zoom in Fig. 4.



(a) displacement vector calculation (b) orientation vector calculation

Fig. 3: 2-point and 3-point displacement and orientation vector calculations for Pattern 2 in Fig. 6.



(a) zoom (b) pinch (c) rotation

Fig. 4: Displacement vectors for a multi-touch gesture, $\vec{\Delta r_1} = \langle \Delta x_1, \Delta y_1, \Delta z_1 \rangle$ and $\vec{\Delta r_2} = \langle \Delta x_2, \Delta y_2, \Delta z_2 \rangle$.

Muiti-touch gesture data collection. For an unlock pattern or a single-finger touch gesture, we can derive one displacement vector and one orientation vector for one trial. It is more challenging to handle multi-touch gestures. Take the two-finger touch gesture as an example. The touch sensor records two time sequences with one for each finger, which are denoted by $r_1^{touch} = \langle r_{x,1}^{touch}, r_{y,1}^{touch}, r_{z,1}^{touch} \rangle$ and $r_2^{touch} = \langle r_{x,2}^{touch}, r_{y,2}^{touch}, r_{z,2}^{touch} \rangle$, respectively. With r_1^{touch} and the magnetometer data, we can derive one displacement vector for finger 1, $\vec{\Delta r_1} = \langle \Delta x_1, \Delta y_1, \Delta z_1 \rangle$. Moreover, the displacement for finger 2, $\vec{\Delta r_2}$, could be derived as

$$\Delta x_2 = \Delta x_1 + x_{2\to1}, \Delta y_2 = \Delta y_1 + y_{2\to1}, \Delta z_2 = \Delta z_1 + z_{2\to1} \quad (10)$$

where $\langle x_{2\to1}, y_{2\to1}, z_{2\to1} \rangle$ is the distance vector from finger 2 to finger 1. Due to the uniqueness of each user's palm size, finger lengths, and swipe habits, these two displacement vectors may represent a user's unique behavioral features. Therefore, we could collect multiple displacement vectors and one orientation vector for the multi-touch gesture mode, e.g., $\langle m_x, m_y, m_z, \Delta x_1, \Delta y_1, \Delta z_1, \Delta x_2, \Delta y_2, \Delta z_2 \rangle$ illustrated in Fig. 4. Then these vectors are fed into the classification module to distinguish different users.

3 MagAuth Design

In this section, we illustrate the design of MagAuth which consists of two phases. In the enrollment phase, the user performs a self-chosen unlock pattern or a single/multi-finger touch gesture on the touchscreen of his smartphone which then records the touch and MFV data. In the subsequent verification phase, the user inputs his unlock pattern or touch gesture in the same way as in the enrollment phase. The smartphone compares the resulting finger movement data generated by the inertial magnetometer and touch sensor with the stored and admits the user if a strong match can be found. In what follows, we detail the enrollment and verification phases.

3.1 Enrollment Phase

The user opens the MagAuth app and presses a soft or hard button on the device to start performing an unlock pattern or a touch gesture. How to select a suitable unlock pattern or touch gesture is deferred to Section 3.3. The resulting raw touch and MFV data go through the following *Data Preprocessing* and *Feature Extraction* modules in sequence.

3.1.1 Preprocessing

After the user performs a chosen unlock pattern or touch gesture, the inertial touch sensor and magnetometer on the smartphone simultaneously generate a touch-data sequence and a raw MFV data sequence, respectively. We first subtract the environmental MFV from the raw MFV data to obtain the net magnet-induced MFV data.

Environmental magnetic field cancellation. Since there exists long-lasting and stable magnetic field around us such as Geomagnetic field and the fields induced by mental infrastructures, we prerecord the average strength of environmental MFV which is assumed to be non-changing during the short enrollment phase. Then using the attitude sensor in the smartphone, we derive the values of the environmental magnetic field strength along the three axes in the magnetometer's coordinate system. The attitude sensor in most smartphones has its own fixed reference coordinate system (e.g., a reference frame with the X-axis pointing towards geomagnetic North and the Z-axis pointing upward away from the geocenter). Therefore, we first measure the environmental magnetic field distribution in this reference system $(x_{refmag}, y_{refmag}, z_{refmag})$ by using one magnetometer and aligning it with the reference system in advance or directly using the magnetometer and the attitude sensor in the smartphne before authentication. During the authentication process, we can get the attitude sensor readings of the smartphone and derive the rotation matrix R. By space transformation in Eq. (11), the environmental magnetic field readings along the three axes of the magnetometer space $(x_{mag}, y_{mag}, z_{mag})$ could be obtained as follows,

$$\begin{bmatrix} x_{mag} \\ y_{mag} \\ z_{mag} \end{bmatrix} = R \begin{bmatrix} x_{refmag} \\ y_{refmag} \\ z_{refmag} \end{bmatrix}. \quad (11)$$

In most real-world scenarios, we just need to consider the impact of the stable environmental magnetic field such as the Geomagnetic field and the field induced by large equipment. Most moving magnetic objects can hardly be sensed due to the exponential attenuation of the magnetic field strength with the distance between the magnet and the magnetometer. Fig. 22 and Fig. 23 in Section 4.3.1 also verify this claim. As a result, we can use the environmental magnetic field cancellation method mentioned above to cancel the stable environmental magnetic impact. In addition, if there appears instantaneous electromagnetic interference

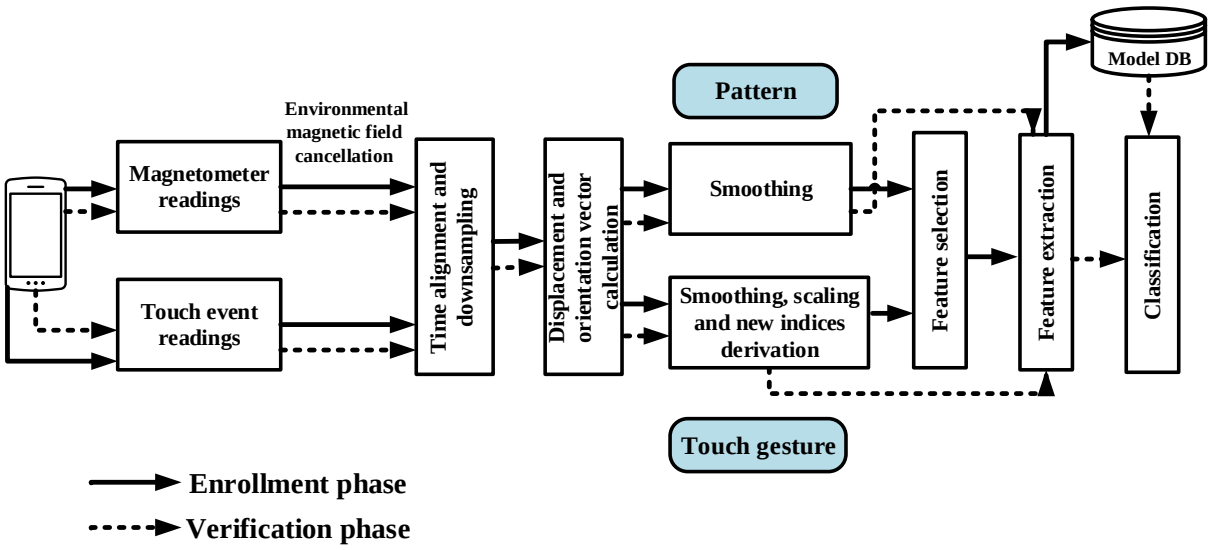


Fig. 5: MagAuth system workflow.

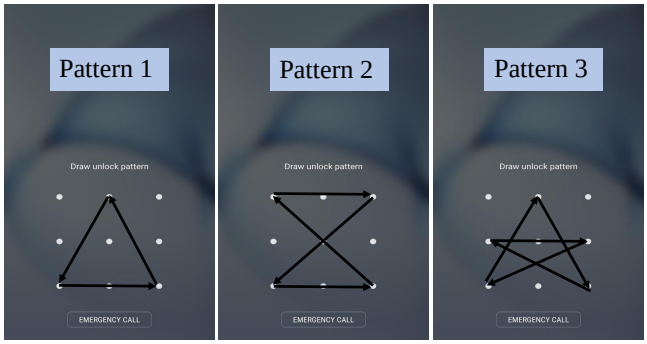


Fig. 6: Illustration of three unlock patterns.

during the authentication process, we can remove the outliers from the signals before the final decision.

To increase the distinction of adjacent sampling points to facilitate solving the equation set in 2.3, we downsample the collected data sequence from 100 Hz to 50 Hz empirically. After time alignment, we obtain the pair of MFV and touch data sequences. Given the prerecorded values of $\langle r_x^{touch}, r_y^{touch}, r_z^{touch} \rangle$ and $\langle r_x^{origin}, r_y^{origin}, r_z^{origin} \rangle$, we can derive the displacement and orientation vectors of finger movement. Due to inevitable hand vibrations during the inputting process, the values of displacement and orientation vectors can have some shaking noise. Therefore, we use the 3-point moving average method to smooth the curves.

