Future Generation Computer Systems 108 (2020) 1324–1337

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

FingerAuth: 3D magnetic finger motion pattern based implicit authentication for mobile devices

Yiting Zhang [a,b], Ming Yang [a,*], Zhen Ling [a], Yaowen Liu [a], Wenjia Wu [a]

[a] School of Computer Science and Engineering, Southeast University, Nanjing, China
[b] School of Computer Science & Technology, Nanjing University of Posts and Telecommunications, Nanjing, China

HIGHLIGHTS

- A 3D magnetic finger motion pattern based implicit authentication method is proposed.
- Magnetometer sensor data is used to derive 3D magnetic finger motion pattern.
- The results of two-round usability tests proved the uniqueness and permanence.

ARTICLE INFO

Article history:
Received 20 November 2017
Received in revised form 21 January 2018
Accepted 3 February 2018
Available online 8 February 2018

Keywords:
Behavioral biometrics
Implicit authentication
Mobile devices

ABSTRACT

Smart devices, as the most widely used platforms for the mobile cyber–physical system (CPS) applications, such as smart home and health care systems, are becoming the prime targets of various attackers for users' considerable private and confidential data in them. To fight against side channel attacks aiming to obtain credentials, e.g., passwords, during the process of user authentication, touch pattern based implicit authentication has been proposed. However, such a defensive technique fails to obtain an entire pattern of user operation by deriving user operation data via a touch-enabled screen. Considering that user operations, including on-screen and in-air finger movements, are performed in three-dimensional (3D) space, we propose a novel 3D magnetic finger motion pattern based implicit authentication technique, referred to as FingerAuth. To use FingerAuth, a user operates on her mobile device, e.g., texting a message and browsing websites, with a magnetic ring on the finger she uses. With the help of a built-in three-axis magnetometer on the mobile device, we can derive the 3D magnetic finger motion pattern as a human behavioral feature for implicitly authenticating the user. By using machine learning techniques, a robust 3D magnetic finger motion pattern detection model can be constructed. Two rounds of usability tests are conducted for the evaluation of FingerAuth. In the initial usability test targeting a given group of smart device users, we test the uniqueness of the proposed trait in typing scenario, achieving high average accuracy of 96.38%, low average false acceptance rate (FAR) of 4.06%, and false rejection rate (FRR) of 3.18%. In the second user usability test, we further evaluate the permanence of 3D finger motion pattern in multiple user–device interaction scenarios. There is an interim of two-week period between the training data collection phase and the testing data collection phase. The results of the high accuracy of over 80%, as well as the FAR and FRR of below 15%, indicate the applicability of FingerAuth.

1. Introduction

Mobile cyber–physical systems are a prominent subcategory of cyber–physical systems. Smart phones, with significant computational resources, multiple sensory input/output devices and communication mechanisms, etc., serve as ideal platforms for mobile CPS applications, e.g., smart home and health care systems. As the most commonly used devices to access various services in these systems, smart phones save extensive sensitive user information. As a result, a growing number of viruses, Trojan horses, and mobile computing worms that target smart phones have been found in the past a few years [1–3]. To prevent disclosure of users' private and confidential data, authentication techniques are pervasively adopted. However, most current authentication techniques (e.g., password, fingerprint recognition and Android pattern look) used on smart devices nowadays are merely invoked at the beginning of a session. Therefore, by retrieving the authentication credential through diverse side channels [4–8], attackers could

* Corresponding author.
E-mail addresses: ytzhang@seu.edu.cn (Y. Zhang), yangming2002@seu.edu.cn (M. Yang), zhenling@seu.edu.cn (Z. Ling), liuyaowen@seu.edu.cn (Y. Liu), wjwu@seu.edu.cn (W. Wu).

https://doi.org/10.1016/j.future.2018.02.006
0167-739X/

still pose a severe security threat and thus perform impersonation attacks against mobile devices in these systems.

Despite of some secure input methods [9] proposed to defend against side channel attacks, implicit authentication [10,11] is generally regarded as a more promising technique to resolve the above issue. Differing from explicit authentication which requires users to perform predefined authentication actions, either by entering the password or placing the finger on top of certain sensor, implicit authentication senses and employs the traits of users in a more

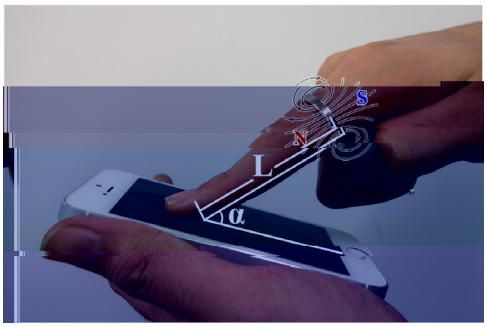

Fig. 1. A magnetic ring on the user's index finger.

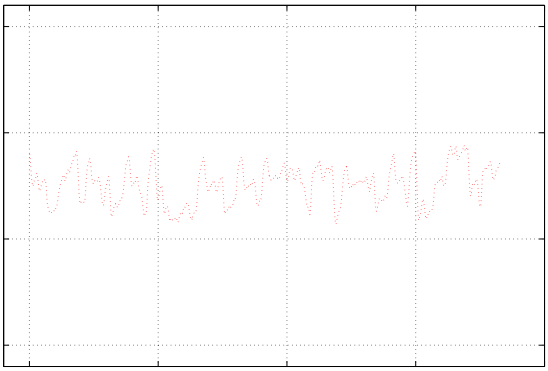

Fig. 2. Magnetometer readings during sentence typing.

3.1. Threat model

Assuming that an attacker is able to obtain a legitimate user's authentication credential including PIN, password or even fingerprint, there would be a high possibility that she could bypass the explicit authentication mechanism which is widely applied in most mobile devices, or have the device under control during an authorized session by all means. Due to the lack of effective protection mechanism adopted by the operating system of the device, the attacker could effortlessly obtain legitimate users' privacy and priceless information.

3.2. Basic idea

What we propose is that the behavioral biometric extracted from users' finger motion during daily interactions with the mobile device could be helpful in implicit authentication. Considering that most user–device interactions happen between the finger and the touchscreen, our implicit authentication through the 3D finger motion pattern ensures users' superior security compared to the earlier work which tend to extract their finger motion pattern with the touchscreen data only.

