MagPairing: Exploiting Magnetometers for Pairing Smartphones in Close Proximity

Rong Jin*, Liu Shi*, Kai Zeng*, Amit Pande†, Prasant Mohapatra†
*Department of Computer and Information Science, University of Michigan - Dearborn, MI 48128
Email: {jinrong, liushi, kzeng}@umich.edu
†Department of Computer Science, University of California, Davis, CA, 95616
Email: {amit,prasant}@cs.ucdavis.edu

***Abstract*—With the prevalence of mobile computing, lots of wireless devices need to establish secure communication on the fly without pre-shared secrets. Device pairing is critical for bootstrapping secure communication between two previously unassociated devices over the wireless channel. Using auxiliary out-of-band channels involving visual, acoustic, tactile or vibrational sensors has been proposed as a feasible option to facilitate device pairing. However, these methods usually require users to perform additional tasks such as copying, comparing, and shaking. It is preferable to have a natural and intuitive pairing method with minimal user tasks.**

In this paper, we introduce a new method, called *MagPairing*, for pairing smartphones in close proximity by exploiting correlated magnetometer readings. In MagPairing, users only need to naturally tap the smartphones together for a few seconds without performing any additional operations in authentication and key establishment. Our method exploits the fact that smartphones are equipped with tiny magnets. Highly correlated magnetic field patterns are produced when two smartphones are close to each other. We design MagPairing protocol and implement it on Android smartphones. We conduct extensive simulation and experiments to evaluate MagPairing. Experimental results show that MagPairing can successfully pair two smartphones with 4.5 seconds on average. It is immune to man-in-the-middle attack even when the attacker is a few centimeters away from the pairing devices.

I. Introduction

Smartphones have become increasingly popular in recent years, leading to many new applications such as file swapping, music sharing, and collaborative gaming, where nearby users engage in spontaneous wireless data communications using Bluetooth or WiFi interfaces. Such device-to-device connectivity is also required to develop plug-and-play solutions to mobile healthcare industry where multiple wireless body sensors collect vital feeds from human body. It enables continuous user monitoring in home, hospital and outdoor scenarios for ubiquitous health monitoring and emergency medical response.

An important security issue during bootstrap phase is to securely associate two devices and generate shared secret keys to protect the subsequent wireless communications, often without any prior context. Such "device pairing" or "first connect" is critical for bootstrapping secure communication between two previously unassociated devices over the wireless channel.

Using auxiliary out-of-band (OOB) channels to facilitate device pairing has been studied as a feasible option involving visual [1, 2, 3, 4, 5, 6], acoustic [7, 8, 9, 10, 11], tactile [12] or vibrational sensors [13, 14]. However, these methods are not optimized in terms of usability, which is considered of utmost importance in pairing scheme based on OOB channels [1, 15, 16, 17], and require users to perform additional tasks such as copying, comparing and shaking. It is preferable to have a natural and intuitive pairing method designed with minimal user tasks.

In this work, we focus on device pairing using magnetometer sensors in the smartphones and develop an intuitive scheme, called MagPairing, which pairs two smartphones when they are tapped together. We prefer the use of magnetometer sensors over audio and visual schemes [1, 7] because this involves minimal user intervention and achieves better usability. Device pairing using accelerometer sensors [13, 14] involves asking user to perform some typical task such as shaking the phones which is less intuitive than simply tapping the devices.

In MagPairing, users only need to naturally tap the smart phones together for a few seconds without performing any additional operations in authentication and key establishment. The embedded magnetometer sensor in smartphones provides a measure of magnetic field along X, Y, and Z directions [18, 19]. Our method exploits the fact that smartphones are equipped with tiny magnets themselves. When two smartphones are tapped together, their magnetometers are reading the magnetic fields at almost the same point, yielding highly correlated sensor data of magnetic field patterns. The sensor data are used to authenticate early established DH-key to prevent man-in-the-middle attacks.

In MagPairing, we tackle the challenge that sensor data collected by distributed smartphones are not synchronized and spatial aligned. Moreover, we consider the problem that user may wag and rotate unconsciously when holding smartphones. We implemented MagPairing on Google Nexus 5 smartphones running Android. Experiments show that MagPairing achieves a high successful pairing rate with short paring time around 4.5 seconds on average.

The main contributions of this paper are summarized as follows:

1) We design a protocol to achieve secure smartphone device paring by using the correlated readings on respective magnetometers.
2) We conduct extensive simulations to evaluate our method.
3) We implement the protocol on Android smartphones and conduct experiments to evaluate and validate our proposed method.

Although MagPairing is validated on smartphones, it can be applied to facilitate the pairing of other wireless devices which are equipped with magnetometers, such as generic body sensors and wearable computing devices [20, 17], providing a method for intuitive secure device pairing.

II. Related work

One prominent research direction for device pairing is the use of auxiliary – also referred to as "out-of-band" (OOB) channels, which are both perceivable and manageable by the users who own and operate the devices. Existing option involves 1) visual, 2) acoustic, 3) tactile or 4) vibrational sensors.

A. Visual Channel

In some early approaches [2, 3, 4], OOB data are encoded into images and the users are asked to compare them on two devices. In a more recent approach [5], "Seeing-is-Believing" (SiB), one device encodes the public key into a two-dimensional bar code and displays it on its screen, and the other device "reads it" using a photo camera, operated by the user. Another approach [6], similar to SiB, requires that LED-equipped device transmits OOB data via the blinking. However, overall, image comparison is considered obtrusive and requires the user's attention.

B. Acoustic Channel

In [8], audio channel is used to represent the information exchanged over the main wireless channel. There are two variants: "Display–Speaker" and "Speaker–Speaker", where the user compares the displayed sentence with its vocalized counterpart and two vocalized sentences, respectively. Follow-on works [9, 10] consider that pairing devices have no common wireless channel at pairing time. They use pure audio to transmit cryptographic protocol messages and requires the user to merely monitor device interaction for any extraneous interference. A pairing method based on synchronized audio-visual patterns [11] are further developed. The proposed methods, "Blink–Blink", "Beep–Beep" and "Beep–Blink", involve users comparing very simple audiovisual patterns, e.g., in the form of "beeping" and "blinking", transmitted as simultaneous streams, forming two synchronized channels. Similar to the use of visual signal, acoustic signal is considered noisy and requires the user's attention. Moreover, it is not preferable in a crowded environment.

C. Tactile Channel

Another approach [12], "Button-Enabled Device Authentication (BEDA)", suggests pairing devices with the help of user button presses, thus utilizing the tactile OOB channel. This method has several variants: "LED–Button", "Beep–Button", "Vibration–Button", and "Button–Button". In the first two variants, the sending device blinks its LED (or vibrates or beeps) and the user presses a button on the receiving device. In the Button–Button variant, the user simultaneously presses buttons on both devices. However, the button press itself requires human intervention.

D. Vibration Channel

"Smart-Its-Friends" [13] and "Shake-Well-Before-Use" [14] exploit common movement pattern to communicate a shared secret to both devices as they are shaken together by the user. The user need to hold the devices together and perform shaking for around 5 seconds. However, their work need an additional action "shaking" comparing with MagPairing.