The derived displacement and orientation vectors of Pattern 2 in Fig. 6 are shown in Fig. 7 and Fig. 8, respectively. As we can see, the X-axis and Y-axis curves of orientations and displacements from the same person present similar trajectories, while the Z-axis curves fluctuate within a small range and have no clear distinction because of the dominating hand-shaking effect. Therefore, we just process the X-axis and Y-axis data of orientation and displacement vectors in what follows.

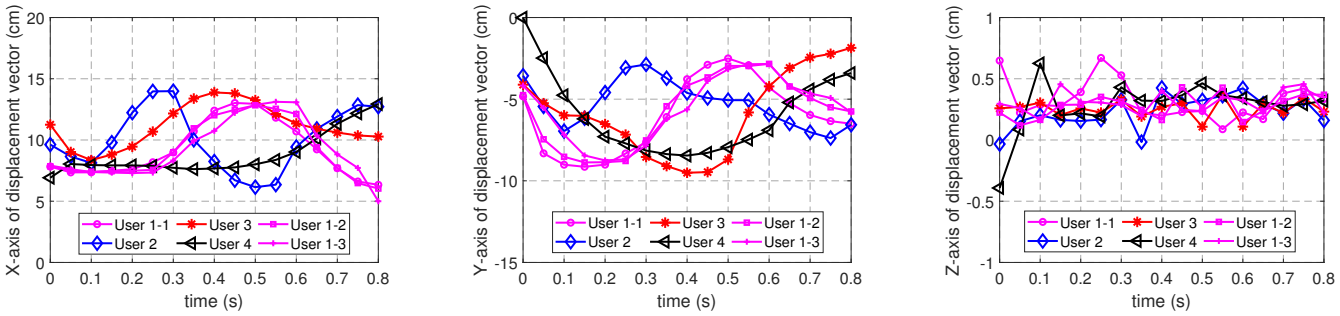
As for touch gestures, the absolute displacements and orientations of finger movement always change for each trial due to the varying gesture location, scale, and rotation. As described in [26], the impact of gesture rotation could be neglected because of a user's input habits. Therefore, we first scale original samples into a bounding box (1*1) to mitigate the impact of different gesture scales in each trial. Then we apply the following three indexes [27] to both displacement and orientation vectors to represent finger movement, which are not impacted by the touch gesture's starting position on the touchscreen.

- **Distance**: the distance between sequential sample points in the X-Y plane, denoted by $d_{xy}(i) = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$.
- **Path angle**: the angle between sequential sample points in the X-Y plane, denoted by $alpha_{xy}(i) = \arccos \frac{\vec{p}(t) \cdot \vec{p}(t+1)}{||\vec{p}(t)|| * ||\vec{p}(t+1)||}$, where $p(i) = [x_{i-1} - x_i; y_{i-1} - y_i]$.
  Curvature: the log radius of curvature in the X-Y plane, denoted by $k_{xy}(i) = \frac{|v_{x,i} * acce_{y,i} - v_{y,i} * acce_{x,i}|}{(v_{x,i}^2 + v_{y,i}^2)^(3/2)}, log_{k_{xy}(i)} = log(1/k_{xy}(i))$.

As shown in Fig. 9 and Fig. 10, the curves from the same user present similar trends while those from different users are significantly distinct for all the three indices. Therefore, instead of the orientation and displacement vectors, we use the derived distance, path angle, and curvature to distinguish touch gestures among different users.

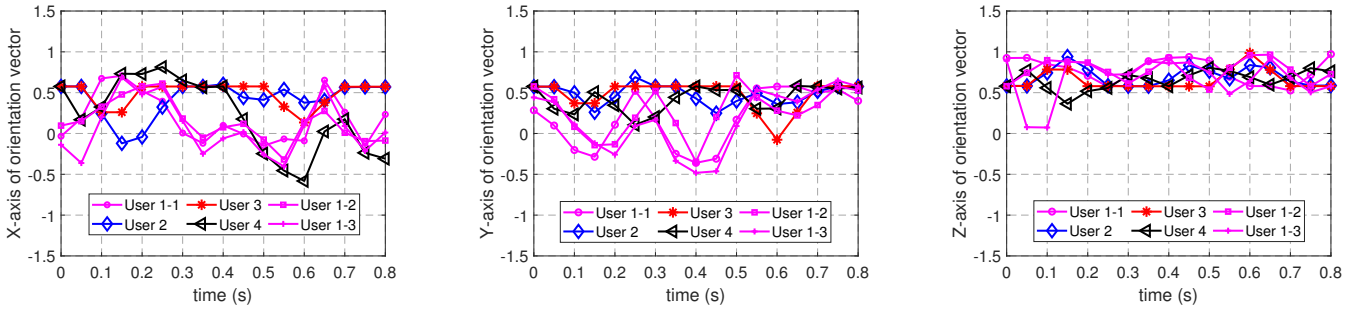
3.1.2 Feature Extraction

In this section, we take the X-axis and Y-axis sequences of the displacement vector of the unlock pattern as an example to illustrate the feature extraction process in MagAuth. The orientation vector and the three indices of the touch gesture are processed in the similar way. We denote the displacement vector by $\{t_i, x_i, y_i\}_{i=1}^N$, where N is the number of sampling points. For the X-axis data series denoted by $V_x := \{x_i\}_{i=1}^N$, we represent its average by μ_x and its variance by σ_x. Similarly, we define other variables



(a) X-axis of displacement vector (b) Y-axis of displacement vector (c) Z-axis of displacement vector

Fig. 7: Displacement vectors of finger movement for Pattern 2 in Fig. 6. As for the X-axis and Y-axis of the displacement vector, the samples from the same user have similar readings while those from distinct users are significantly different.



(a) X-axis of orientation vector (b) Y-axis of orientation vector (c) Z-axis of orientation vector

Fig. 8: Orientation vectors of finger movement for Pattern 2 in Fig. 6. As for the X-axis and Y-axis of the orientation vector, the samples from the same user have similar curves while those from distinct users are different.

TABLE 2: List of initialized features

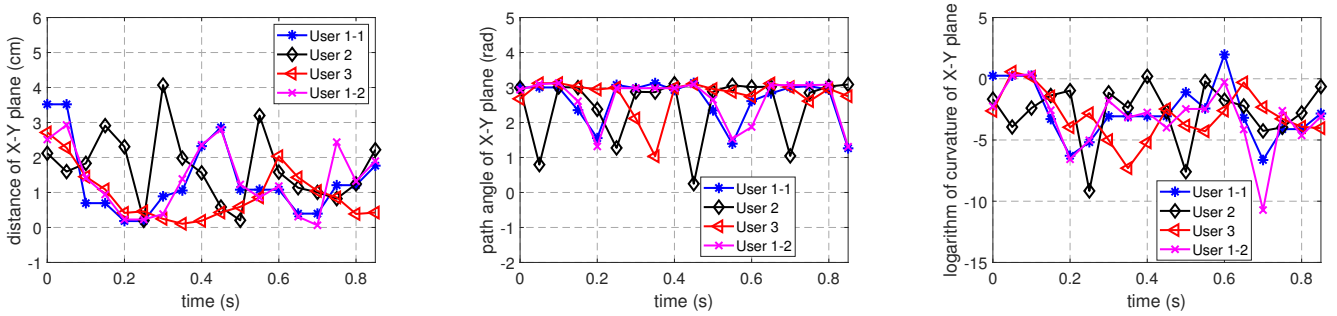
feature name	Description
Coefficient of Variation	$cv_x = std(x)/\bar{x}$
Fast Fourier Transform	$\{FFT_{x,i}\}_{i=1}^{M_x}$
Energy	$energy_x = \frac{\sum_{i=1}^{N} \|x_i\|^2}{N}$
Zero Crossing Rate	$zcr_x = \frac{1}{N-1}\sum_{i=1}^{N-1} \mathbf{1}_{x_i x_{i-1}<0}$
Interquartile range	$iqr_x = Q_3(x) - Q_1(x)$
Skewness	$\frac{\frac{1}{N}\sum_{i=1}^{N}(x_i-\mu_x)^3}{\left[\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\mu_x)^2\right]^{3/2}}$
Kurtosis	$\frac{\frac{1}{N}\sum_{i=1}^{N}(x_i-\mu_x)^4}{(\frac{1}{N}\sum_{i=1}^{N}(x_i-\mu_x)^2)^2} - 3$
Pearson Correlation Coefficient	$\frac{\mathbf{cov}(V_x,V_y)}{\sigma_x \sigma_y}$
Velocity	$\{v_{x,i} = \frac{x_{i+1}-x_i}{t_{i+1}-t_i}\}_{i=1}^{N-1}$
Acceleration	$\{acce_{x,i} = \frac{v_{x,i+1}-v_{x,i}}{t_{i+1}-t_i}\}_{i=1}^{N-1}$
Hu Invariant Moments	$M_i(\mu_{j,k}), i = 1,2,...,7,$ $\mu_{j,k} = \sum_x \sum_y (x-\bar{x})^j (y-\bar{y})^k,$ $j,k \in \{0,1,2,3\}$

including V_y, μ_y, and σ_y. We initialize 19 features in Table 2 to characterize the MFV data.