A built-in three-axis magnetometer is required for deriving the 3D magnetic finger motion pattern. For the purpose, the user is asked to wear a magnetic ring on one of her fingers as shown in Fig. 1. So that when she interacts with the device, the magnetic ring will cause changes in the magnetic field value around the device, which could be sensed by a built-in magnetometer. The changing pattern of magnetic field value indicates the 3D motion pattern of the finger. The magnetometer readings from a typical typing scenario are shown in Fig. 2. During the normal user–device interactions, we record the readings and apply machine learning techniques to them so to implicitly verify whether the current user is a legitimate one or not. Fig. 3 presents the workflow of our proposed system. We elaborate on the workflow as follows.

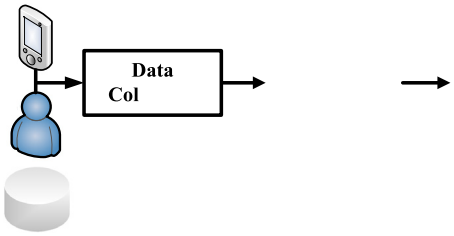

Fig. 3. Workflow of the FingerAuth approach.

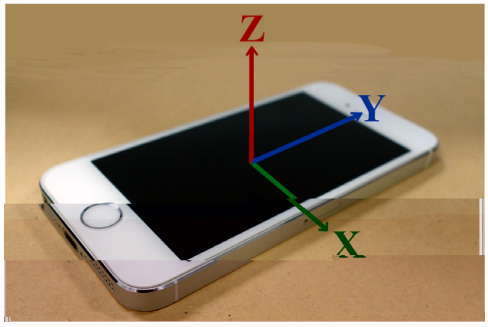

Fig. 4. Coordinate system of the used mobile device.

3.3. Sensor data preprocessing

We collect and obtain readings on the magnetometer every day when the user wearing a magnetic ring interacts with the mobile device. Having eliminated the background magnetic field, we divide the magnetic field data into segments corresponding to on-screen and in-air gestures. What is noteworthy is that the device attitude data is recorded for the cancellation of background magnetic field, while the touchscreen data is for data segmentation purpose.

A magnetometer refers to a three-axis sensor which is typically applied for navigation on mobile devices. On iOS platform, the sensor is often of vector magnetometer type and can measure the vector components of a magnetic field at a point in space. Fig. 4 depicts the coordinate system used on iOS devices, in which the field strength sensed by a magnetometer along each axis is in units of microteslas, while the direction of the field is represented by signs of sensor readings. The readings of strength and signs change as the finger moves around the device. What is more, different locations might have an impact on the environment magnetic field, which we have to exclude during study to mitigate unexpected influence.

3.3.1. Outlier processing and multi-sensor time alignment

To eliminate the influence of the sensor outlier in the experiment, the potential magnetic field sensor outlier data should be filtered. Through the analysis of the magnetic field sensor data, it is found that the normal value ranges from tens to hundreds. Accordingly, we set an upper limit of the magnetic field strength value as $T = 1000$. If the reading of the magnetic field sensor on any axis of magnetic field meets the criteria of $|B_k.i/| > T;$ where $k \in \{x; y; z\}$ and $i \in [1; n]$, it is treated as an outlier. The filter process of the magnetic field sensor outlier is as follows:

- Traverse each data in the record of the magnetic field sensor data $B_k(i)$;
- Based on the upper limit T, determine whether the current magnetic field sensor data is normal or not. If not, the outlier would be processed in the following step. Otherwise, the next data would be processed;
- Determine whether the current data belongs to the first record of the magnetic field sensor record. If so, set its value to zero and process the next data. If not, proceed to the next step;
- Determine whether the current data belongs to the last record of the magnetic field sensor record. If so, set its value to zero and process the next data. Otherwise, proceed to the next step;
- Determine whether the data before and after the current data is an outlier or not. If both are not, the current data will be set to the value of an average of the data before and after it, and the next data will be processed. Otherwise, the current data will be set to zero and the next data will be processed.

There must exist differences in sampling frequency among a variety of built-in sensors. Taking the iPhone 5s used in the experiment for example,the 0 frequency-4.(8(of)-458(its-4.(8(built-ie)]TJ0 0 0 rg 0 0 0 RG0 0 0 rg 0 0 0 RG 0 -10.623
is
Thebefo,is
alignmene on the data of sensone before the data oe
mulampta sensone inseivene the

(in)-307(ysis)-3 [(experime,ne)-403the the magnetic field sensor dataasofaoe
,ne
record31577yrd31364(the)1577quehmarksensor

Table 1
Definition of some features.

Feature	Definition
Coefficient of Variation	$C_V = \frac{\sigma}{\mu}$
Kurtosis	$\mathrm{K} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^4}{(\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2)^2}$
Skewness	$s = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^3}{(\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2})^3}$
Root Mean Square	$rms = \sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}$
Zero Crossing Rate	$zcr = \frac{1}{n-1}\sum_{i=2}^{n}1_{R<0}(x_i \cdot x_{i-1})$ where $1_{R<0}$ is an indicator function

Table 2
Counting information of input instances.

Participant ID	Count of Instances with Different Labels		Total Count
	Legitimate	Illegitimate	
#1	1529	1526	3055
#2	1563	1554	3117
#3	1904	1904	3808
#4	1920	1918	3838
#5	1544	1540	3084
#6	1470	1470	2940
#7	1561	1554	3115
#8	1443	1442	2885
#9	1937	1932	3869
#10	1528	1526	3054
#11	1456	1456	2912
#12	1570	1568	3138
#13	1687	1680	3367
#14	1375	1372	2747
#15	1462	1456	2918

Eq. (7). The last step is to further classify magnetometer data into smaller segments in terms of on-screen and in-air finger movements for the purpose of later data analysis with touch information recorded from touch sensor.

4. First study: uniqueness of 3D magnetic finger motion pattern in typing scenario

In this section, we conduct the first usability study to test the uniqueness of the proposed trait among a given group of smart device users.

4.1. Feature extraction

After the data is properly preprocessed, a feature extraction process is performed. For each magnetic field data segment S_i obtained from data segmentation phase, a corresponding feature vector $\mathbf{F} = \{f_1(S_i); f_2(S_i); \cdots; f_n(S_i)\}$ is extracted for each axis data. Sixteen features are adopted: Mean, Median, Variance, Standard Deviation, Mode, Coefficient of Variation, Kurtosis, Skewness, Root Mean Square, Zero Crossing Rate, and the 1st, 5th, 25th, 75th, 95th, 99th Percentile. Apart from some well-known ones, the definitions are given in Table 1.