A usability analysis of the existing popular device pairing schemes are presented in [15]. It reports that many of the existing schemes have a large computational time and high fatal error rate, and are perceived difficultly by the end-user. Another comprehensive study on usability of secure device pairing schemes [21] advocates the user of limited visual information over methods that require comparing more extensive information. Results from another usability study [16] show that simple number comparison is quite attractive overall, being both fast and secure as well as readily acceptable by users over blinking, audio, visual, phrase comparison approaches. It takes an average time of 8.6 seconds but requires human intervention.

As a conclusion, existing works on pairing are still not optimized in terms of usability. Thus, in this paper, aiming at an intuitive, fast, secure, and user-friendly device pairing, we propose MagPairing, which achieves high successful pairing rate with short pairing time and is immune to the man-in-the-middle attack.

III. Overview of MagPairing

In this paper, we consider the scenario where two smartphones, Alice and Bob, want to bootstrap a secure communication by generating a shared secret key between themselves over a wireless channel without any pre-shared secret. The two smartphones are both equipped with magnetometers and wireless interfaces (i.e., WiFi).

Attacker model: We assume a powerful active attacker. The attacker can intercept all messages sent by Alice and Bob and inject arbitrary messages over its wireless interface. It can make independent connections with the victims and relays messages between them, making them believe that they are talking directly to each other over a private connection, while in fact the entire conversation is controlled by the attacker. Such an attacker is generally known as man-in-the-middle (MITM) attacker. The attacker may have more powerful computational ability than the smartphones and can conduct sophisticated signal processing. The attacker is also equipped with magnetometers. It can be close to Alice and Bob, but cannot be at the same points as Alice and Bob due to the physical constraints.

Fig. 1 shows the work flow of MagPairing. After triggering, two devices are tapped together and initialize a standard Diffie-Hellman (DH) key agreement protocol. During DH key exchange, the two devices records their magnetometer readings simultaneously. Because the generated DH key is susceptible to MITM attack, after DH key exchange completes, the devices need to verify that their keys are equivalent. They encrypt and exchange their magnetometer readings via an interlock protocol, which guarantees no disclosure of sensor data during transmission. Afterwards, sensor data are decrypted, and mutual authentications are executed locally on the respective devices by comparing the similarity of the sensor data collected separately. If the similarity check is passed, the early generated DH key will be used for consecutive secure communication. On the other hand, the attacker is unable to sense or fabricate a correlated sensor data, thus would not be able to pass the similarity check and would be detected if it ever launched a MITM attack.

Note that raw sensor data are not directly suitable for similarity check, because they are collected by different smartphones and are not synchronized and spatially aligned. Moreover, users may wag and rotate unconsciously when holding smartphones. Thus, a series of sensor data pre-processing must be conducted before the similarity check. We introduce sensor data pre-processing in detail in section IV. We introduce

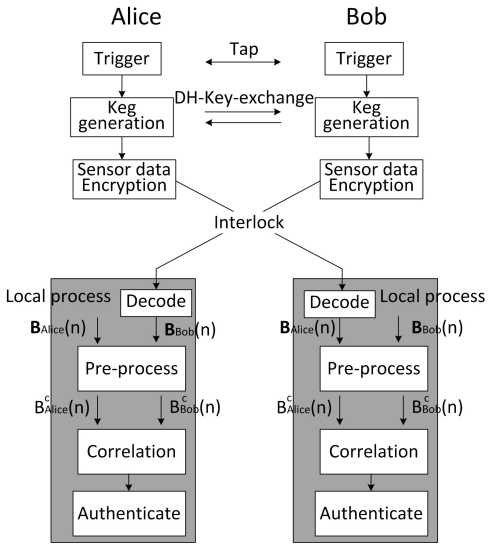

Fig. 1. Architecture of MagPairing protocol

triggering, DH key agreement, interlock schemes as well as the whole MagPairing protocol in Section V.

IV. Sensor Data Pre-Processing

According to the superposition principle, the net magnetic field is simply the vector sum of all contributing fields. When two smartphones are tapped together, their magnetometers are reading the magnetic fields at almost the same point, which can be approximated as

$$\mathbf{B}_{net} = \mathbf{B}_{Earth} + \mathbf{B}_1 + \mathbf{B}_2 \tag{1}$$

where $\mathbf{B}_{Earth}$ is the Earth's magnetic field. $\mathbf{B}_1$ and $\mathbf{B}_2$ are the magnetic fields produced by Alice and Bob's inside magnets.

Conceptually, $\mathbf{B}_{net}$ can be sampled to time series by Alice and Bob respectively as their shared information for authentication. However, $\mathbf{B}_{net}$ is the net magnetic field vector with respect to the Earth's coordinates. The magnetometers' readings, $\mathbf{B}_{Alice}$ and $\mathbf{B}_{Bob}$, are under Alice and Bob's own coordinates (as shown in Fig. 2).

$$\begin{aligned} \mathbf{B}_{Alice} &\approx \mathbf{T}_{Earth\to Alice}\mathbf{B}_{net} \\ \mathbf{B}_{Bob} &\approx \mathbf{T}_{Earth\to Bob}\mathbf{B}_{net} \end{aligned} \tag{2}$$

where $\mathbf{T}_{Earth\to Alice}$ and $\mathbf{T}_{Earth\to Bob}$ are transformation matrices from the Earth's coordinates to Alice and Bob's coordinates, respectively.

Four pre-processing tasks executed as consecutive steps are used to sample and align the sensor data so that correlation can build on normalized time series. 1) sensor data acquisition (output $\mathbf{B}_{Alice}$, $\mathbf{B}_{Bob}$), 2) synchronization (output $\mathbf{B}^a_{Alice}$, $\mathbf{B}^a_{Bob}$), 3) spatial alignment (output $\mathbf{B}^b_{Alice}$, $\mathbf{B}^b_{Bob}$), 4) mean value removal (output $\mathbf{B}^c_{Alice}$, $\mathbf{B}^c_{Bob}$),

A. Sensor data acquisition

In this step, magnetic field data $\mathbf{B}_{net}$ is sampled by Alice and Bob, yielding $\mathbf{B}_{Alice}(i)$ and $\mathbf{B}_{Bob}(i)$, respectively. Sensor data acquisition is conceptually straightforward, but requires careful implementation. Magnetometer readings are assumed to be available in the form of time series of magnetic fields in all three directions, sampled at equidistant time steps. These must be taken locally and not be communicated wirelessly – for security purposes, it is critical not to leak any of this raw data, which can be difficult considering the possibility of powerful side-channel attacks. Our practical experience shows a sample rate of 50 Hz to be appropriate.

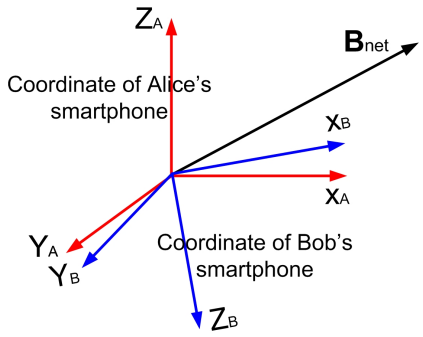

Fig. 2. Magnetic field under the coordinates of Alice and Bob

B. Sensor data synchronization

As the two devices sample magnetic field time series independently, we require sensor data synchronization for comparison. We assume that Alice and Bob are equipped with similar clocks so that the difference in sampling rate is insignificant. However, they may start the sampling at different time. Therefore, we need to synchronize the starting points for time series comparison.