In the next step, we convert each feature vector into a scalar using the root-mean-square (RMS) metric. Consider the FFT feature vector $\{FFT_{x,i}\}_{i=1}^{M_x}$ for the X-axis as an example. Let $\{\overline{FFT_{x,i}}\}_{i=1}^{M_x}$ denote the averages of multiple legitimate training samples for the same gesture. The RMS value of $\{FFT_{x,i}\}_{i=1}^{M_x}$ is computed as $\sqrt{\frac{1}{M_x}\sum_{j=1}^{M_x}(FFT_{x,j} - \overline{FFT_{x,j}})^2}$. Finally, we obtain a scalar vector of 18 RMS values, one for each feature.

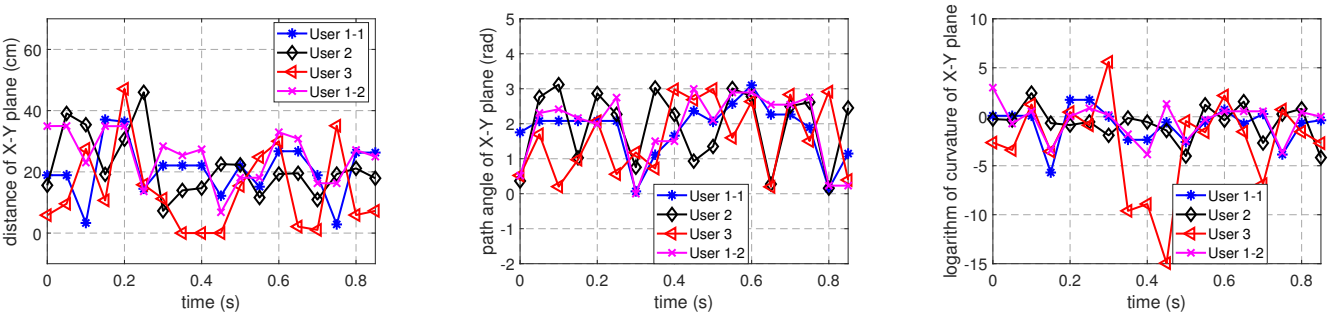
Feature Selection. We notice that not all features are necessary for each unlock pattern or touch gesture. Therefore, after scalarization, all the 19 features are fed into a feature-selection module which trains a neighborhood component analysis (NCA) model to learn and find the most powerful feature combination. Compared with PCA which focuses on dimensionality reduction, NCA belongs to supervised learning and aims to find the best prediction results. Here we use 5-fold cross-validation for model training. The classification accuracy before and after feature selection is shown in Table 3. In the table, we use P1, P2, P3, G1, G2, G3 to represent Pattern 1, 2, 3, and Gesture 1, 2, 3. As we can see, feature selection significantly improves the classification accuracy for both the pattern-lock and touch-gesture mode.

Then we use the resulted scalar vector to train a classifier based on sophisticated machine learning algorithms, for which we experimentally compare the performance of some popular ones in Section 4. For this purpose, the user needs to perform the same unlock pattern or touch gesture multiple times, each leading to a legitimate scalar vector. MagAuth also maintains a library of random scalar vectors that serve as illegitimate training samples for the user. The accuracy of the classifier can be improved and dynamically retrained as the user supplies more legitimate samples in subsequent authentication phases. Since the classifier is infrequently updated, it can be trained in the cloud and pushed to the smartphone which may have limited computational capabilities.



(a) distance of displacement in X-Y plane (b) path angle of displacement in X-Y plane (c) curvature of displacement in X-Y plane

Fig. 9: Indices (distance, path angle, and curvature) of the moving finger's displacement in X-Y plane when inputting the touch gesture similar to Pattern 2 in Fig. 6. Samples from the same user have the similar curves while that from different users are totally different.



(a) distance of orientation in X-Y plane (b) path angle of orientation in X-Y plane (c) curvature of orientation in X-Y plane

Fig. 10: Indices (distance, path angle, and curvature) of the moving finger's orientation in X-Y plane when inputting the touch gesture similar to Pattern 2 in Fig. 6. Samples from the same user have the similar curves while that from different users are totally different.

3.2 Verification Phase

Assume that a user wants to unlock his device. He wears the enrolled magnetic wearable to perform his personal unlock pattern or touch gesture, resulting in a swipe event and MFV sequences recorded by MagAuth. After going through the same data processing in Section 3.1, the data sequences are converted into a scalar vector which is then fed into the trained classifier. Because feature selection has been done during the enrollment phase, it is omitted in this phase. The user passes the authentication only when both conditions are met: (1) the difference between the derived value of finger movement and the recorded one is below a system threshold, and (2) the pattern/gesture-induced scalar vector is classified as legitimate. The system password mechanism is invoked as a fall-back mechanism if the user fails to pass MagAuth after a threshold number of authentication attempts. Fig. 5 demonstrates the whole authentication process for clarity.

3.3 Selection of Unlock Patterns or Touch Gestures

What kind of unlock patterns or touch gestures should the user pick? As analyzed before, we want to get the vector $\text{Index} = \langle m_x, m_y, m_z, \Delta x, \Delta y, \Delta z \rangle$. where (m_x, m_y, m_z) characterizes the orientation change of the wrist wearable's magnetic clasp, and $(\Delta x, \Delta y, \Delta z)$ characterizes the relative finger-clasp position changes and relates to the user's unlock pattern/gesture. Since our goal is to make Index clearly distinguishable among different users, the best method is to increase the fluctuation of Index.

We observe that the relative magnetometer-wrist position (r_x, r_y, r_z) is more stable than the finger movement (r_x^d, r_y^d, r_z^d) due to most users' input habits. In particular, we have the following result

$$r_x = r_x^d \uparrow + \Delta x \downarrow; r_y = r_y^d \uparrow + \Delta y \downarrow; r_z = r_z^d \uparrow + \Delta z \downarrow . \tag{12}$$

From Eq. (12), when (r_x, r_y, r_z) is stable, the unlock-pattern or touch-gesture curves (r_x^d, r_y^d, r_z^d) should be as complex as possible to achieve noticeable fluctuation $(\Delta x, \Delta y, \Delta z)$.

Armed with the above observation, we first discuss the selection of unlock patterns. In [28], the authors define the (security) strength of an arbitrary unlock pattern P as

$$PS_P = S_P \times log_2(L_P + I_P + O_P), \tag{13}$$

where PS_P denotes the pattern strength. S_P, L_P, I_P, and O_P are the size, length, the number of intersections, and the number of overlaps of the pattern P. It is clear that larger values of S_P, L_P, I_P, and O_P result in more complex pattern curves and thus more elaborate curves of $(\Delta x, \Delta y, \Delta z)$. In Section 4, we evaluate three patterns in Fig. 6 which correspond to the complex ($PS_P \geq 33$), medium ($19 \leq PS_P < 33$), and simple ($PS_P < 19$) level

of pattern strength, respectively and provide the guidelines on unlock-pattern selection.

In the touch-gesture mode, the touch gesture should be easy for the user to remember and reproduce (high usability) but difficult for the attacker to emulate (high security). The authors in [29] first discretize one touch gesture, then use the similar representation of PIN numbers with the n-gram Markov Model, and finally, refer to the partial guessing entropy estimation [30]. If the discretized gesture is complex, we could claim that the original touch gesture is also complex. MagAuth follows a white list of good touch gestures in [29] which achieve the usability and security at the same time properly. Note that although there is no fixed input area for a touch gesture, the size of the gesture could not be too large. Otherwise, the value of (r_x^d, r_y^d, r_z^d) may dominate that of (r_x, r_y, r_z) and thus cause low discrimination of $(\Delta x, \Delta y, \Delta z)$.

3.4 Application Scenarios

The great potential of MagAuth is based on the rapid growth of wrist wearables. In particular, wrist wearables, including smartwatches, basic watches, and wrist bands, reached 34.2 million units in the second quarter of 2019; the global fitness trackers are predicted to reach 59.22 billion by 2023. Almost all wrist wearables except basic watches are required to be paired with a host device, and many people are used to carrying their paired smartphones along with their wrist wearables devices. Other magnetic wearables such as rings and bracelets are also very common in daily life.

We picture three possible application scenarios of MagAuth, as shown in Fig. 11. The first is that the user holds the smartphone while trying to unlock it with the same hand wearing the wrist wearable. For example, the user wants to unlock his smartphone while walking. The second is that the user holds the smartphone with one hand and tries to unlock it with the other hand wearing the wrist wearable. The third scenario is that the user tries to unlock the smartphone placed on a flat surface. In this paper, we compare the system performance under these three scenarios in the experiment part.

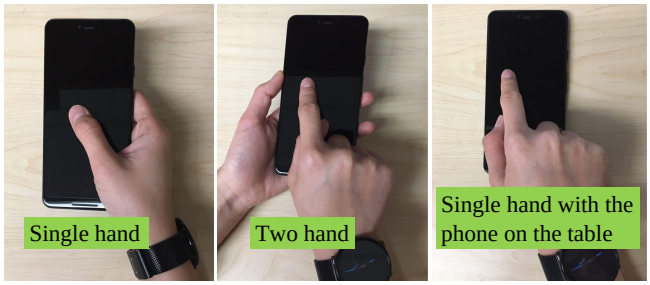


Fig. 11: Application scenarios.