4.2. Data collection

We design and conduct extensive experiments to test the applicability of using magnetometer to collect and extract motion pattern information of the finger with a magnetic ring on, and the effectiveness of utilizing this pattern to implicitly authenticate the user. In this study, only the typing scenario is considered, which mainly involves tap gesture, as well as in-air gestures between taps.

In order to collect sensor data in the scenario mentioned above, an app for iOS devices is designed and implemented, and it logs data from touch and magnetic field sensors, as well as device attitude data for preprocessing purpose while the user is typing. Fifteen volunteers from our campus are recruited to participate in the data collection activity, and each one is asked to type the same ten predefined sentences for three times using the app we have developed. Since the application scenario is that a user's mobile phone needs to be able to identify whether the current user is the owner, the same iPhone 5s smartphone is used, as well as the same magnetic ring, which is put on each participant's right index finger with identical direction, and all operations are performed using the index finger. Before each collection session, background magnetic field value without the presence of magnetic ring is also collected for background magnetic field cancellation purpose. Each extracted feature vector is first labeled with corresponding participant's name to make the data traceable. Then, each participant is assumed as a legitimate user in turn. Corresponding data is copied and labeled with the string ``legitimate'', and approximately the same amount of ``illegitimate'' data is produced by evenly copying data from other participants with the labels are changed to ``illegitimate''. The newly generated data, referred to as input instances, is stored in specific format that the machine learning software later used could utilize it. Counting information of input instances for each participant is as Table 2 shows.

4.3. Performance evaluation

We use both classification and authentication metrics to evaluate the performance of the proposed approach, specifically, classification accuracy, false acceptance rate (FAR), and false rejection

Table 3
Evaluation results of first study.

Participant ID	Naive Bayes			Random Forest			Support Vector Machine		
	Accuracy	*FAR*	*FRR*	*Accuracy*	*FAR*	*FRR*	*Accuracy*	*FAR*	*FRR*
#1	90.44%	18.02%	1.11%	97.68%	3.54%	1.11%	93.72%	7.27%	5.30%
#2	76.48%	44.66%	2.50%	97.34%	2.25%	3.07%	95.73%	4.89%	3.65%
#3	64.44%	65.97%	5.15%	95.06%	5.36%	4.52%	87.50%	11.08%	13.92%
#4	64.36%	69.24%	2.08%	95.44%	5.53%	3.59%	85.10%	19.34%	10.47%
#5	65.99%	60.91%	7.19%	99.42%	0.78%	0.39	98.80%	1.69%	0.71%
#6	83.57%	17.96%	14.90%	97.79%	1.43%	2.99%	94.80%	4.69%	5.71%
#7	67.67%	61.39%	3.40%	96.73%	4.25%	2.31%	86.04%	7.79%	20.12%
#8	68.77%	59.85%	2.63%	97.61%	1.32%	3.47%	95.42%	4.37%	4.78%
#9	72.09%	54.24%	1.65%	96.33%	1.97%	5.37%	87.34%	12.63%	12.70%
#10	73.12%	52.10%	1.70%	95.68%	6.29%	2.36%	89.95%	10.16%	9.95%
#11	66.41%	65.80%	1.37%	90.69%	12.84%	5.77%	77.30%	19.71%	25.69%
#12	71.03%	57.02%	0.96%	95.60%	4.34%	4.46%	87.09%	11.86%	13.95%
#13	92.16%	2.44%	13.22%	98.40%	0.48%	2.73%	96.35%	3.39%	3.91%
#14	88.75%	6.49%	16.00%	98.91%	0.66%	1.53%	97.67%	2.62%	2.04%
#15	68.71%	60.58%	2.12%	93.04%	9.82%	4.10%	75.91%	25.41%	22.78%
Average	74.27%	46.44%	5.06%	96.38%	4.06%	3.18%	89.91%	9.79%	10.38%

Table 4
Geometric features.

No.	Features	Description
1	lenOfLineSeg	The distance between the first point and the last point
2	avgLineSegLen	The average distance between the adjacent points
3	angleBtwnFirstLastVec	The angle between the vector formed by the first two points and the vector formed by the last two points
4	angleBtwnVecXYPlane	The angle between the vector formed by the first and last point and the *XY* plane
5	angleBtwnVecXZPlane	The angle between the vector formed by the first and last point and the *XZ* plane
6	angleBtwnVecYZPlane	The angle between the vector formed by the first and last point and the *YZ* plane
7	angleBtwnPlaneXYPlane	The angle between the plane defined by the first, middle and last point and the *XY* plane
8	angleBtwnPlaneXZPlane	The angle between the plane defined by the first, middle and last point and the *XZ* plane
9	angleBtwnPlaneYZPlane	The angle between the plane defined by the first, middle and last point and the *YZ* plane
10	lenFPointXYPlane	The distance between the first point and the *XY* plane
11	lenFPointXZPlane	The distance between the first point and the *XZ* plane
12	lenFPointYZPlane	The distance between the first point and the *YZ* plane
13	lenLPointXYPlane	The distance between the last point and the *XY* plane
14	lenLPointXZPlane	The distance between the last point and the *XZ* plane
15	lenLPointYZPlane	The distance between the last point and the *YZ* plane
16	lenMPointXYPlane	The distance between the middle point and the *XY* plane
17	lenMPointXZPlane	The distance between the middle point and the *XZ* plane
18	lenMPointYZPlane	The distance between the middle point and the *YZ* plane
19	volOfFirstCuboid	The volume of the cuboid that contains the first two data points, divided by the number of data points
20	volOfLastCuboid	The volume of the cuboid that contains the last two data points, divided by the number of data points
21	volOfCuboid	The volume of the cuboid that contains all the data points, divided by the number of data points

Table 5
Counting information of train and test instances.