Suppose Alice and Bob sample the magnetic field respectively to get N_A and N_B sample points at each direction, yielding

$$\begin{aligned} \mathbf{B}_{Alice} &= [\mathbf{B}^x_{Alice}{}^T,\ \mathbf{B}^y_{Alice}{}^T,\ \mathbf{B}^z_{Alice}{}^T]^T \\ \mathbf{B}_{Bob} &= [\mathbf{B}^x_{Bob}{}^T,\ \mathbf{B}^y_{Bob}{}^T,\ \mathbf{B}^z_{Bob}{}^T]^T \end{aligned} \tag{3}$$

where

$$\mathbf{B}^{x(y,z)}_{Alice} = [B^{x(y,z)}_{Alice}(1), B^{x(y,z)}_{Alice}(2), \cdots, B^{x(y,z)}_{Alice}(N_A)] \tag{4}$$

$$\mathbf{B}^{x(y,z)}_{Bob} = [B^{x(y,z)}_{Bob}(1), B^{x(y,z)}_{Bob}(2), \cdots, B^{x(y,z)}_{Bob}(N_B)] \tag{5}$$

where $B^x_{Alice}(i)$, $B^y_{Alice}(i)$, $B^z_{Alice}(i)$ represent the ith sample points at X, Y, and Z directions respectively measured by Alice. $B^x_{Bob}(i)$, $B^y_{Bob}(i)$, $B^z_{Bob}(i)$ represent the sample points measured by Bob.

We then calculate the average cross correlation between $\mathbf{B}_{Alice}$ and $\mathbf{B}_{Bob}$

$$C(n) = \frac{|C_x(n)| + |C_y(n)| + |C_z(n)|}{\sigma^x_{Alice}\sigma^x_{Bob} + \sigma^y_{Alice}\sigma^y_{Bob} + \sigma^z_{Alice}\sigma^z_{Bob}} \tag{6}$$

where $C_x(n)$, $C_y(n)$ and $C_z(n)$ are the cross correlations at X, Y and Z directions by shifting Alice's readings to the left by n, respectively, defined as

$$C_{x(y,z)}(n) = \frac{1}{N_s - n}\sum_{i=1}^{N_s-n}(B^{x(y,z)}_{Alice}(i+n) - \mu^{x(y,z)}_{Alice})(B^{x(y,z)}_{Bob}(i) - \mu^{x(y,z)}_{Bob}) \tag{7}$$

where

$$\begin{aligned} \mu^{x(y,z)}_{Alice} &= 1/N_A\sum_{i=1}^{N_A}[B^{x(y,z)}_{Alice}(i)] \\ {\sigma^{x(y,z)}_{Alice}}^2 &= 1/N_A\sum_{i=1}^{N_A}[B^{x(y,z)}_{Alice}(i) - \mu^{x(y,z)}_{Alice}]^2 \end{aligned} \tag{8}$$

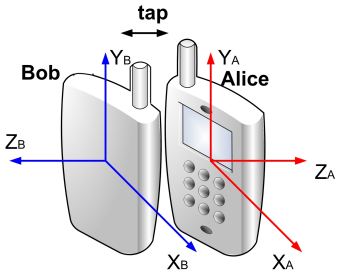

Fig. 3. Two smart phones are face to face tapped

where $N_S = \mathbf{Min}(N_A, N_B)$. $\boldsymbol{\mu}_{Alice} = [\mu^x_{Alice}, \mu^y_{Alice}, \mu^z_{Alice}]$ is the three directional mean magnetic field measured by Alice. $\boldsymbol{\sigma}_{Alice} = [\sigma^x_{Alice}, \sigma^y_{Alice}, \sigma^z_{Alice}]$ is the standard deviation. Accordingly, $\boldsymbol{\mu}_{Bob}$ and $\boldsymbol{\sigma}_{Bob}$ are the mean and standard deviation of magnetic field measured by Bob.

When tapping two smart phones together as illustrated in Fig. 3, the relative coordinate relationship of Alice and Bob is "face to face". That is, if $\mathbf{B}_{Alice}$ and $\mathbf{B}_{Bob}$ are synchronized:

$$\begin{aligned} B^x_{Alice}(i) &\approx B^x_{Bob}(i) \\ B^y_{Alice}(i) &\approx B^y_{Bob}(i) \\ B^z_{Alice}(i) &\approx -B^z_{Bob}(i) \end{aligned} \tag{9}$$

It can be derived by substituting (9) into (6) that

$$C(n) \leq C(0) \approx 1 \tag{10}$$

When there is a synchronization offset n_0 between Alice and Bob, we can get a similar equation

$$C(n) \leq C(n_0) \approx 1 \tag{11}$$

Thus in implementation, we can take a peak search on $C(n)$ to get the synchronization offset n_0. We compensate the offset to get synchronized data $\mathbf{B}^a_{Alice}$ and $\mathbf{B}^a_{Bob}$.

It must be pointed out that $C(n)$ is used for the purpose of synchronization, which is not a *qualified correlation* for the authentication of the shared key. In practice, Alice and Bob may not have an ideal face to face coordinate relationship as illustrated in Fig. 3. Deviations come from the differences of the smart phones' manufacturing, so that two devices may have heterogeneous internal coordinates. Moreover, users can hardly tap two smartphones exactly face to face. Fig. 2 illustrates practical coordinate relationship. To achieve higher correlation coefficient, spatial alignment is further required to match Alice and Bob's coordinates.

C. Spatial alignment

After two smartphones are tapped together, their relative coordinate relationship is fixed, which can be written as

$$\mathbf{T}_{Alice \to Bob} \times \mathbf{B}^a_{Alice} = \mathbf{B}^a_{Bob} \tag{12}$$

where $\mathbf{T}_{Alice \to Bob} = [\mathbf{T}_{Earth \to Alice}]^{-1} \times \mathbf{T}_{Earth \to Bob}$ is the coordinate transformation matrix between Alice and Bob, representing the spatial misalignment.

The least squares estimation of $\mathbf{T}_{Alice \to Bob}$ is

$$\hat{\mathbf{T}}_{Alice \to Bob} = \mathbf{B}^a_{Bob} \times pinv(\mathbf{B}^a_{Alice}) \tag{13}$$

where $pinv(\mathbf{B}_{Alice})$ is the Generalized inverse matrix of $\mathbf{B}_{Alice}$.

We compensate $\mathbf{T}_{Alice \to Bob}$ to get the spatial aligned sensor data $\mathbf{B}^b_{Alice}$ and $\mathbf{B}^b_{Bob}$.

D. Mean value removal

Final correlation should be performed on the randomness of the sensor data after removing the mean value $\boldsymbol{\mu}^{Alice}$, $\boldsymbol{\mu}^{Bob}$, otherwise a *Reply attacker* can keep the magnetometer readings in the first attempt, and replay the readings in the second attempt.