3.5 MagAuth with Other Password Authentication Modes

In addition to unlock patterns and touch gestures, MagAuth can work with PINs and alphanumeric passwords as well. Take PINs as an example. In the Android system, a user can make his PIN as long as 17 digits. Assume that the user chooses a 17-digit PIN for MagAuth. According to the remarks in Section 2, when the user inputs his PIN, MagAuth generates six time sequences, i.e., $\langle m_x, m_y, m_z, \Delta x, \Delta y, \Delta z \rangle$. Each sequence contains either 8 or 5 sampling points which correspond to Remark 2 and 3, respectively. Then four sequences (i.e., $\langle m_x, m_y, \Delta x, \Delta y \rangle$) are used as the user's behavioral features. In total, we use 32 or 20 sampling points for authentication, which are far less than hundreds of points for unlock patterns or touch gestures. But we could easily augment the data set for authentication according to the continuity of the inputting process. In contrast, for a 4- or 6-digit PIN, since there are only less than 10 sampling points (sometimes even less than 5 points) generated to represent one user's behavioral features, the authentication performance might be degraded significantly. So we recommend users to apply MagAuth to longer PINs. The extension of MagAuth to PINs and alphanumeric passwords is left as future work.

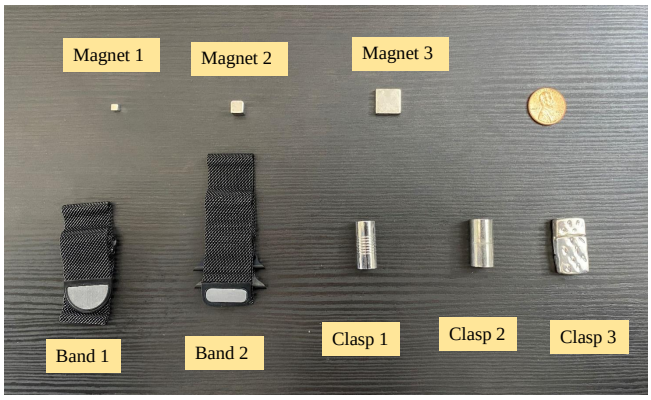


Fig. 12: Different magnets and magnet bands/clasps.

TABLE 3: Classification accuracy before and after feature selection

accuracy	P1	P2	P3	G1	G2	G3
before	88.5%	89.3%	91.7%	86.4%	88.2%	90.5%
after	95.8%	98.6%	98.1%	94.6%	96.2%	98.0%

3.6 Adversary Models

In this section, we consider the adversary models against MagAuth. The attacker $\mathcal{A}$ possesses and attempts to log into the locked smartphone which has been turned on by the legitimate user when it is stolen or lost. $\mathcal{A}$ knows that MagAuth is installed and can be used to unlock the smartphone. $\mathcal{A}$ is also fully aware of how MagAuth is designed and works. In particular, $\mathcal{A}$ knows that the user can unlock the smartphone by performing the self-chosen unlock pattern or touch gesture with a magnetic wrist wearable. If the user chooses the authentication scheme demonstrated in Remark 3 of Section 2.3, the knowledge of the band/clasp is not necessary because attackers could unlock the device with a random magnetic band/clasp. If the user chooses the scheme in Remark 2 instead, attackers need to obtain the same band/clasp to achieve their goals. We assume that the

attacker can obtain the same band/clasp used by the user for authentication. Below, we classify the attacker into one of the following types according to his increasing capability.

- Type-I (guess): $\mathcal{A}$ does not know the user's unlock pattern or touch gesture; $\mathcal{A}$ knows neither how the user performs the unlock pattern or touch gesture (i.e., the pattern/gesture trajectory) nor the details of his finger movement (i.e., behavioral features) during the authentication process.
- Type-II (replay): $\mathcal{A}$ has a rough idea about the user's unlock pattern or touch gesture; he knows neither the pattern/gesture trajectory nor the behavioral features of the user.
- Type-III (one-time-observation mimicry): $\mathcal{A}$ knows the user's unlock pattern or touch gesture; he knows the pattern/gesture trajectory but not the behavioral metrics.
- Type-IV (five-time-observation mimicry): $\mathcal{A}$ knows the user's unlock pattern or touch gesture; he knows both the pattern/gesture trajectory and the behavioral metrics of the user.

$\mathcal{A}$ can observe the user's unlock pattern or touch gesture via shoulder surfing and/or stealthy video recording via a spy camera. The resilience of MagAuth to the above attacker types is experimentally evaluated in Section 4.

4 Performance Evaluation

In this section, we experimentally evaluate the security and usability of MagAuth on a Samsung Galaxy S5 which has the magnetometer on its upper-right corner of the front face.

4.1 Experimental Setup

We designed experiments to verify that legitimate users could be distinguished by their behavioral features when inputting an unlock pattern or a touch gesture with a magnetic wearable. To eliminate the effect of different magnetic bands, we used the same one [31] in all the experiments. The phone was placed on a table with the front face up (as shown in Fig. 11), unless otherwise stated.

According to the analysis in Section 3, elaborate unlock patterns or touch gestures can produce more proper MFV data suitable for user authentication. We chose three simple patterns: Pattern 1, Pattern 2, and Pattern 3 shown in Fig. 6. In addition, the touch gestures we chose are similar to the three unlock patterns except that they were performed at arbitrary starting positions on the touch screen.

We recruited 15 volunteers, all of whom are college students aged 18 or above. All volunteers acted as legitimate users, and 10 among them also served as attackers who tried to mimic the unlock patterns/gestures of other users. Every legitimate user input each of the three unlock patterns/gestures 20 times, so we collected 1,800 legitimate samples. In order to avoid the bias caused by other restrictions, the whole experiment process had no time limitation, and participants were encouraged to perform unlock patterns/gestures in a natural way as they normally do. Our experiments were approved by our Institutional Review Board (IRB).

We also conducted experiments to evaluate the performance of MagAuth under attacks. Section 3.6 lists three types of attackers against MagAuth. Due to space limitations, we only report the resilience to more capable Type-II, Type-III, and Type-IV attackers, which corresponds to the performance of MagAuth in the worst-case scenarios. In particular, since the same magnetic band was used, each participant can be considered a Type-II attacker against all other participants. To emulate Type-III and Type-IV attackers, we first video-recorded the pattern/gesture-performing process of each legitimate user. We asked every attacker to pick six videos consisting of different patterns/gestures performed by different users. Note that once a video was chosen, it cannot be chosen again by other attackers. The attacker mimicked the patterns/gestures in the chosen videos one by one to unlock the device. The attackers were told that both the orientation and displacement of finger movement may affect the MFV data. In the first scenario corresponding to Type-III attackers, every attacker watched the video of a chosen pattern/gesture once. In the second scenario corresponding to Type-IV attackers, every attacker watched the video of a chosen pattern/gesture five times and was told to pay close attention to the finger movement. In both scenarios, almost all attackers could remember the detail of every pattern/gesture. Every attacker mimicked each observed pattern/gesture five times. Therefore, each kind of patterns/gestures was attacked 10*5 times.

We recruited another 8 volunteers, who are college students aged 18 or above, to evaluate the authentication performance with different band tightness, swiping speed, application scenarios, and environments. We also compared the authentication performance of various bands/clasps and smartphones.

We tested the performance of MagAuth with popular classifiers including SVM, Naive Bayes, and Random Forest (RF), which led to comparable results. Because RF slightly outperform other methods, we only report the results with RF except for the algorithm comparison.

4.2 Performance Metrics

We use the receiver-operating-characteristic (ROC) and precision-recall curves as the main performance metrics.

An ROC curve illustrates the performance of a binary classifier as its discrimination threshold changes. The Area Under the Curve (AUC) is an indicator of the overall quality of an ROC curve. For example, the ROC of the ideal classifier has AUC equal to 1. We can plot an ROC curve by plotting the true positive rate (TPR) with respect to the false positive rate (FPR) for different thresholds. Denote the number of true positives, false positives, true negatives, and false negatives by $\#\mathtt{TP}, \#\mathtt{FP}, \#\mathtt{TN}$, and $\#\mathtt{FN}$. Then TPR and FPR can be calculated as

$$\mathtt{TPR} = \frac{\#\mathtt{TP}}{\#\mathtt{TP} + \#\mathtt{FN}} \text{ and } \mathtt{FPR} = \frac{\#\mathtt{FP}}{\#\mathtt{FP} + \#\mathtt{TN}}. \quad (14)$$

Another indicator is the Equal Error Rate (EER), the point on the ROC curve that corresponds to an equal probability of miss-classifying a positive or negative sample (i.e., FPR=FNR). The lower the EER, the more robust the system, the better the classification performance, and vice versa.

The precision-recall curve shows the trade-off between precision and recall for different thresholds. Precision represents the percentage of legitimate users out of all admitted users for an authentication system, which is calculated as

$$\text{Precision} = \frac{\#\texttt{TP}}{\#\texttt{TP} + \#\texttt{FP}} \text{ and} \tag{15}$$

$$\text{Accuracy} = \frac{\#\texttt{TP} + \#\texttt{TN}}{\#\texttt{TP} + \#\texttt{FP} + \#\texttt{FN} + \#\texttt{TN}} \tag{16}$$

Recall in authentication systems is the same as TPR which measures the proportion of legitimate users correctly identified as such. A high area under the precision-recall curve represents both high recall and high precision, where high precision relates to a low FPR, and high recall relates to a low FNR. The classifier is claimed to have high performance if both high precision and high recall can be obtained. Besides, the recognition accuracy in this paper is defined as the proportion of correctly classified samples including both positive and negative samples. High accuracy means a good tradeoff between robustness and security.