ID	Sentence typing		Picture browsing		Web surfing	
	Train	Test	Train	Test	Train	Test
#1	3055 (1529/1526)	4748 (2396/2352)	1094 (548/546)	979 (499/480)	1766 (884/882)	1532 (776/756)
#2	3117 (1563/1554)	4023 (2021/2002)	1548 (778/770)	905 (463/442)	2409 (1205/1204)	2576 (1302/1274)
#3	3808 (1904/1904)	3753 (1891/1862)	869 (435/434)	468 (242/226)	2549 (1275/1274)	3628 (1822/1806)
#4	3838 (1920/1918)	4195 (2109/2086)	989 (499/490)	446 (234/212)	2166 (1088/1078)	3686 (1852/1834)
#5	3084 (1544/1540)	3953 (1993/1960)	1803 (907/896)	775 (398/377)	2420 (1216/1204)	2102 (1052/1050)
#6	2940 (1470/1470)	4808 (2414/2394)	1769 (887/882)	1045 (535/510)	2660 (1330/1330)	2331 (1169/1162)
#7	3115 (1561/1554)	4127 (2083/2044)	1206 (604/602)	595 (303/292)	2410 (1206/1204)	1842 (932/910)
#8	2885 (1443/1442)	3846 (1942/1904)	2219 (1113/1106)	1061 (539/522)	2163 (1085/1078)	1452 (738/714)
#9	3869 (1937/1932)	3744 (1882/1862)	1626 (814/812)	834 (421/413)	2268 (1134/1134)	2316 (1168/1148)
#10	3054 (1528/1526)	4005 (2017/1988)	1576 (792/784)	978 (500/478)	2522 (1262/1260)	1653 (841/812)
#11	2912 (1456/1456)	3923 (1977/1946)	1038 (520/518)	403 (207/196)	1823 (913/910)	2073 (1051/1022)
#12	3138 (1570/1568)	3949 (1989/1960)	1435 (721/714)	598 (306/292)	1855 (931/924)	1104 (558/546)
#13	3367 (1687/1680)	4260 (2146/2114)	1156 (582/574)	538 (272/266)	3506 (1756/1750)	2938 (1482/1456)
#14	2747 (1375/1372)	4458 (2246/2212)	1176 (588/588)	523 (269/254)	1466 (738/728)	2053 (1031/1022)
#15	2918 (1462/1456)	3809 (1919/1890)	1297 (653/644)	787 (403/384)	1772 (890/882)	1166 (592/574)

rate (FRR). In classification scenarios, accuracy is the proportion of correctly classified instances over a given instances set, while FAR and FRR are used in biometric systems to measure the probability of incorrectly accepting a malicious user and falsely rejecting a legitimate user respectively [13]. Let *TP* denote the number of instances that correctly classified as legitimate, *TN* denote the number of instances correctly classified as illegitimate, *FP* denote the number of instances that incorrectly classified as legitimate, *FN* denote the number of instances that incorrectly classified as illegitimate. Then, the formulas for calculating the accuracy, the FAR and the FRR are shown as follows.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

$$FAR = \frac{FP}{FP + TN} \tag{9}$$

$$FRR = \frac{FN}{FN + TP} \tag{10}$$

Recall that the goal is to study the feasibility of using the 3D magnetic finger motion pattern to verify current user's authenticity, which could be abstracted as a classification problem in the domain of machine learning over feature vectors extracted from corresponding sensor data. Since the study itself is not targeting at machine learning issues, the widely used open-source

Table 6
Algorithms used in second study.

No.	Algorithm (Weka API)
1	weka.classifiers.bayes.NaiveBayes
2	weka.classifiers.functions.Logistic
3	weka.classifiers.functions.SimpleLogistic
4	weka.classifiers.functions.SMO
5	weka.classifiers.rules.DecisionTable
6	weka.classifiers.rules.JRip
7	weka.classifiers.rules.OneR
8	weka.classifiers.rules.PART
9	weka.classifiers.rules.ZeroR
10	weka.classifiers.trees.DecisionStump
11	weka.classifiers.trees.HoeffdingTree
12	weka.classifiers.trees.J48
13	weka.classifiers.trees.RandomForest
14	weka.classifiers.trees.RandomTree
15	weka.classifiers.trees.REPTree
16	weka.classifiers.functions.LibSVM
17	weka.classifiers.functions.MultilayerPerceptron

machine learning software Weka [32] is used. Three classification algorithms are employed upon the same input data to learn which algorithm has the best potential performance. Specifically, the algorithms used are Naive Bayes, Random Forest and Support Vector Machine, all of which are supported by Weka. Evaluation results using 10-fold cross-validation are shown in Table 3. From the table we could see that although Naive Bayes could achieve high accuracy on some users' data, the FAR and FRR remain high compared with those of the other two algorithms, which could lead to security and usability issues. Although SVM has considerable performance, it fails on some users' data. In general, Random Forest has the best performance among the three, with an average accuracy of 96.38%, an average FAR of 4.06%, and an average FRR of 3.18%.

The promising results verifies the uniqueness of the proposed trait among the given group of users, as well as the applicability of the proposed approach for implicit authentication purpose.

5. Second study: permanence of 3D finger motion pattern in multiple user–device interaction scenarios

A useful biometric trait should also remain sufficiently invariant over a period of time, thus we further conduct the second usability study to evaluate the permanence of the proposed trait.

5.1. User–device interaction scenarios

Typing constitutes only a small portion of user–device interaction gestures, we therefore take users' other gestures into consideration as well, specifically, swipe and zoom gestures. In order to test these gestures in a natural way that resembles users' daily activities, we consider three user–device interaction scenarios, namely, sentence typing, picture browsing and web surfing. The same fifteen participants from the first usability study are recruited in the second study.

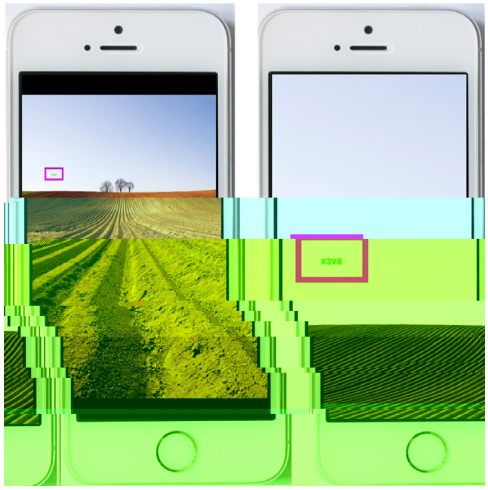

Fig. 5. Snapshot of the unzoomed and zoomed picture.

The training data collection procedure of sentence typing scenario is the same as that in the first usability study. In order to collect sensor data under the other two scenarios, we develop an image gallery app and a web surfing app, and the sensor data recording code is added into both apps. For picture browsing, twenty-six pictures are used, and every picture is watermarked with a string composed of two digits and two letters, while the position of the watermark is randomized. The initial size of the watermark is quite small that a participant can hardly recognize without a zoom in action, as shown in Fig. 5. Upon browsing, each participant is first required to zoom in the picture to the extent that she could see the watermark string clearly, then zoom out and swipe to the next picture. For web surfing, ten web page addresses are used, all of which are from well-known news websites. Upon surfing the Internet, each participant is required to read every web page in a way that resembles her daily behavior to the greatest extent.