A problem is that users prone to wag and rotate unconsciously when holding smartphones, which makes the mean value of the sensor data a time varying parameter $\boldsymbol{\mu}^{Alice}(t)$, $\boldsymbol{\mu}^{Bob}(t)$. To deal with the problem, we take short term average on $\mathbf{B}^b_{Alice}$ and $\mathbf{B}^b_{Bob}$ to follow the change of the mean value.

$$\begin{aligned} \mu^{x(y,z)}_{Alice}(m) &= \frac{1}{N_w} \sum_{m-(N_w-1)/2}^{m+(N_w-1)/2} B^{x(y,z)}_{Alice}(i) \\ \mu^{x(y,z)}_{Bob}(m) &= \frac{1}{N_w} \sum_{m-(N_w-1)/2}^{m+(N_w-1)/2} B^{x(y,z)}_{Bob}(i) \end{aligned} \tag{14}$$

We remove the impact of mean value to get calibrated sensor data $\mathbf{B}^c_{Alice}$ and $\mathbf{B}^c_{Bob}$.

$$\begin{aligned} \mathbf{B}^c_{Alice} &= \mathbf{B}^b_{Alice} - \boldsymbol{\mu}_{Alice} \\ \mathbf{B}^c_{Bob} &= \mathbf{B}^b_{Bob} - \boldsymbol{\mu}_{Bob} \end{aligned} \tag{15}$$

E. Sensor data reshaping

In our case, the magnetometer readings are three dimensional time series with arbitrary length – $3 \times N$ matrices (we call them *Matrix format*). They must be reshaped to $1 \times 3N$ strings to perform the correlation (we call them *String format*). In addition, our encryption and decryption are based on block ciphers. Messages must fit in the size of the cipher block length (we call them *Block format*).

Fig. 4 illustrates the sensor data reshaping scheme in MagPairing. To transform Matrix format to String format, we simply align their row vectors together, and an opposite operation is used for inverse transformation.

$$\begin{aligned} \mathbf{B}^{str}_{Alice} &= [\mathbf{B}^x_{Alice}, \quad \mathbf{B}^y_{Alice}, \quad \mathbf{B}^z_{Alice}] \\ \mathbf{B}^{str}_{Bob} &= [\mathbf{B}^x_{Bob}, \quad \mathbf{B}^y_{Bob}, \quad \mathbf{B}^z_{Bob}] \end{aligned} \tag{16}$$

To transform String format to Block format, we truncate the string to several blocks and add zeros at the end of the string to fit in the final block, and an opposite operation is used for inverse transformation.

In the rest of this paper, we will omit data format transformation process and assume that the sensor data are always transformed to correct format before processing.

F. Correlation

The correlation is performed on pre-processed sensor data $\mathbf{B}^c_{Alice}$, $\mathbf{B}^c_{Bob}$ of String format, which can be written as

$$r = \frac{\mathbf{B}^c_{Alice} {\mathbf{B}^c_{Bob}}^T}{\sqrt{\mathbf{B}^c_{Alice} {\mathbf{B}^c_{Alice}}^T \mathbf{B}^c_{Bob} {\mathbf{B}^c_{Bob}}^T}} \tag{17}$$

Two devices that are tapped together will experience similar, but not exactly the same magnetic field patterns due to their spatial separation, manufacturing differences and the impact of noise. According to our experiment, r is around 0.7, We set the threshold $r_0 = 0.5$ to judge whether $\mathbf{B}^c_{Alice}$ and $\mathbf{B}^c_{Bob}$ are correlated.

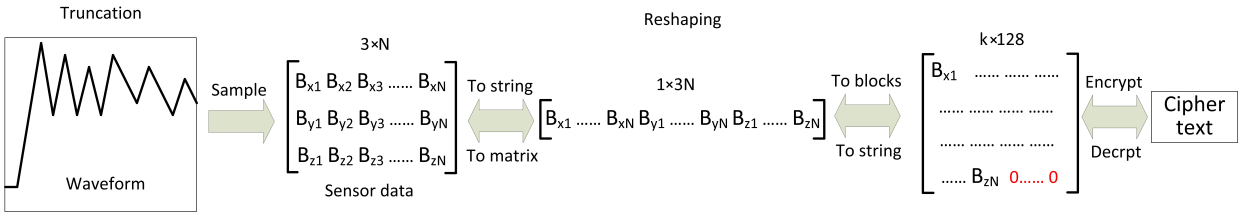

Fig. 4. Sensor data reshaping method in MagPairing

V. Key Establishment and Authentication Protocol

In this section, we describe the detail of MagPairing protocol, which includes pairing process triggering, DH key exchange, and interlock protocol used for mutual authentication.

A. Pairing process triggering

Triggering can be direct user input, e.g. pressing an "authenticate now" button on both devices within a short time frame, or implicit, simply by starting to tap both devices together. We prefer the second protocol due to its ease of use.

As two smartphones approach, Alice's magnetometer readings $\mathbf{B}_{Alice}(t)$ change abruptly due to the proximity of Bob. The same thing also happens to Bob. We use this abrupt magnetometer reading change as the signal of the start of device pairing. Note that there are other situations which will also cause the changes of magnetometer readings on smartphone: 1) the user shakes or rotates the smartphone. This will cause the change of the relative coordinate relationship between the Earth and the smartphone. The magnetic field vector in free space is always aligned to the Earth's magnetic line. As a result, the magnetometer readings on the smartphone at two individual directions (or all three directions) change abruptly. 2) a magnet or magnetic substance is coming close to the smartphone.

The triggering process must be carefully designed to have small false alarm probability, otherwise the battery of smartphone will be drained quickly. The problem of verifying that two devices are tapping together becomes a classification problem. To separate from the first situation, we use the amplitude change of the magnetometer readings ($B_{Alice} = |\mathbf{B}_{Alice}|$, $B_{Bob} = |\mathbf{B}_{Bob}|$, $|\mathbf{B}| = \sqrt{B_x^2 + B_y^2 + B_z^2}$ as triggering indicator. We ignore the magnetometer reading changes at individual directions, which will not lead to the amplitude changes. To separate from the second situation, first we set up a triggering interval $[B_{low},\ B_{high}]$ which matches the magnetic field strength of tapped smartphones to reduce the probability of false alarm. If the magnetometer's amplitude change falls into the interval $B_{Alice} \in [B_{low},\ B_{high}]$, the smartphone will try to contact to the respective device by sending a request. The pairing process will be terminated if no reply is received within τ seconds (e.g. 3 seconds). After this termination, the pairing process will not be restarted unless a pre-defined minimum time interval t_{inv} is passed or the pairing process is restarted manually by the user.

B. Diffie-Hellman and interlock

In order to establish an identical key, we create a cryptographically secure secret key via a standard Diffie-Hellman (DH) key agreement. Because DH is susceptible to MITM attack, it should be verified that their keys are equivalent.