We also measured the execution time of MagAuth, which relates to its usability and should be as short as possible.

4.3 Results

4.3.1 *Performance with Type-II Attackers*

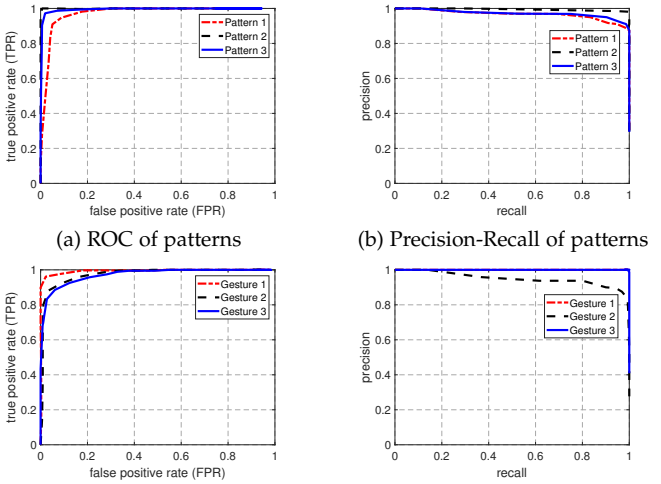
Recall that each of the 15 legitimate users was required to input each of six unlock patterns/gestures 20 times, producing 1,800 magnetic gesture samples. We randomly chose 10 samples from each legitimate user and 140 samples from other users for classifier training. The remaining samples were used for testing, and the samples of all other users are equivalent to Type-II attackers' inputs against every legitimate user. The process was repeated 20 times, and the average results are reported below.

We tested the performance of MagAuth with popular classifiers including SVM, Naive Bayes (NB), and Random Forest (RF). RF and NB led to comparable results. Table 4 compares the recognition accuracy of all the pattern/gestures based on these two classification algorithms. It's shown that the accuracy of every pattern/gesture is above 90% which verifies the security of our system. Moreover, among these patterns/gestures, Pattern 2 and Gesture 3 have the highest average accuracy because of the good trade-off of pattern/gesture complexity and coverage area. In the meanwhile, Pattern 1 and Gesture 1 have the lowest accuracy due to small covering area or simple shape. Because in most scenarios, the performance of RF are a little better than NB, we choose Random Forest as the classification algorithm in the following experimental part.

TABLE 4: Classification accuracy of Random Forest (RF) and Naive Bayes (NB)

accuracy	P1	P2	P3	G1	G2	G3
RF	95.8%	98.6%	98.1%	94.6%	96.2%	98.0%
NB	94.5%	97.3%	96.7%	96.4%	97.2%	97.5%

Fig. 13 shows the results for both the unlock-pattern and touch-gesture modes. According to Fig. 13a, the ROC curves for all three unlock patterns/gestures are located in the top-left corner, so MagAuth can achieve high TPR and low FPR at the same time. Similarly, the Precision-Recall curves in Fig. 13b are all located in the top-right corner and show that MagAuth could simultaneously achieve high precision and high recall.



(a) ROC of patterns

(b) Precision-Recall of patterns

(c) ROC of touch gestures

(d) Precision-Recall of touch gestures

Fig. 13: ROC and Precision-Recall curves of three patterns and gestures.

The three unlock patterns have slightly different performance. In particular, the highest TPR is 96.3%, which was achieved with Pattern 2; the highest FPR is 5.5%, which was achieved with Pattern 1. In contrast to Pattern 2 and Pattern 3, Pattern 1 led to slightly worse performance in both the ROC and Precision-Recall curves. One reason is although Pattern 1 is more complex, the coverage area is not that large for elastic finger bending which results in smaller discrimination among different users. Therefore, when a user chooses an unlock pattern/gesture for authentication, he should consider both the complexity and covering area of the pattern. Another reason is that positive samples are relatively easier to be classified as negative because the testing dataset has many more negative samples than positive ones.

The ROC curves and Precision-Recall curves for all three touch gestures are shown in Fig. 13c and Fig. 13d, respectively. Gesture 1 has better performance than both Gesture 2 and Gesture 3 with TPR value 95.4%. Gesture 3 has the highest FPR value 8.4% One reason is Gesture 2 and Gesture 3 are both simpler than Gesture 1, resulting in smaller discrimination among different users. Another reason is Gesture 2 and Gesture-3 have worse repeatability than Gesture 1.

Fig. 14 further verifies the classification performance. The EER values of all unlock patterns/gestures are below 0.1, which verifies the feasibility of MagAuth. In addition, Pattern 1 and Gesture 3 have worse performance than other patterns/gestures, which is consistent with the result shown in Fig. 13. In addition, based on the CDF of recognition

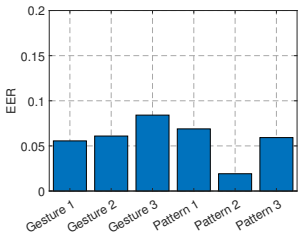


Fig. 14: EER of different patterns/gestures.

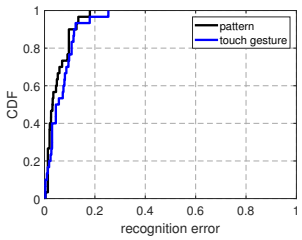
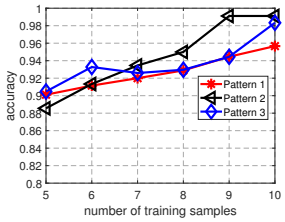


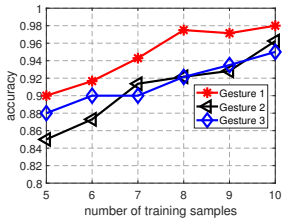
Fig. 15: CDF of recognition errors of patterns and gestures.

errors in Fig. 15, most users have more than 90% recognition accuracy. Unlock patterns slightly outperform touch gestures because of more stable indexes fed into the classifier for the former.

Fig. 16 demonstrates the classification accuracy—the sum of true positives and negatives over all test samples—as the number of legitimate samples varies. It is clear that MagAuth has very high accuracy for all six patterns/gestures, which increases with the number of legitimate samples. In particular, when the number of legitimate samples for unlock patterns and touch gestures reach 6 and 7, respectively, the classification in both modes is above 90%. In other words, the classification accuracy can be boosted if the legitimate user supplies more legitimate samples in the enrollment phase and/or later successful authentication instances.



(a) accuracy of patterns



(b) accuracy of touch gestures

Fig. 16: Classification accuracy vs. the number of legitimate samples.

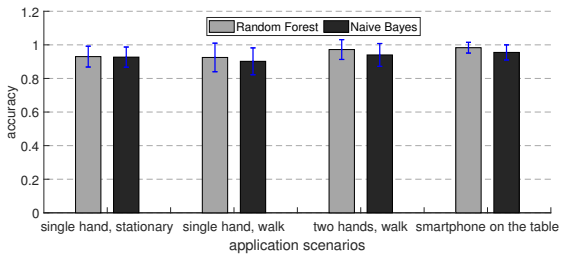


Fig. 17: Accuracy under different application scenarios

Fig. 17 demonstrates the classification accuracy under different application scenarios in Fig. 11. For the stationary scenario with a single hand, we asked the volunteers to stand still during the authentication process. Here we show the results of Pattern 2 mentioned above. As we can see, the authentication accuracy is above 90% with acceptable standard deviation for all the four scenarios. In particular, consistent with our intuition, the accuracy reaches the 98.5% with Random Forest when users input their patterns with the smartphone on the table. Besides, under the two-hand scenario and the scenario with the smartphone on the table, MagAuth achieves similar high accuracy with both algorithms. In addition, the accuracy in the two-hand input mode is higher than that in the single-hand mode. One possible reason is that the input actions cause the vibration of the device in the single-hand mode. The other reason may be that the wrist movement range in the single-hand mode is smaller than that in the two-hand mode, which may impact the feature extraction. The walking scenario under the single-hand mode achieves the accuracy of 92%, which is slightly lower than that under the stationary scenario due to the vibration noise and the time-delay of the attitude sensor during the posture estimation of the smartphone. We also evaluated MagAuth in three environments and show the results in Fig. 18. Note that we required that the users input Pattern 2 with the two-hand mode in Fig. 11 and conducted the experiments under the stationary scenario. The results show that MagAuth can achieve the average accuracy of 91.4%, 93.0%, and 98.1% in the lab, car, and home environments, respectively. In particular, the accuracy in the lab and car environments are lower than that at home due to the occasionally appeared, unstable, and strong magnetic field disturbance caused by the magnetic equipment in the lab or the vibration noise in the car.