5.2. Geometric features

In the first usability study, sixteen features are extracted, but most of them are statistical ones. By taking the magnetic field

Table 7
Experiment results considering scenarios jointly.

ID	Algorithm No.	Parameter No.	Feature Selection	Scenario	Accuracy	FAR	FRR
#1	16	92	GSCE	WS	89.36%	13.10%	8.25%
#2	6	25	GSCE	PB	84.20%	17.42%	14.25%
#3	15	75	GR	PB	83.12%	16.37%	17.36%
#4	16	84	GSCE	WS	87.17%	10.09%	15.55%
#5	17	98	GR	PB	93.16%	9.02%	4.77%
#6	4	14	GSCE	WS	87.99%	10.41%	13.60%
#7	16	81	GR	ST	73.03%	27.30%	26.64%
#8	16	89	GSCE	WS	95.18%	4.34%	5.28%
#9	1	3	GR	PB	85.13%	9.69%	19.95%
#10	14	66	GSCE	WS	85.18%	10.71%	18.79%
#11	6	26	GR	WS	86.06%	14.77%	13.13%
#12	16	92	GR	WS	98.91%	1.10%	1.08%
#13	11	45	GSCE	WS	76.51%	36.81%	10.39%
#14	16	84	GSCE	WS	68.58%	25.73%	37.05%
#15	17	100	GSCE	WS	93.31%	7.67%	5.74%
Average					85.79%	14.30%	14.12%

Table 8
Algorithm parameters used.

No.	Parameter String (Weka API)	Algorithm No.
1	``''	
2	``-D''	1
3	``-K''	
4	``''	
5	``-R 1.0E-8 -M -1 -num-decimal-places 4''	2
6	``-C -R 1.0E-8 -M -1 -num-decimal-places 4''	
7	``''	
8	``-I 0 -M 500 -H 50 -W 0.0''	3
9	``-I 0 -M 200 -H 50 -W 0.0''	
10	``-I 0 -S -M 500 -H 50 -W 0.0 –A''	
11	``''	
12	``-C 1.0 -L 0.001 -P 1.0 E-12 -N 0 -V -1 -W 1 -K \``weka.classifiers.functions.supportVector.PolyKernel -E 1.0 -C 250007\'' -calibrator \``weka.classifiers.functions.Logistic -R 1.0 E-8 -M -1 -n um-decimal-places 4\'''	
13	``-C 1.0 -L 0.001 -P 1.0 E-12 -N 1 -V -1 -W 1 -K \``weka.classifiers.functions.supportVector.PolyKernel -E 1.0 -C 250007\'' -calibrator \``weka.classifiers.functions.Logistic -R 1.0 E-8 -M -1 -n um-decimal-places 4\'''	
14	``-C 1.0 -L 0.001 -P 1.0 E-12 -N 1 -V -1 -W 1 -K \``weka.classifiers.functions.supportVector.RBFKernel -G 0.01 -C 250007\'' -calibrator \``weka.classifiers.functions.Logistic -R 1.0 E-8 -M -1 -n um-decimal-places 4\'''	4
15	``-C 1.0 -L 0.001 -P 1.0 E-12 -N 0 -M -V -1 -W 1 -K \``weka.classifiers.functions.supportVector.PolyKernel -E 1.0 -C 250007\'' -calibrator \``weka.classifiers.functions.Logistic -R 1.0 E-8 -M -1 -n um-decimal-places 4\'''	
16	``-C 1.0 -L 0.001 -P 1.0 E-12 -N 1 -M -V -1 -W 1 -K \``weka.classifiers.functions.supportVector.PolyKernel -E 1.0 -C 250007\'' -calibrator \``weka.classifiers.functions.Logistic -R 1.0 E-8 -M -1 -n um-decimal-places 4\'''	
17	``''	
18	``-X 1 -S \``weka.attributeSelection.BestFirst -D 1 -N 5\'''	
19	``-X 1 -E auc -S \``weka.attributeSelection.BestFirst -D 1 -N 5\'''	
20	``-X 1 -S \``weka.attributeSelection.GreedyStepwise -T -1.7976931348623157E308 -N -1 -num-slots 1\'''	
21	``-X 1 -S \``weka.attributeSelection.GreedyStepwise -C -T -1.7976931348623157E308 -N -1 -num-slots 1\'''	5
22	``-X 1 -E auc -S \``weka.attributeSelection.GreedyStepwise -T -1.7976931348623157E308 -N -1 -num-slots 1\'''	
23	``-X 1 -E auc -S \``weka.attributeSelection.GreedyStepwise -C -T -1.7976931348623157E308 -N -1 -num-slots 1\'''	
24	``''	
25	``-F 3 -N 2.0 -O 2 -S 1''	6
26	``-F 3 -N 2.0 -O 2 -S 1 –P''	
27	``''	
28	``-B 6''	7
29	``-B 50''	
30	``-B 120''	
31	``''	
32	``-M 2 -C 0.25 -Q 1''	
33	``-R -M 2 -N 3 -Q 1''	
34	``-M 2 -C 0.25 -Q 1 –J''	
35	``-M 2 -C 0.25 -Q 1 –doNotMakeSplitPointActualValue''	8
36	``-B -M 2 -C 0.25 -Q 1''	
37	``-M 7 -C 0.25 -Q 1''	
38	``-M 7 -C 0.25 -Q 1 –doNotMakeSplitPointActualValue''	
39	``-U -M 7 -C 0.75 -Q 1 –doNotMakeSplitPointActualValue''	
40	``-U -M 7 -C 0.75 -Q 1''	
41	``''	9
42	``''	10
43	``''	
44	``-L 2 -S 1 -E 1.0E-7 -H 0.05 -M 0.01 -G 200.0 -N 0.0''	
45	``-L 0 -S 1 -E 1.0E-7 -H 0.05 -M 0.01 -G 200.0 -N 0.0''	11
46	``-L 1 -S 1 -E 1.0E-7 -H 0.05 -M 0.01 -G 200.0 -N 0.0''	
47	``-L 0 -S 0 -E 1.0E-7 -H 0.05 -M 0.01 -G 200.0 -N 0.0''	
48	``-L 2 -S 0 -E 1.0E-7 -H 0.05 -M 0.01 -G 200.0 -N 0.0''	
49	``''	
50	``-C 0.25 -M 2''	
51	``-C 0.25 -M 2 –doNotMakeSplitPointActualValue''	
52	``-C 0.25 -M 2 –A''	12
53	``-C 0.25 -M 2 -A –doNotMakeSplitPointActualValue''	
54	``-C 0.25 -M 7 –doNotMakeSplitPointActualValue''	
55	``-C 0.25 -M 7''	