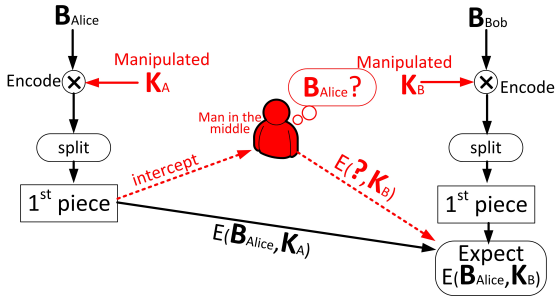

Fig. 5. Flow chart of interlock

We then authenticate the key using the correlated strings as illustrated in Fig. 1.

To achieve the goal of authentication, both strings need to be available completely to both devices. Therefore, the magnetometers' readings, $\mathbf{B}_{Alice}$ and $\mathbf{B}_{Bob}$ must be exchanged during the interactive protocol – in a way that does not reveal them to an attacker.

This sensor data exchange is done with an interlock protocol [22, 14]. Interlock is an efficient (in terms of message length) method to verify that two parties share the same key. The strength of the protocol lies in the fact that half of an encrypted message cannot be decrypted. Thus, if Eve begins her attack and intercepts Bob and Alice's keys, Eve will be unable to decrypt Alice's half-message (encrypted using her key) and re-encrypt it using Bob's key. Subsequently, Eve who try to separately generate independent keys with Alice and Bob will be exposed (shown in Fig. 5).

C. MagPairing Protocol

For the formal descriptions of our protocol, we use the following notation: $c = E(K, m)$ describes the encryption of plain text m under key K with a symmetric cipher, and $m = D(K, c)$ is the corresponding decryption. $H(m)$ represents the hashing of message m with some secure hash function, and $m|n$ is the concatenation of strings m and n. The notation M[a : b] is used to describe the substring of a message M starting at bit a and ending at bit b. The symbol $\oplus$ describes bit-wise XOR. We use AES with 128-bit key size as a block cipher for $E()$ and $D()$.

Fig. 6 shows our authentication protocol. Using DH key agreement, Alice and Bob generate two shared keys K_A, K_B and K_A^{Sess}, K_B^{Sess}, where it is impossible to infer one from the other (under the assumption that the hash function does not allow to find a pre-image). Creating two keys, one for authentication, one as session key, provides forward secrecy. Because DH is susceptible to MITMA, the devices need to verify that their keys are equivalent. The unique key property of DH guarantees with a very high probability, that is, if $K_A = K_B$, there can be no attacker E with $K_{EA} = K_A$ and $K_{EB} =$

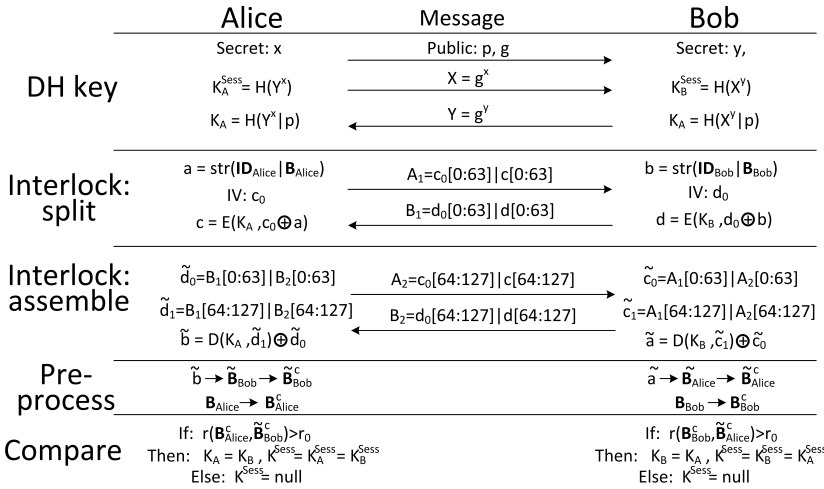

Fig. 6. Protocol: Diffie-Hellman key agreement followed by the exchange of sensor data via interlock

K_B, and subsequently, no $K_{EA}^{Sess} = K_A^{Sess}$ and $K_{EB}^{Sess} = K_B^{Sess}$.

After DH keys are established, Alice and Bob encrypt their IDs and magnetometer readings with the keys. Because interlock is based on block ciphers, we reshape $\mathbf{B}_{Alice}$ and $\mathbf{B}_{Bob}$ to Block format of standard length as introduced in subsection IV-E, getting a and b. For our authentication protocol, we simply use the cipher block chaining (CBC) mode with a random initialization vector (IV). The resulting ciphertexts c and d are then split into two messages by concatenating the first halves of cipher blocks into the first messages A1 and B1, and the second halves into the second messages A2 and B2. This ensures that the attacker cannot decrypt any of the blocks or learn parts of the plain text messages.

After exchanging their messages a and b, Alice and Bob verify the similarity between two sensor data. This is done by using the pre-processing and correlation method described in section IV. A threshold r_0 is used to judge whether the device pairing is successful.

D. Security Analysis

In the following, we analyse MagPairing tackling passive attacks, MITM attacks, replay attacks and reflection attacks respectively.

1) *Passive Attacks:* A passive attacker only eavesdropping on the communications will not interrupt the key agreement process. In this case, Alice and Bob can successfully generate DH key and pass the authentication. The DH key is guaranteed to be computational secure and will not be revealed to the attacker.

2) *MITM Attacks:* It is known that MITM attack is achieved by an attacker making independent keys K_A, K_B and connections with the victims and relaying messages between them. This makes the victims believe that they are talking directly to each other over a private connection. But in fact, the entire conversation is controlled by the attacker. To remain undetected, the attacker must pass the authentication. Normally, if Alice encrypts the packet $\mathbf{B}_{Alice}$ with the key K_A, the attacker can decrypt the packet by K_A, re-encrypt the packet with the key K_B, and forward it to Bob.

However, this attack won't succeed against interlock protocol, since the attacker cannot decrypt half-message as shown in Fig. 5.

After the attacker receives half-message, she is left with only two options: either to forward the original packets, or to create packets on her own. In the former case, Alice and Bob will be unable to decrypt the messages properly, because they do not share the same key. In the latter case, the attacker must guess the contents of the messages, and encrypt them with the appropriate keys, before it has access to the actual messages. When the messages sent by Alice and Bob have an entropy of e bits, this leaves the attacker with a single 2^{-e} chance of correctness.

As a conclusion, what can a MITM attacker do is causing the failure of device pairing, but MITM attacks can not pass the authentication process and will be detected by our protocol. Thus, the conversations will not be transmitted by compromised keys and revealed to the attacker.

3) *Replay Attacks:* A smart MITM attacker may keep the magnetometer readings in the first attempt. Then in the second attempt, it may just replay the readings. However this attack won't succeed because there is no strong correlation between consecutive measurements. What we make use of is the random component in the readings, not the raw readings. The randomness is induced by human motion and ambient noise, which is not correlated temporally or spatially.

4) *Reflection Attacks:* A MITM attacker may reflect the messages sent by Alice and Bob back to themselves. In this way, Alice and Bob will receive their own sensor data, yielding high correlations equaling to 1, and pass the correlation check. However, this method won't succeed and can be easily detected by checking the ID of the message sender.