Table 5 demonstrates the classification accuracy with different swiping speeds and band tightness. We let users input Pattern 2 in slow, normal, and fast modes, corresponding to about 0.5, 1.2, and 2.5 seconds, respectively. As we can see, the accuracy drops when the user inputs the pattern with the fast mode because with the sampling rate constraint of 100 Hz, the sampling points after preprocessing in the fast mode may not represent the behavior characteristics sufficiently. The good news is that more and more smartphones begin to use IMU sensors with the sampling rate up to 400 Hz which is enough for MagAuth. In addition, we evaluated the impact of band tightness in three ways: the band can rotate freely around the wrist (loose), l slightly move around (normal), and hardly move around (tight). It is of no surprise to see that MagAuth is very accurate in normal and tight modes in contrast to the loose mode.

TABLE 5: classification accuracy of MagAuth under different impact factors

band tightness	loose	normal	tight
accuracy	76.2%	97.3%	97.9%
swiping speed	slow	normal	fast
accuracy	98.1%	97.5%	88.6%

Fig. 19, Fig. 20, and Table 6 demonstrate MagAuth's performance for various smart smartphones and magnetic bands/clasps illustrated in Fig. 12. First, we randomly selected one user who has a medium-size palm and let him input Pattern 2 twice. We compare the MFS values of three COTS magnetic bands/clasps in Fig. 20. As we

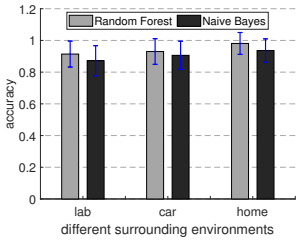


Fig. 18: Accuracy under different environments

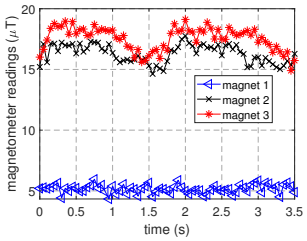


Fig. 19: MFS of various magnets

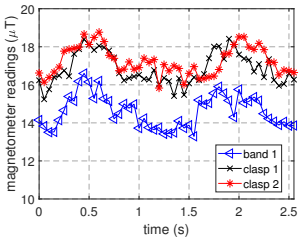


Fig. 20: MFS of various magnetic bands/clasps

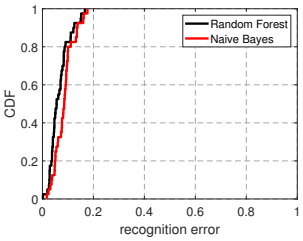


Fig. 21: CDF of recognition errors in continuous days.

can see, all three bands/clasps have enough magnetism to represent users' behaviors. Moreover, for the same user, the MFS curves are also similar, so we can use any proper magnetic bands/clasps for authentication when the method in Remark 3 of Section 2 is applied to MagAuth. However, not all magnetic bands/clasps are proper for authentication. We conducted experiments on different magnets to provide guidelines on future band/clasp selection for our goal. Fig. 19 shows the overall magnetic field strength (MFS) of three magnets which are NdFeB and Grade N40 with sizes of 3.2mm*3.2mm*3.2mm, 6.4mm*6.4mm*6.4mm, and 3.2mm*12.7mm*12.7mm, respectively. As we can see, all but Magnet 1 induce significant changes in the magnetic field during pattern input. The results show that only if the tiny magnet in the band/clasp has magnetism similar to or stronger than that of Magnet 2 whose size is about half of a coin, it could be used for user authentication. In Table 6, we further compare the classification accuracy of various magnets, bands/clasps and smartphones. Magnet 2 and 3 have better performance than Magnet 1, and all the three magnetic bands/clasps achieve high accuracy. In addition, all the three smartphones—Samsung S5, Samsung S7, and LG Nexus 5–achieve good classification accuracy. The accuracy for Samsung S5 and Nexus 5 are a little higher than that for Samsung S7. The possible reasons are the magnetometer of Samsung S5 is on the upper-right corner of the phone front which is close to the inputting hand, and Nexus 5 is relatively small and easier to collect more elaborate features of the wrist movement. These results further verify the usability of MagAuth.

TABLE 6: Accuracy vs devices/bands

magnet	block 1	block 2	block 3
accuracy	66.3%	95.5%	98.8%
band/clasp	band 1	clasp 1	clasp 2
accuracy	97.4%	97.6%	98.1%
smartphone	Samsung S5	Samsung S7	LG Nexus 5
accuracy	97.9%	95.2%	97.7%

In Fig. 21, we show the classification accuracy based on the data collected in one week continuously. More than 90% of users have recognition errors smaller than 10% , which means that most users could input the pattern consistently in a relatively long period. This result further verifies the usability of MagAuth.

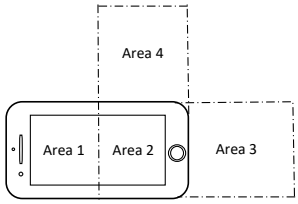


Fig. 22: Illustrations of movement areas of the hand (horizontal)

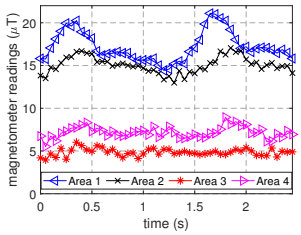


Fig. 23: MFS with various distances between the magnetic clasp and the magnetometer

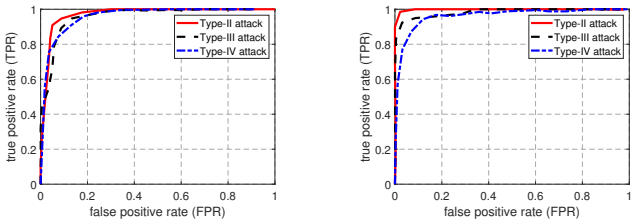
We also demonstrate the impact of the distance between the magnetic clasp and the magnetometer. As shown in Fig. 22, we split the sensing space into four areas and let one user input Gesture 2 in these areas. Obviously, Area 1 is the closest place to the magnetometer, and Area 3 and 4 are a little farther. Fig. 23 shows the MFS changes in these areas. As we can see, MFS has the most dramatic change which benefits user classification and authentication in Area 1. In Area 2 where the pattern keypad is always located, the change is also large enough for authentication. In contrast, the MFS curves in Area 3 and 4 have no obvious fluctuation which means that most magnetic objects around the device have little impact on the authentication process.

4.3.2 *Performance with Type-III and Type-IV attackers*

In this group of experiments, we added false samples for each legitimate user into the testing set, which were generated by the attackers aiming at the user. Therefore, the testing set of a pattern/gesture from each user consists of 10 legitimate samples and 5 fake samples.

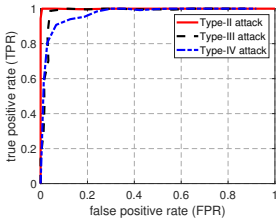
Fig. 24 shows the ROC curves with Type-III and Type-IV attackers. It is clear that the unlock patterns/gestures are highly resilient to attacks. With the unlock-pattern mode, the highest FPR is 6.9% observed for Pattern 1 with Type-IV attackers. With the touch-gesture mode, the highest FPR is 9.7% observed for Gesture 3 with Type-IV attackers.

Fig. 25a further shows the FPR values with Type-III and Type-IV attackers. It is clear that the unlock patterns/gestures are highly resilient to attacks. With the unlock-pattern mode, the highest FPR is 6.9% observed for Pattern 1 with Type-IV attackers. With the touch-gesture mode, the highest FPR is 9.7% observed for Gesture 3 with Type-IV attackers. The FPRs of these patterns/gestures are below 10% in all cases, which confirms the security of MagAuth. It is also not surprising to see that Type-IV

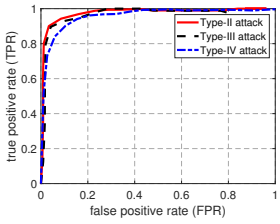


(a) ROC of Pattern 1

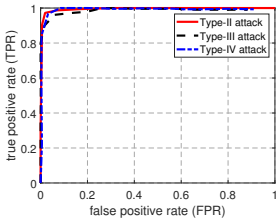
(b) ROC of Gesture 1



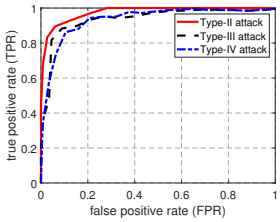
(c) ROC of Pattern 2



(d) ROC of Gesture 2



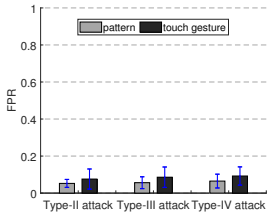
(e) ROC of Pattern 3



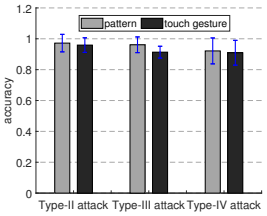
(f) ROC of Gesture 3

Fig. 24: ROC curves of three unlock patterns and touch gestures under three attcks.

attackers have a higher success rate than Type-II and Type-III attackers. In addition, unlock patterns have slightly better performance than touch gestures because the indexes of unlock patterns fed into the feature extraction module are more stable than those of touch gestures.



(a) FPR



(b) accuracy

Fig. 25: MagAuth's resilience to Type-III and Type-IV attacks.