(*continued on next page*)

Table 8 (*continued*)

No.	Parameter String (Weka API)	Algorithm No.
56	``''	
57	``-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1''	
58	``-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1 –B''	
59	``-P 100 -I 100 -num-slots 1 -K 13 -M 1.0 -V 0.001 -S 1''	
60	``-P 100 -I 100 -num-slots 1 -K 13 -M 1.0 -V 0.001 -S 1 –B''	13
61	``-P 100 -I 100 -num-slots 1 -K 27 -M 1.0 -V 0.001 -S 1 –B''	
62	``-P 100 -I 100 -num-slots 1 -K 27 -M 1.0 -V 0.001 -S 1''	
63	``-P 100 -I 100 -num-slots 1 -K 51 -M 1.0 -V 0.001 -S 1 –B''	
64	``-P 100 -I 100 -num-slots 1 -K 51 -M 1.0 -V 0.001 -S 1''	
65	``''	
66	``-K 0 -M 1.0 -V 0.001 -S 1''	
67	``-K 0 -M 1.0 -V 0.001 -S 1 –B''	
68	``-K 13 -M 1.0 -V 0.001 -S 1''	
69	``-K 13 -M 1.0 -V 0.001 -S 1 –B''	14
70	``-K 27 -M 1.0 -V 0.001 -S 1''	
71	``-K 27 -M 1.0 -V 0.001 -S 1 –B''	
72	``-K 51 -M 1.0 -V 0.001 -S 1''	
73	``-K 51 -M 1.0 -V 0.001 -S 1 –B''	
74	``''	
75	``-M 2 -V 0.001 -N 3 -S 1 -L -1 -I 0.0''	15
76	``-M 2 -V 0.001 -N 3 -S 1 -L -1 -P -I 0.0''	
77	``''	
78	``-S 0 -K 2 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -seed 1''	
79	``-S 0 -K 2 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -Z -seed 1''	
80	``-S 0 -K 2 -D 3 -G 0.5 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -Z -seed 1''	
81	``-S 0 -K 2 -D 3 -G 0.5 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -seed 1''	
82	``-S 0 -K 3 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -seed 1''	
83	``-S 0 -K 3 -D 3 -G 0.5 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -seed 1''	
84	``-S 0 -K 3 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -Z -seed 1''	
85	``-S 0 -K 3 -D 3 -G 0.5 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -Z -seed 1''	16
86	``-S 0 -K 3 -D 3 -G 0.0 -R 1000.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -Z -seed 1''	
87	``-S 0 -K 3 -D 3 -G 0.0 -R -1000.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -Z -s eed 1''	
88	``-S 0 -K 1 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -seed 1''	
89	``-S 0 -K 1 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -Z -seed 1''	
90	``-S 0 -K 1 -D 3 -G 0.5 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -Z -seed 1''	
91	``-S 0 -K 1 -D 2 -G 0.5 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -Z -seed 1''	
92	``-S 0 -K 1 -D 4 -G 0.5 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -Z -seed 1''	
93	``-S 0 -K 1 -D 5 -G 0.5 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -Z -seed 1''	
94	``''	
95	``-L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a''	
96	``-L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a –I''	
97	``-L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a –D''	17
98	``-L 0.8 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a''	
99	``-L 0.15 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a''	
100	``-L0.55 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a''	
101	``-L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a –R''	

data of x, y, and z axis as points in the three-dimensional space, 21 geometric features are further extracted, such as the length of line segments, the angle between vectors, the angle between vector and plane, etc. The full list of extracted geometric features is illustrated in Table 4.

5.3. Data collection

Apart from the data logging app developed for typing scenario, two more data logging apps are designed and implemented on iOS platform for picture browsing and web surfing scenarios respectively. These apps log magnetometer data, touch sensor data and device attitude data during the user performing interaction gestures under aforementioned scenarios. The same fifteen people from the first study participated in this data collection task. For typing scenario, the data collected in the first study is used as the training data. For the remaining two scenarios, the training data logging procedure lasted for two weeks, during which every participant accomplished the data collection tasks three times for each scenario. The testing data logging procedure lasted for four weeks, and each participant accomplished the data collection task once every week under each scenario. In order to evaluate the permanence of the proposed trait, there exists a two week time separation between the training data collection task and the testing data collection task. The background magnetic field data is first logged at the beginning of each data collection session, which is the same as that in the first study.

After all data collection tasks are accomplished, the background magnetic field is canceled out using the method described in sensor data preprocessing section. Besides the statistical features as listed in the feature extraction part of the first study, geometric features are extracted as well, which are listed in Table 4. Then each participant is regarded as the legitimate user of the smart device system in turn, and the feature vectors are labeled accordingly. Counting information of training instances and testing instances for each participant under different scenarios is shown in Table 5. Please note that since different users have dissimilar interacting habits, thus the sizes of the data sets vary among participants.

5.4. Feature standardization and selection

The range of feature values may vary a lot, without a proper scaling, and algorithms using the distance between points during the learning process may not work properly. Generally speaking, standardization is one of the commonly used methods, which scales the values by calculating the z-score. The formula is as

Table 9
Filtered results of sentence typing (Information Gain Ratio)