VI. Performance Evaluation

The correlation check in MagPairing can be modeled as a hypothesis test:

$$H_0 : \textit{No attack}$$
$$H_1 : \textit{There is an attack}$$

where H_0 and H_1 are the null and alternative hypothesis, respectively.

The performance of the hypothesis test is usually evaluated by the **receiver operating characteristic** (ROC) curve. The ROC curve plots the false alarm rate α against detection rate β. The false alarm rate is the probability of assuming an attack but there is actually no attack. The detection rate is the probability of detecting the attack when the attack happens.

Our goal is to achieve high detection rate with low false alarm rate.

According to the protocol implementation, we have

$$\begin{aligned} \alpha &= Pr(r \leq r_0 | H_0) = \int_{r \leq r_0} f_0(r) dr \\ \beta &= Pr(r \leq r_0 | H_1) = \int_{r \leq r_0} f_1(r) dr \end{aligned} \tag{18}$$

where f_0 and f_1 are the pdf of the sample correlation coefficient under null and alternative hypothesis, respectively. These two pdfs are hard to obtain due to the unavailability of the close-form expression for distribution of the sample correlation coefficients. Even under Gaussian assumption, there is no close form solution for the sample correlation coefficient given the population correlation coefficient [23, 24]. Next, we will numerically analyze the correlation and test the performance of MagPairing.

In the simulation, we randomly generate Alice's three dimensional magnetic field: $\mathbf{B}^{field}_{Alice}(n) = \mathbf{B}^{0}_{Alice} + \mathbf{B}^{disturb}_{Alice}(n)$; where $\mathbf{B}^{0}_{Alice}$ is the mean net magnetic field, which is the sum of all contributing fields: the Earth's magnetic field, Alice and Bob's inside magnet's field. $\mathbf{B}^{disturb}_{Alice}(n)$ is the magnetic field disturbance due to the collision of phones when tapping them together and the user's unintended shaking. According to our experiments, We set $\mathbf{B}^{0}_{Alice}$ at each direction to a uniform distribution between $[-400\mu T \ 400\mu T]$. We set $\mathbf{B}^{disturb}_{Alice}(n)$ at each direction to a zero mean Gaussian distribution with the standard deviation of $40\mu T$. Then, we generate a similar magnetic field (but not exactly the same magnetic field since their measurements do not take place at exactly the same spot) $\mathbf{B}^{field}_{Bob}(n)$ for Bob. We set the correlation coefficient ρ between $\mathbf{B}^{field}_{Alice}(n)$ and $\mathbf{B}^{field}_{Bob}(n)$ to 0.9. Further, we assign to Bob a face-to-face coordinate relationship with respect to Alice, and introduce a deviation (because their coordinate relationship is not exactly face-to-face due to their heterogeneous internal coordinates and non-ideal user operations). The deviation is introduced by a rotation at a random direction of a random degree uniformly distributed between 0 to 20 degree. After that, a random synchronization offset of less than 10 sample points is added between $\mathbf{B}^{field}_{Alice}(n)$ and $\mathbf{B}^{field}_{Bob}(n)$.

The magnetometer readings are generated as follows: $\mathbf{B}_{Alice}(n) = \mathbf{B}^{field}_{Alice}(n) + w(n)$, $\mathbf{B}_{Bob}(n) = \mathbf{B}^{field}_{Bob}(n) + w(n)$; where $w(n)$ represent a zero-mean Gaussian noise. As introduced in section V, Alice and Bob use standard Diffie-Hellman key agreement protocol to generate a 128 bit key. Then, they reshape their three dimensional sensor data to strings a, b and exchange them by the interlock protocol.

1) False alarm rate: In this simulation, Alice and Bob establish an identical key K. They use K to encrypt, exchange and decrypt the sensor data (assuming no bit error in wireless transmission).

2) Detection rate: In this simulation, we assume that DH key agreement has been manipulated by a MITM attacker, who generates two keys, one with Alice K_A and one with Bob K_B separately. Alice (Bob) reshapes its magnetometer readings to a (b), encrypt it with K_A (K_B) and send its first half to Bob (Alice). The attacker intercepts the piece but it cannot decrypt the content a (b) now. The attacker has no other choice but to guess the content a_A (b_A) by generating a random string with the same distribution as Alice (Bob). The attacker then encrypts a_A (b_A) with K_B (K_A) and forwards it to Bob (Alice). Then Alice (Bob) sends its second half to Bob (Alice). This time, the attacker intercepts and gets both halves; it decrypts the content a (b) now, encrypts the second half of a (b) with K_B (K_A) and forward it to Bob (Alice).

After the sensor data exchange, Alice and Bob use the method in section IV to pre-process the data and compute the correlation. They use a threshold r_0 to judge whether there is an attacker.

A. Impact of SNR

We set SNR to 10dB, 5dB and 0dB respectively. We vary r_0 from 0 to 1 to draw the ROC curve. For each point, we run the same simulation 10000 times to get the false alarm rate and detection rate. In the simulation, the sensor data are quantized to Bytes (8 bits), the number of effective Bytes at each direction is $N_{point} = 20$ (60 for all three directions). Fig. 7 (left) shows that good performance is achieved even with ultra low SNR (for SNR = 0, 90% detection rate with 7% false alarm rate).

B. Impact of the number of effective Bytes

We set N_{point} to 30, 20 and 10, respectively. We fix SNR to 5dB. We use the same method in previous subsection to draw the ROC curve. Fig. 7 (middle) shows that good performance is achieved even with ultra small N_{point} (for $N_{point} = 10$, 95% detection rate with 2% false alarm rate). Therefore, MagPairing requires a very short period of sensor data capturing time. For example, assuming effective Bytes are sampled at 10Hz, it only takes 2s to capture sufficient amount of sensor data of $N_{point} = 20$.

C. Correlation PDF

We numerically draw the probability density function (PDF) of correlation with SNR = 5dB, $N_{point} = 20$. The PDFs under two conditions (no attacker, an attacker) are well separated, and thus appropriate for threshold detection. When there is no attacker, Alice and Bob have an average correlation of 0.78; the correlations are more than 0.6 in most cases. When there is an attacker, the average correlation decreases to 0.15; the correlations are less than 0.4 in most cases.

VII. Implementation

We implemented MagPairing and conduct experiments using two Google Nexus 5 smartphones running Android version 4.4.2 developed by *Eclipse*. As introduced in section V, after triggering, two phones collect magnetometer readings separately. Then, they use standard DH-protocol to establish a shared key of 1024 bits (hash to 128 bits); where p is a random prime number of 1024 bits, and $g = 5$. The sensor data are encrypted split into two messages and exchanged through Interlock protocol. Two phones decrypt sensor data, pre-process (as introduced in section IV) and compute the correlation r. We set the correlation threshold to $r_0 = 0.5$ to judge whether device pairing is successful. There are 6 message transmissions during the whole device pairing process as illustrated in Fig. 6.