Fig. 25b shows the accuracy with Type-III and Type-IV attackers. MagAuth can achieve very high accuracy for both pattern and touch gesture modes. Moreover, as anticipated, the most capable Type-IV attackers have a success rate slightly higher than Type-II and Type-III attackers.

4.3.3 Computation time

We also evaluated the computation time of MagAuth. The one-time enrollment time was 20 s and 25 s for unlock patterns and touch gestures, respectively, which is quite comparable with that of finger or face authentication on smartphones. Since the classifier is very infrequently updated, we opt to let the cloud train the classifier with user-supplied data and then push the classifier to the smartphone, which is the preference approach advocated by almost all behavioral authentication techniques for mobile devices. Given the nature of our classifier, it took negligible time (much less than a second) to train the classifier on a Dell desktop with 2.67 GHz CPU, 9 GB RAM, and Windows 10 32-bit Professional. In addition, the runtime authentication time measured on Samsung Galaxy S5 with unlock patterns and touch gestures were 1.2 s and 1.5 s, respectively.

5 Related Work

In this section, we briefly outline the prior work most germane to MagAuth.

There are some prior efforts on using hardware tokens for user authentication. For example, the IR ring [32] transmits a cryptographic pseudorandom bit sequence in the form of infrared light pulses to authenticate its wearer to a multi-touch display. In [5], Bojinov and Boneh proposed to build a lightweight, cheap authentication device for unlocking a user's smartphone. They built two token prototypes which can transmit a user's authentication signals to the smartphone via either the magnetic field or the acoustic channel. In addition, the prototype device in [6] transmits a user's authentication information through a capacitive touchscreen. Such hardware tokens have to be specially built and are not available on the market. In addition, these techniques actually authenticate a hardware token rather than a user to the mobile device, so an attacker stealing the token can log into the device as well. In contrast, MagAuth explores COTS magnetic band that have been widely used for wearables. MagAuth is also a two-factor authentication technique built upon the secrecy of a user's self-chosen pattern/gesture, and the special way a user performs a chosen pattern/gesture.

Previous work on the association of wearables and mobile devices (e.g. smartphones) focus on the strong motion correlation of sensor readings in temporal and spatial domains in these two kinds of devices [33] [34] [35]. This property has two sides, to enhance security of mobile devices or to leak information of motion dynamics. In contrast, this paper regards a wearable as the carrier of a token and their applications are totally different.

Also relevant is the line of work on using the magnetometer for text entry or interaction with mobile devices. MagWrite [36] is a digit-entry method based on drawing (writing) digits in the 3D space around the device using a magnet taken in hand. GaussSense [37] enables a stylus to be used as an input device on any thin non-ferromagnetic flat surface by utilizing directional magnetism. MagPen [38] is a magnetically driven pen interface for interacting with mobile devices. MagGetz [39] is a toolkit that enables tangible interaction on and around mobile devices via the magnetic field. Magboard [40] places a magnet in the specific position of a customized keyboard and infers the keystroke by monitoring the magnetic field with the magnetometer in a nearby phone. The authors try to learn the magnetic field data which are only related to keystroke locations and independent from users, mobile devices, and magnets. Therefore, the authors actually fingerprinted the keystroke

on a customized keyboard, not the user. In contrast, our system aims to authenticate a user through specific movement behavior features derived from the magnetic field readings. Thus, the techniques and purposes of these two systems are totally different.

Researchers have explored the magnetometer for user authentication as well. MagiSign in [36] lets a user draw 3D signatures around the mobile device with a magnet, and its security is further evaluated in [41]. In contrast to MagiSign, we performed a more in-depth study about a user's biometrics exhibited in pattern/gesture based user authentication. We also conducted more comprehensive security and usability studies, in which we experimentally show that capable attackers can easily bypass MagiSign but not MagAuth due to very different classification features. Besides, in [9], researchers fuse the motion sensors' data (including the magnetometer) collected by the smartphone for authentication when the user touches the phone's screen. These behavioral characteristics are similar with the touching pressure and face a big challenge of intra-class similarity and variability. To continuously (implicitly) authenticate a user, FingerAuth [42] keeps checking his/her implicit finger motion patterns through the on-device magnetometer and a magnetic ring worn by the user when using the device. By comparison, MagAuth aims to complement pattern-based device authentication rather than continuous authentication, and it quantitatively verifies a user's behavioral biometrics. Therefore, the application contexts of FingerAuth and MagAuth are totally different.

Our work is also related to research on touch-based authentication. The first category of papers use the data passively collected from the IMU sensors and the screen. Hutchins *et al.* [10] characterized a user's PIN by the timing of beats when the user taps the device. TouchIn [7] authenticates a user based on the user's finger-drawing curves. RhyAuth [8] verifies a user according to a sequence of rhythmic taps/slides on the touchscreen device. Authors in [11] studied features of the touch gesture, including the velocity, device acceleration, and stroke time. Authors in [43] extracted the multi-touch information such as velocity, touching pressure, and shape of the traces for user authentication. The features used in these systems are relatively simple and vulnerable to mimicry attacks [12] because they are only related to the fingertip. MagAuth extracts the dynamic features which characterize the 3D relative movement between the user's fingertip and wrist. These features are related to the size and structure of the user's palm and are thus more resilient to mimicry attacks.

Another category of papers explore a user's unique behavioral characteristics by actively generating signals and measuring the response of the user in the inputting process [13]–[15]. In [13], the authors scanned the user's hand posture using active acoustic sensing for authentication. The features used in the system are coarse-grained and sensitive to the environmental acoustic noise. Besides, authors in [14], [15] studied the physical characteristics of touch fingers demonstrated in the vibration signals generated by the vibration motor embedded in the phone. Since the systems rely on device vibration, they may generate audible noise which is improper for many quiet environments such as the meeting room and classroom. In addition, many mobile and IoT devices such as the tablets are not equipped with vibration motor, so the vibration-based techniques cannot be used for these devices. We aim to provide more secure and user-friendly authentication for mobile devices.

There are also rich literature on novel authentication methods for mobile users beyond the traditional password-based approach. For example, Liu *et al.* designed finger-input-based authentication by altering vibration propagation inside a solid surface in [44]. Lu *et al.* proposed lip-reading-based user authentication using acoustic signals in [45]. In [46], Kong *et al.* authenticated users through finger gestures using CSI information extracted from WiFi signals. This line of work is orthogonal and complimentary to our work. Some most recent work explore other potential biometric features [47]. Researchers in [47] recently proposed to utilize the induced body electric potentials caused by the ambient electric field as the human feature. these work require extra electric devices which has not be applied to mobile devices.

Researchers also explore the security of the magnetometer based authentication systems. For example, in [48], the authors proposed one attack scenario in which an adversary can use the readings of the magnetometer in a nearby mobile device to infer the movement of a stylus pen during the input process. This attack is based on a strong assumption that the involved magnet embedded in the pen can only move in a 2D plane without any rotation or vibration. This assumption makes the coordinate transformation from the magnetometer's space to the magnet's space possible. In this case, if the attacker gets the magnetometer's readings, he could recover the magnet movement. However, in our case, the user's wrist moves in a 3D space with dynamic displacement and orientation. Therefore, even if the attacker obtains the magnetometer's readings, he still could not recover the movement trace of the magnet.

6 CONCLUSION

In this paper, we presented the design and evaluation of MagAuth, a secure and usable two-factor user authentication scheme with magnetic wrist wearables. We experimentally confirmed that the finger movement of a user while inputting an unlock pattern or touch gesture can be distinguished by the induced MFV. In addition, we provided guidelines on the selection of unlock patterns and touch gestures to strike a good balance between security and usability according to the magnetic field theory. Finally, comprehensive user experiments confirmed the high security and usability of MagAuth. As the first two-factor authentication scheme based on COTS magnetic wrist wearables, MagAuth has great potential in many applications. We plan to perform more security and usability studies in the future work.

REFERENCES

[1] "face scanner hacking1," 2020. https://www.welivesecurity.com/2019/01/10/face-unlock-many-android-smartphones-falls-photo/
[2] "face scanner hacking2," 2020. https://www.secureforensics.com/blog/android-phones-faceid-hacked/
[3] "face scanner hacking3," 2020. https://www.wired.com/story/hackers-say-broke-face-id-security/

[4] "face scanner hacking4," 2020. https://bgr.com/2017/12/31/iphone-x-face-id-hack-family-members/

[5] H. Bojinov and D. Boneh, "Mobile token-based authentication on a budget," in *ACM HotMobile*, Phoenix, AZ, Apr. 2011.

[6] T. Vu, A. Baid, S. Gao, M. Gruteser, R. Howard, J. Lindqvist, P. Spasojevic and J. Walling, "Distinguishing Users with Capacitive Touch Communication," in *ACM MobiCom*, Istanbul, Turkey, Aug. 2012.

[7] 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, Oct. 2014.

[8] Y. Chen, J. Sun, R. Zhang and Y. Zhang, "Your Song Your Way: Rhythm-Based Two-Factor Authentication for Multi-Touch Mobile Devices," in *IEEE INFOCOM*, Hong Kong, China, Apr. 2015.