User ID	Algorithm No.	Parameter No.	Accuracy	FAR	FRR
#1	2	6	53.69%	27.42%	64.86%
	10	42	52.61%	34.74%	59.81%
	16	85	46.74%	73.00%	33.89%
	16	89	43.60%	52.64%	60.10%
#2	4	12	71.64%	34.12%	22.66%
	13	60	66.72%	17.78%	48.64%
	16	83	43.80%	55.69%	56.70%
	16	91	72.09%	45.20%	10.79%
#3	10	41	35.23%	65.74%	63.83%
	16	82	74.85%	44.63%	5.98%
	16	83	73.33%	16.54%	36.65%
#4	10	42	49.23%	53.50%	48.08%
	16	82	60.29%	51.20%	28.35%
	16	88	62.34%	71.14%	4.55%
#5	4	15	68.45%	30.71%	32.36%
	16	79	71.41%	41.99%	15.40%
	16	80	70.02%	30.97%	29.00%
	17	97	64.76%	25.87%	44.46%
#6	5	21	48.84%	62.32%	40.10%
	8	39	62.48%	16.12%	58.74%
	12	52	63.87%	6.85%	65.16%
#7	1	2	77.80%	41.93%	2.83%
	6	25	65.71%	34.30%	34.28%
	8	38	62.35%	25.64%	49.45%
	16	81	73.03%	27.30%	26.64%
#8	5	23	51.79%	47.22%	49.18%
	16	83	57.90%	6.99%	76.52%
#9	6	26	65.87%	23.85%	44.31%
	16	88	70.49%	24.81%	34.17%
#10	5	23	47.97%	54.48%	49.63%
	6	25	70.51%	24.60%	34.31%
	8	33	72.28%	19.32%	35.99%
	15	76	68.31%	18.56%	44.62%
#11	4	12	57.25%	7.09%	77.85%
	5	23	34.34%	65.26%	66.06%
	11	47	55.57%	53.85%	35.15%
#12	5	20	51.81%	49.80%	46.61%
	14	67	64.95%	18.62%	51.23%
#13	9	41	50.38%	100.00%	0.00%
	16	82	49.60%	44.56%	56.15%
	16	83	49.74%	64.81%	35.93%
#14	16	83	69.52%	38.79%	22.31%
	16	85	38.78%	60.71%	61.71%
	16	93	58.82%	37.79%	44.52%
#15	7	28	57.29%	24.76%	60.40%
	10	42	39.14%	60.95%	60.76%
	16	84	52.19%	48.41%	47.21%

follows, where μ is the mean of the feature values, and σ is the standard deviation of the feature values.

$$x' = \frac{x - \mu}{\sigma}. \tag{11}$$

To remove the irrelevant and redundant features, a feature selection procedure is usually needed. Upon performing the feature selection procedure, the algorithm first search through a reasonable subsets of the original features, then certain evaluation criteria is applied to measure ``how good'' the feature subset is [33]. Specifically, two classification algorithm irrelevant feature selection methods are applied, of which one is based on the information gain ratio, and the other is based on the APIs that Weka provides. Through dividing the information gain by the entropy, information gain ratio could alleviate the drawback of information gain that tends to choose features with more distinct values. For the second approach, the weka.attributeSelection.GreedyStepwise searching method is used to greedily search the subsets of features, and the

Table 10
Filtered results of sentence typing (GreedyStepwise + CfsSubsetEval)

User ID	Algorithm No.	Parameter No.	Accuracy	FAR	FRR
#1	5	23	59.41%	24.62%	56.26%
	10	42	52.61%	34.74%	59.81%
	16	85	44.23%	78.49%	33.47%
#2	4	14	69.85%	13.39%	46.76%
	5	23	58.09%	42.06%	41.76%
	8	39	77.31%	15.58%	29.74%
#3	10	42	35.23%	65.74%	63.83%
	16	83	53.88%	45.01%	47.22%
#4	10	42	49.23%	53.50%	48.08%
	16	83	71.82%	46.12%	10.43%
#5	4	14	80.52%	23.21%	15.81%
	16	78	70.20%	21.22%	38.23%
	16	93	66.84%	30.61%	35.67%
#6	5	15	49.02%	67.71%	34.38%
	12	52	57.40%	21.18%	63.84%
	16	81	57.07%	64.04%	22.00%
#7	8	33	65.59%	32.58%	36.20%
	10	42	71.94%	56.31%	0.34%
	13	59	70.63%	9.98%	48.39%
	16	80	59.20%	39.77%	41.81%
#8	5	21	51.27%	47.48%	49.95%
	16	85	56.55%	41.33%	45.52%
	16	88	61.57%	23.74%	52.83%
#9	16	85	52.38%	45.70%	49.52%
	16	88	67.36%	43.45%	21.94%
#10	2	6	62.95%	29.93%	44.08%
	16	78	63.85%	30.89%	41.35%
	17	97	62.25%	37.53%	37.98%
#11	9	41	50.40%	100.00%	0.00%
	16	85	48.08%	50.67%	53.16%
	16	90	58.55%	20.35%	62.22%
#12	16	85	61.15%	45.26%	32.53%
#13	17	100	51.69%	49.81%	46.83%
	17	101	54.04%	42.72%	49.16%
#14	2	6	63.48%	17.86%	54.90%
	4	16	62.61%	20.52%	54.01%
	9	41	50.38%	100.00%	0.00%
	16	82	56.01%	27.94%	59.80%
#15	5	21	59.88%	59.42%	21.10%
	16	79	58.13%	39.10%	44.61%
	16	84	54.79%	47.09%	43.36%
	16	90	57.60%	36.51%	48.20%

weka.attributeSelection.CfsSubsetEval evaluation method is used to evaluate the worth of subsets of features for classification purpose. Since the selected feature subset using the second approach contains about 20 features on average, when applying the information gain ratio approach, 20 features with the highest information gain ratio are selected.

5.5. Experimental evaluation

To better test the applicability of using the proposed trait for user authentication purpose, more classification algorithms are taken into consideration in this usability study, specifically, seventeen algorithms are used in this experiment, which are listed in Table 6. Different algorithms may have various numbers of parameters, and some parameters have continuous possible value range, so it is infeasible to exhaustively seek best values for the parameters. A list of possible values for the parameters are empirically determined and used in the experiment, which is shown in Table 8 in Appendix.

The experiment is conducted using Weka APIs, and aforementioned algorithms and parameters are applied on the training and

Table 11
Filtered results of picture browsing (Information Gain Ratio)

User ID	Algorithm No.	Parameter No.	Accuracy	FARAccuracy

Table 13
Filtered results of web surfing (Information Gain Ratio)