Fig. 8 (left) shows the result of an experiment when we tap two phones face to face as requested by MagPairing (less than 1cm). In this case, $r \approx 0.7 > r_0$; thus the authentication and device pairing succeeded as desired. Fig. 8 (right) shows the result of an experiment when we separate two phones to

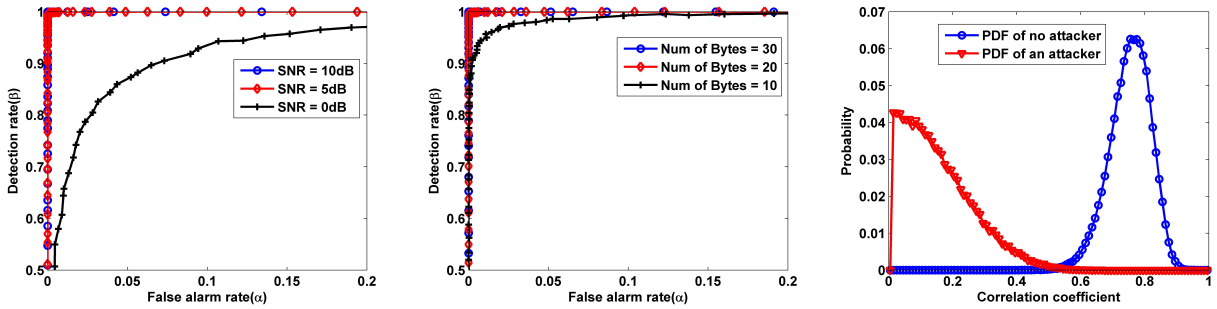

Fig. 7. Performance of MagPairing (left) impact of SNR (middle) impact of the number of effective Bytes at each direction N_{point} (right) PDF of correlation

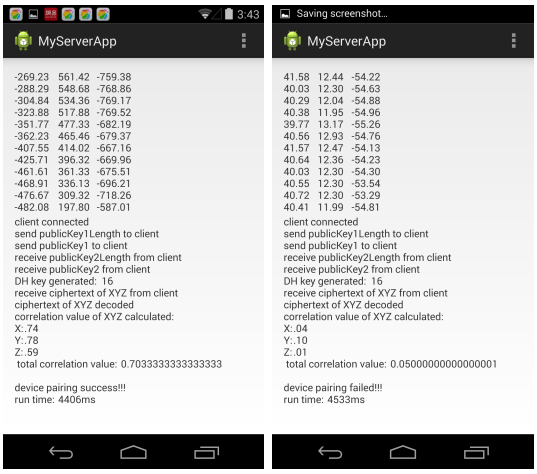

Fig. 8. Screen shot of MagPairing (left) two phones are tapped face to face as requested (right) two phones are separated

perform the same experiment (about 20cm apart). In this case, correlation decreases dramatically, due to the separation of magnetometers $r \approx 0.05 < r_0$; thus the authentication and device pairing failed as expected.

A. Sensor data correlation verification

In this experiment, we are aiming at verifying the similarity of sensor data. We capture magnetometer readings on two smartphones when they are tapping together and analyse the data on a computer. Fig. 9 (a) shows the raw sensor data $\mathbf{B}_{Alice}$, $\mathbf{B}_{Bob}$ at X direction. The two sequences have the similar trends of rise and fall. It can be seen that the correlation between raw data is not high, which is a normal phenomenon as explained in section IV, and sensor data preprocessing is needed. First, we perform synchronization and spatial alignment to get $\mathbf{B}^b_{Alice}$, $\mathbf{B}^b_{Bob}$ as introduced in subsection IV-B and IV-C. Fig. 9 (b) shows the preprocessed sensor data. An obvious improvement on similarity is achieved. Next, we remove the short-term mean value to get $\mathbf{B}^c_{Alice}$ and $\mathbf{B}^c_{Bob}$ as introduced in subsection IV-D. Fig. 9 (c) shows the preprocessed sensor data. It can be seen from the figure that the two sequences look like zero mean random noises with high similarity. The result confirms that magnetometer readings on two smartphones are highly correlated when they are close to each other. Authentication and key agreement can be performed based on the sensor data. Finally, we use another smartphone (10cm away from tapped ones) to eavesdrop the sensor data. Fig. 9 (d) shows the sensor data collected by Alice and a nearby attacker. The similarity is very low, which confirms that the sensor data can be treated as a shared secret between legitimate devices.

B. Usability test

In this experiment, we ask 6 different testers without any prior training to implement MagPairing application. Each participant is requested to naturally tap two smartphones together 20 times. We record their success rate and time consuming.

Table 1 shows the result. The success rates are more than 90% for all of the testers. The minimum, maximum, average time consuming are 4.1 s , 5.9 s and 4.5 s. The result validates that MagPairing is fast and easy to use in practice.

Tester	Success rate	Minimum time	Maximum time	Average time
No. 1	95%	4.2s	5.9s	4.6s
No. 2	90%	4.1s	4.8s	4.4s
No. 3	90%	4.2s	4.9s	4.5s
No. 4	100%	4.2s	4.6s	4.4s
No. 5	90%	4.2s	4.9s	4.5s
No. 6	95%	4.1s	4.7s	4.4s

TABLE I
SUCCESS RATE AND TIME CONSUMING ON DIFFERENT TESTERS

VIII. Discussion

Although we only implement MagPairing on two Google Nexus 5 smartphones, the method is generally applicable to various kinds of smartphones equipped with magnetometers. MagPairing requires two smartphones are tapped together such that their magnetometers are close to each other. However, the location of magnetometer may varies from smartphone to smartphone. In some cases, "face-to-face" may not be the optimal choice to guarantee the proximity of two smart phones' inside magnetometers. Since MagPairing is designed for ordinary users, the inner structure of smartphone must be assumed unknown to the users.

To deal with the problem, we can develop an app that stores the magnetometer location information for the various popular smartphones on the market. Then two smartphones can exchange this information before pairing and the user can find out the best way to tap the smartphones.

IX. Conclusion

We have designed a reliable, fast and easy-to-use secure device pairing scheme, MagPairing, by using magnetometers on smartphones. Our method exploits the fact that smartphones are equipped with tiny magnets. Highly correlated magnetic

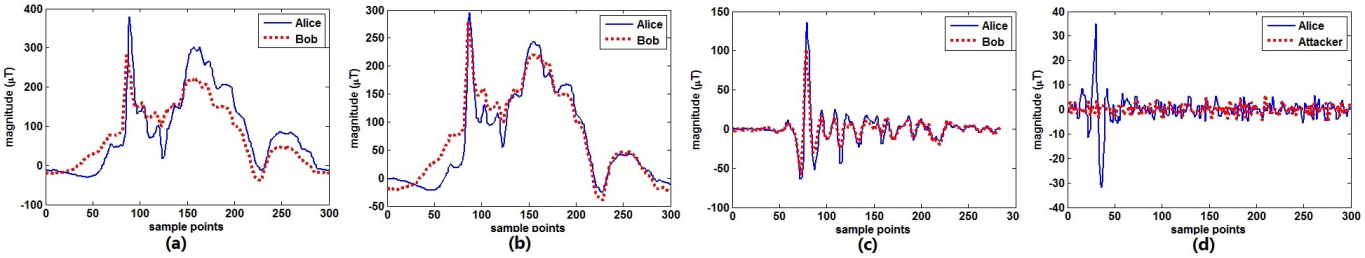

Fig. 9. Captured magnetometer readings of x-direction on both smartphones (a) raw sensor data $\mathbf{B}_{Alice}$, $\mathbf{B}_{Bob}$ (b) synchronized and spatial aligned sensor data $\mathbf{B}^b_{Alice}$, $\mathbf{B}^b_{Bob}$ (c) pre-processed sensor data after mean value removed $\mathbf{B}^c_{Alice}$, $\mathbf{B}^c_{Bob}$ (d) comparison with the sensor data of a nearby attacker

field patterns are produced when two devices are close to each other.