[9] A. Buriro, B. Crispo, F. Delfrari and K. Wrona, "Hold and sign: A novel behavioral biometrics for smartphone user authentication," in *IEEE SPW*, San Jose, CA, May 2016.

[10] B. Hutchins, A. Reddy and W. Jin, M. Zhou, M. Li and L. Yang, "Beat-PIN: A User Authentication Mechanism for Wearable Devices Through Secret Beats," in *ACM ASIACCS*, Songdo, Korea, May 2018.

[11] M. Shahzad, A. Liu and A. Samuel, "Behavior Based Human Authentication on Touch Screen Devices Using Gestures and Signatures," in *IEEE Transactions on Mobile Computing*, vol. 16, no. 10, pp. 2726–2741, 2016.

[12] S. Abdul and P. Vir V, "When kids' toys breach mobile phone security," in *ACM CCS*, Berlin, Germany, Nov. 2013.

[13] H. Chen, F. Li, W. Du, S. Yang, M. Conn, and Y. Wang, "Listen to Your Fingers: User Authentication Based on Geometry Biometrics of Touch Gesture," in *ACM Ubicomp*, online, Sep. 2020.

[14] X. Xu, J. Yu, Y. Chen, Q. Hua, Y. Zhu, Y. Chen and M. Li, "TouchPass: towards behavior-irrelevant on-touch user authentication on smartphones leveraging vibrations," in *ACM MobiCom*, London, United Kingdom, Sep. 2020.

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

[16] "wrist-worn devices," 2019. https://www.idc.com/getdoc.jsp?containerId=prUS45521319/

[17] "fitness tracker," 2018. https://www.businesswire.com/news/home/20181206005489/en/Global-Fitness-Tracker-Market-Opportunities---Forecast/

[18] "Magnetic Band," 2020. https://rb.gy/ymijua/

[19] "Magnetic Clasp," 2020. https://www.amazon.com/gp/product/B07VFQBRLT/ref=ppx_yo_dt_b_asin_title_o03_s01?ie=UTF8&psc=1/

[20] "The average iPhone is unlocked 80 times per day," 2016. https://www.businessinsider.com/the-average-iphone-is-unlocked-80-times-per-day-2016-4/

[21] "Magnetic field," 2020. https://www.kjmagnetics.com/glossary.asp

[22] "Materials and Grades," 2018. https://www.amazingmagnets.com/t-magnetic-grade-chart.aspx

[23] "Iron Chrome Cobalt," 2015. https://magnetsim.com/materials/iron-chrome-cobalt

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

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

[26] C. Liu, G. Clark and J. Lindqvist, "Where usability and security go hand-in-hand: Robust gesture-based authentication for mobile systems", in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, Denver, CO, May 2010.

[27] J. Tian and C. Qu and W. Xu and S. Wang, "KinWrite: Handwriting-Based Authentication Using Kinect", in *NDSS*, San Diego, CA, Feb. 2013.

[28] S. Chen and Y. Wang and J. Zheng, "Dissecting pattern unlock: The effect of pattern strength meter on pattern selection," in *Journal of Information Security and Applications*, vol. 19, no. 4-5, pp. 308–320, 2014.

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

[30] J. Bonneau, "The science of guessing: analyzing an anonymized corpus of 70 million passwords", in *IEEE SP*, San Francisco, CA, May 2012.

[31] "Magnetic Clasp1," 2020. https://www.amazon.com/Compatible-Frontier-Stainless-Replacement-Smartwatch/dp/B081BW18ZB/ref=psdc_11591904011_t1_B07S1MCWD1

[32] V. Roth and P. Schmidt and B. Guldenring, "The IR Ring: Authenticating Users' Touches on a a Multi-touch Display", in *ACM UIST*, New York City, NY, Oct. 2010.

[33] C. Leung, C. Fu and P. Heng, "TwistIn: Tangible Authentication of Smart Devices via Motion Co-analysis with a Smartwatch", in *ACM Ubicomp*, Singapore, Oct. 2018.

[34] L. Zhang, D. Han, A. Li, T. Li, Y. Zhang and Y. Zhang, "WristUnlock: Secure and Usable Smartphone Unlocking with Wrist Wearables", in *IEEE CNS*, Washington, D.C., Jun. 2019.

[35] C. Wang, J. Liu, X. Guo, Y. Wang and Y. Chen, "WristSpy: Snooping Passcodes in Mobile Payment Using Wrist-worn Wearables", in *IEEE INFOCOM*, Paris, France, Apr. 2019.

[36] H. Ketabdar, K. Yuksel, A. Jahnbekam, M. Roshandel and D. Skripko, "MagiSign: user identification/authentication based on 3d around device magnetic signatures", in *ACM Ubicomp*, Florence, Italy, Oct. 2010.

[37] R. Liang, K. Cheng, C. Su, C. Weng, B. Chen and D. Yang, "GaussSense: attachable stylus sensing using magnetic sensor grid", in *ACM UIST*, Cambridge, MA, Oct. 2012.

[38] S. Hwang, A. Bianchi, M. Ahn and K. Wohn, "MagPen: magnetically driven pen interactions on and around conventional smartphones", in *ACM MobileHCI*, Miami, Florida, Sep. 2013.

[39] S. Hwang, M. Ahn and K. Wohn, "MagGetz: customizable passive tangible controllers on and around conventional mobile devices", in *ACM UIST*, ST ANDREWS, UK, Oct. 2013.

[40] H. Abdelnasser, M. Youssef and K. Harras, "Magboard: Magnetic-based ubiquitous homomorphic off-the-shelf keyboard", in *IEEE SECON*, London, United Kingdom, Jun. 2016.

[41] A. Sahami, P. Moghadam, H. Ketabdar and A. Schmidt, "Assessing the vulnerability of magnetic gestural authentication to video-based shoulder surfing attacks", in *ACM CHI*, Austin, TX, May 2012.

[42] Y. Liu, M. Yang, Z. Ling and J. Luo, "Implicit Authentication for Mobile Device based on 3D Magnetic Finger Motion Pattern", in *IEEE CSCWD*, Wellington, New Zealand, Apr. 2017.

[43] Y. Song, Z. Cai and Z. Zhang, "Multi-touch Authentication Using Hand Geometry and Behavioral Information", in *IEEE SP*, San Jose, CA, May 2017.

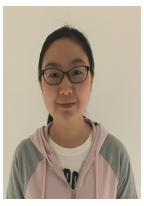
[44] J. Liu, C. Wang, Y. Chen and N. Saxena, "VibWrite: Towards finger-input authentication on ubiquitous surfaces via physical vibration", in *ACM CCS*, Dallas, Texas, Oct. 2017.

[45] L. Lu, J. Yu, Y. Chen, H. Liu, Y. Zhu, Y. Liu and M. Li, "Lippass: Lip reading-based user authentication on smartphones leveraging acoustic signals", in *IEEE INFOCOM*, Honolulu, Hawaii, Apr. 2018.

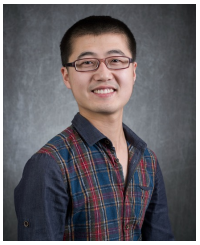
[46] H. Kong, L. Lu, J. Yu, Y. Chen, L. Kong and M. Li, "FingerPass: Finger Gesture-based Continuous User Authentication for Smart Homes Using Commodity WiFi", in *ACM Mobihoc*, Catania, Italy, Jul. 2019.

[47] Z. Yan, Q. Song, R. Tan, Y. Li and A. Kong, "Towards touch-to-access device authentication using induced body electric potentials", in *ACM MobiCom*, Los Cabos, Mexico, Oct. 2019.

[48] Y. Liu, K. Huang, X. Song, B. Yang and W. Gao, "MagHacker: eavesdropping on stylus pen writing via magnetic sensing from commodity mobile devices", in *ACM MobiSys*, Toronto, Canada, Jun. 2020.



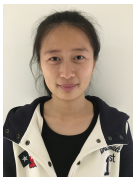
Yan Zhang received the B.S. in Information and Computing Science from Xi'an Jiaotong University, China, in 2014, the M.S. in Communication and Information System from Beijing Normal University, in 2017. Currently, she is a Ph.D. student in Computer Engineering from Arizona State University. Her research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



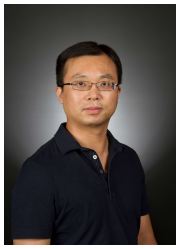
Tao Li received a Ph.D. in Computer Engineering from Arizona State University in 2020, a M.S. in Computer Science & Technology from Xi'an Jiaotong University in 2015, and a B.E. in Software Engineering from Hangzhou Dianzi University in 2012. His primary research is on security and privacy issues in 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).

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.



Lili Zhang received the B.S. Information Security from University of Science and Technology of China, in 2016. Currently, she is a Ph.D. student in Computer Engineering from Arizona State University. Her research interest is about cyber security and privacy issues in mobile systems. She is a student member of IEEE.



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 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. He also chaired the 2017 IEEE Conference on Communications and Network Security (CNS), the 2016 ARO-funded Workshop on Trustworthy Human-Centric Social Networking, the 2015 NSF Workshop on Wireless Security, and the 2010 IEEE GLOBECOM Communication and Information System Security Symposium.