User ID	Algorithm No.	Parameter No.	Accuracy	FAR	FRR
#1	2	6	81.59%	18.52%	18.30%
	11	48	69.58%	11.64%	48.71%
#2	16	80	67.20%	16.09%	49.16%
	16	84	72.59%	33.28%	21.66%
	16	88	78.11%	41.44%	2.76%
	16	89	67.08%	29.59%	36.18%
#3	3	9	72.96%	30.40%	23.71%
	3	10	72.30%	29.51%	25.91%
	8	33	63.09%	27.85%	45.88%
	8	35	67.31%	32.00%	33.37%
#4	10	42	79.68%	37.19%	3.62%
	16	82	70.02%	29.93%	30.02%
#5	16	83	78.21%	37.62%	5.99%
	16	88	56.04%	45.43%	42.49%
#6	6	26	73.49%	14.46%	38.49%
	11	47	81.08%	25.90%	11.98%
	12	52	78.08%	20.14%	23.70%
	17	97	72.24%	27.71%	27.80%
#7	7	30	54.94%	46.70%	43.45%
	12	54	66.67%	16.48%	49.79%
	16	92	76.17%	31.98%	15.88%
#8	8	33	74.10%	2.52%	48.51%
	16	84	93.80%	5.18%	7.18%
#9	4	12	76.04%	22.82%	25.09%
	4	16	69.39%	15.33%	45.63%
	16	84	81.00%	35.45%	2.83%
#10	7	30	82.21%	25.25%	10.58%
	15	75	75.98%	7.88%	39.60%
	16	80	73.08%	26.48%	27.35%
	17	100	79.49%	19.09%	21.88%
#11	6	26	86.06%	14.77%	13.13%
	12	55	73.08%	4.31%	48.91%
#12	16	81	80.25%	0.00%	39.07%
	16	92	98.91%	1.10%	1.08%
#13	5	20	67.67%	13.87%	50.47%
	5	21	67.49%	14.42%	50.27%
	16	82	53.44%	34.62%	58.30%
	16	85	65.90%	66.69%	2.09%
#14	16	79	59.52%	31.60%	49.27%
	16	84	63.22%	35.32%	38.22%
	16	85	63.52%	73.29%	0.00%
	16	89	38.33%	61.45%	61.88%
#15					

possibility of the proposed 3D magnetic finger motion pattern being applied for implicit authentication. It is encouraging that the experiment results considering three different interaction scenarios show an average accuracy rate of above 80%, together with average FAR and FRR of below 15%.

We would delve further into the causes and countermeasures of false acceptances and false rejections targeting a larger group of users so that an approach of real-world application could be obtained. It cannot be denied that the necessity of a magnetic ring sets limits on wide range deployment. Nevertheless, as more and more mobile devices have been equipped with certain kind of sensors that could track the finger motion in a three-dimensional way, the problem would soon be resolved. Furthermore, since authentication and authorization issues have not been studied in depth in the context of cyber–physical systems [34], our future work will expand on the application of implicit authentication techniques for device identification purpose.

Acknowledgments

This work was supported by the National Key R&D Program of China (No. 2017YFB1003000), National Natural Science Foundation of China (Nos. 61572130, 61502100, 61532013, 61632008 and 61320106007), Jiangsu Provincial Natural Science Foundation of China (No. BK20150637), Qing Lan Project of Jiangsu Province, China, Jiangsu Provincial Key Laboratory of Network and Information Security, China (No. BM2003201), and Key Laboratory of Computer Network and Information Integration of Ministry of Education of China (No. 93K-9).

Appendix

See Tables 8–14.

References

[1] Y. Wang, S. Wen, Y. Xiang, W. Zhou, Modelling the propagation of worms in networks: A survey, IEEE Commun. Surv. Tutor. (2014) 942–960.
[2] S. Wen, W. Zhou, J. Zhang, Y. Xiang, W. Zhou, W. Jia, Modeling propagation dynamics of social network worms, IEEE Trans. Parallel Distrib. Syst. 24 (8) (2013) 1633–1643.
[3] J.J. Jiang, S. Wen, S. Yu, Y. Xiang, W. Zhou, Identifying propagation sources in networks: State-of-the-art and comparative studies, IEEE Commun. Surv. Tutor. 19 (1) (2017) 465–481.
[4] A.J. Aviv, K. Gibson, E. Mossop, M. Blaze, J.M. Smith, Smudge attacks on smartphone touch screens, in: Proc. of the 4th USENIX Workshop on Offensive Technologies, Washington, DC, 2010.
[5] L. Cai, H. Chen, TouchLogger: inferring keystrokes on touch screen from smartphone motion, in: Proc. of the 6th USENIX Workshop on Hot Topics in Security, San Francisco, CA, 2011.
[6] Q. Yue, Z. Ling, X. Fu, et al., Blind recognition of touched keys on mobile devices, in: Proc. of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, 2014, pp. 1403–1414.
[7] X. Pan, Z. Ling, A. Pingley, et al., Password extraction via reconstructed wireless mouse trajectory, IEEE Trans. Dependable Secure Comput. 13 (2016) 461–473.
[8] Y. Zhang, P. Xia, J. Luo, et al., Fingerprint attack against touch-enabled devices, in: Proc. of the 2nd Workshop on Security and Privacy in Smartphones and Mobile Devices, Raleigh, NC, 2012, pp. 57–68.
[9] Z. Ling, J. Luo, Q. Chen, et al., Secure fingertip mouse for mobile devices, in: Proc. of the 35th IEEE International Conference on Computer Communications, San Francisco, CA, 2016, pp. 343–351.
[10]

[32] M. Hall, E. Frank, G. Holmes, et al., The WEKA data mining software: an update, ACM SIGKDD Explor. Newsl. 11 (2009) 10-18.
[33] S.B. Kotsiantis, D. Kanellopoulos, P.E. Pintelas, Data preprocessing for supervised leaning, Int. J. Comput. Sci. 1 (2006) 111–117.
[34] S. Ivan, S. Wen, X. Huang, H. Luan, An overview of fog computing and its security issues, Concurr. Comput.: Pract. Exper. 28 (10) (2015) 2991–3005.

Yiting Zhang is currently working toward the Ph.D. degree in Computer Science at Southeast University, China. Her research interests include network security and privacy.

Ming Yang received the Ph.D. degree in computer science from Southeast University, China, in 2007. Currently, he is an associate professor at the School of Computer Science and Engineering in Southeast University, Nanjing, China. His research interests include network security and privacy. Dr. Yang is a member of CCF and ACM, as well as Deputy Director of Key Laboratory of Computer Network and Information Integration, Ministry of Education of China.

Zhen Ling received the B.S. degree (2005) and Ph.D. degree (2014) in Computer Science from Nanjing Institute of Technology, China and Southeast University, China, respectively. He is an associate professor in the School of Computer Science and Engineering, Southeast University, Nanjing, China. He won ACM China Doctoral Dissertation Award and China Computer Federation (CCF) Doctoral Dissertation Award, in 2014 and 2015, respectively. His research interests include network security, privacy, and Internet of Things.

Yaowen Liu is currently working toward the M.S. degree in Computer Science at Southeast University, China. His research interests include network security and privacy.

Wenjia Wu received the B.S. and Ph.D. degrees in computer science in 2006 and 2013, respectively, from Southeast University. He is an associate professor at the School of Computer Science and Engineering in Southeast University. His research interests include wireless and mobile networks.