Numerical analysis show that MagPairing achieves high detection rate and low false alarm rate even under low SNR or within a short period of sensor data capturing time. We implemented MagPairing and conduct experiments using two Google Nexus 5 smartphones. Concept proof experiment confirms that magnetometer readings captured by two smartphones are highly correlated. Authentication and key agreement can be performed based on the sensor data. Usability experiment shows that MagPairing has a high success rate and short total device paring time in practice. More investigation on the usability and security strength of MagPairing used for heterogeneous devices and under active attackers will be our future work.

REFERENCES

[1] N. Saxena, M. B. Uddin, and J. Voris, "Universal device pairing using an auxiliary device," in *Proceedings of the 4th symposium on Usable privacy and security*. ACM, 2008, pp. 56–67.

[2] A. Perrig and D. Song, "Hash visualization: a new technique to improve real-world security," in *In International Workshop on Cryptographic Techniques and E-Commerce*, 1999, pp. 131–138.

[3] C. Ellison and S. Dohrmann, "Public-key support for group collaboration," *ACM Trans. Inf. Syst. Secur.*, vol. 6, no. 4, pp. 547–565, Nov. 2003. [Online]. Available: http://doi.acm.org/10.1145/950191.950195

[4] V. Roth, W. Polak, E. Rieffel, and T. Turner, "Simple and effective defense against evil twin access points," in *Proceedings of the First ACM Conference on Wireless Network Security*, ser. WiSec '08. New York, NY, USA: ACM, 2008, pp. 220–235. [Online]. Available: http://doi.acm.org/10.1145/1352533.1352569

[5] J. McCune, A. Perrig, and M. Reiter, "Seeing-is-believing: using camera phones for human-verifiable authentication," in *Security and Privacy, 2005 IEEE Symposium on*, May 2005, pp. 110–124.

[6] N. Saxena, J.-E. Ekberg, K. Kostiainen, and N. Asokan, "Secure device pairing based on a visual channel," in *Security and Privacy, 2006 IEEE Symposium on*, May 2006, pp. 6 pp.–313.

[7] M. T. Goodrich, M. Sirivianos, J. Solis, C. Soriente, G. Tsudik, and E. Uzun, "Using audio in secure device pairing," *International Journal of Security and Networks*, vol. 4, no. 1, pp. 57–68, 2009.

[8] M. Goodrich, M. Sirivianos, J. Solis, G. Tsudik, and E. Uzun, "Loud and clear: Human-verifiable authentication based on audio," in *Distributed Computing Systems, 2006. ICDCS 2006. 26th IEEE International Conference on*, 2006, pp. 10–10.

[9] C. Soriente, G. Tsudik, and E. Uzun, "Hapadep: Human-assisted pure audio device pairing," in *In ISC*, 2008, pp. 385–400.

[10] M. T. Goodrich, M. Sirivianos, J. Solis, C. Soriente, G. Tsudik, and E. Uzun, "Using audio in secure device pairing," *Int. J. Secur. Netw.*, vol. 4, no. 1/2, pp. 57–68, Feb. 2009. [Online]. Available: http://dx.doi.org/10.1504/IJSN.2009.023426

[11] R. Prasad and N. Saxena, "Efficient device pairing using "human-comparable" synchronized audiovisual patterns," in *Proceedings of the 6th International Conference on Applied Cryptography and Network Security*, ser. ACNS'08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 328–345. [Online]. Available: http://dl.acm.org/citation.cfm?id=1788857.1788877

[12] C. Soriente, G. Tsudik, and E. Uzun, "Beda: Button-enabled device association," in *International Workshop on Security for Spontaneous Interaction IWSSI,UbiComp Workshop Proceedings*, 2007.

[13] L. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl5, and H.-W. Gellersen, "Smart-its friends: A technique for users to easily establish connections between smart artefacts," in *Ubicomp 2001: Ubiquitous Computing*, ser. Lecture Notes in Computer Science, G. Abowd, B. Brumitt, and S. Shafer, Eds. Springer Berlin Heidelberg, 2001, vol. 2201, pp. 116–122.

[14] R. Mayrhofer and H. Gellersen, "Shake well before use: Authentication based on accelerometer data," in *Pervasive computing*. Springer, 2007, pp. 144–161.

[15] E. Uzun, K. Karvonen, and N. Asokan, "Usability analysis of secure pairing methods," in *Financial Cryptography and Data Security*. Springer, 2007, pp. 307–324.

[16] A. Kumar, N. Saxena, G. Tsudik, and E. Uzun, "Caveat eptor: A comparative study of secure device pairing methods," in *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*. IEEE, 2009, pp. 1–10.

[17] K. K. Venkatasubramanian, A. Banerjee, and S. K. S. Gupta, "Pska: usable and secure key agreement scheme for body area networks," *IEEE Transactions on Information Technology in Biomedicine,*, vol. 14, no. 1, pp. 60–68, 2010.

[18] H. Ketabdar, A. Jahanbekam, K. A. Yuksel, T. Hirsch, and A. Haji Abolhassani, "Magimusic: using embedded compass (magnetic) sensor for touch-less gesture based interaction with digital music instruments in mobile devices," in *Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction*. ACM, 2011, pp. 241–244.

[19] H. Ketabdar, K. A. Yüksel, and M. Roshandel, "Magitact: interaction with mobile devices based on compass (magnetic) sensor," in *Proceedings of the 15th international conference on Intelligent user interfaces*. ACM, 2010, pp. 413–414.

[20] X. H. Le, R. Sankar, M. Khalid, and S. Lee, "Public key cryptography-based security scheme for wireless sensor networks in healthcare," in *Proceedings of the 4th International Conference on Uniquitous Information Management and Communication*. ACM, 2010, p. 5.

[21] A. Kobsa, R. Sonawalla, G. Tsudik, E. Uzun, and Y. Wang, "Serial hook-ups: a comparative usability study of secure device pairing methods," in *Proceedings of the 5th Symposium on Usable Privacy and Security*. ACM, 2009, p. 10.

[22] R. L. Rivest and A. Shamir, "How to expose an eavesdropper," *Commun. ACM*, vol. 27, no. 4, pp. 393–394, Apr. 1984. [Online]. Available: http://doi.acm.org/10.1145/358027.358053

[23] J. F. Kenney and E. S. Keeping, *Mathematics of Statistics*. Pt. 2, 2nd ed. Van Nostrand, 1951.

[24] K. Zeng, K. Govindan, D. Wu, and P. Mohapatra, "Identity-based attack detection in mobile wireless networks," in *INFOCOM, 2011 Proceedings IEEE*, April 2011, pp. 1880–1888.