TouchSense: Accurate and Transparent User Re-authentication via Finger Touching

Chong Zhang, Songfan Li, Yihang Song, Li Lu(✉), and Mengshu Hou

University of Electronic Science and Technology of China,
Chengdu 611731, Sichuan, China
{zhangchong,sfli,songyihang}@std.uestc.edu.cn
{luli2009,mshou}@uestc.edu.cn

Abstract. Re-authentication identifies the user during the whole usage to enhance the security of smartphones. To avoid frequent interrupts to users, user features should be imperceptibly collected for identification without user assistance. Conventionally, behavior habits (*e.g.* movement, trail) during the user operation are commonly considered as the most appropriate features for re-authentication. The behavior features, however, are often fluctuating and inevitably sacrifice the accuracy of re-authentication, which puts the phones at risk increasingly. In this paper, we propose *TouchSense*, an accurate and transparent scheme for user re-authentication. The basic idea is to leverage the combined information of human biometric capacitance and touching behavior for user identification. When the user touches capacitive-based sensors, both information can be automatically collected and applied in the authentication, which is transparent to the user. Based on the authentication results, we build up user-legitimate models to comprehensively evaluate the user's legitimacy, which reduces misjudgment and further improves accuracy. Moreover, we implement *TouchSense* on an SX9310 EVKA board and conduct comprehensive experiments to evaluate it. The results illustrate that *TouchSense* can identify 98% intruders within 10 s, but for legitimate users, the misjudgment is less than 0.9% in 2.6-hours-usage.

Keywords: User re-authentication · User-transparent · Touching behavior · Biometric capacitance · Continuous security

1 Introduction

Over the past decades, the dramatic outpouring of digital information has generated a mass of invaluable data stored in computer phones. One of the crucial concerns lies in the private data that is sensitive to be accessed by illegal users. To prevent this, user authentication schemes [3,35,41] are proposed to recognize who is operating the device at several critical operations (*e.g.* unlocking,

Supported by University of Electronic Science and Technology of China.

H. Jiang et al. (Eds.): ICECI 2020, LNICST 368, pp. 105–125, 2021.
https://doi.org/10.1007/978-3-030-73429-9_7

paying). These schemes, however, are vulnerable to attacks in that attackers can tamper with the private data between two authentications or even steal the keys (*e.g.* password, fingerprint) to pass the authentication [28,40]. Therefore, in order to enhance security, re-authentication schemes are desired to identify the user during the whole usage.

At first glance, user re-authentication may be a simple repeat of conventional authentications. However, many approaches for authentication are inadequate to re-authentication as they require user assistance and interrupt fluent operations. For instance, users may suffer from the bother of frequently checking their fingerprints for security. Therefore, we desire to find out a way to authenticate the user transparently from their operations. To this goal, a typical approach may exploit video streaming for continuous face recognition [4]. This approach, however, becomes insecure in that recent reports [24] have proofed that 3D face masks can fool facial recognition at airports.

To offer a guarantee to the privacy data, several works have been proposed for user-transparent re-authentication. Lingjun *et al.* [19] presents an on-screen gestures monitoring system to identify users according to screen operations including basic sliding and tapping. Further, the authors in [21,34] provide approaches for continuous authentication using movement data measured on smartphones. As a consequence, both gestures and movements belong to human habits built from individuals' daily life. Although habit features show potentials to be utilized in user authentication, they hardly achieve high accuracy in practice as the behaviors performed from habits are often fluctuating and mutable. Specifically, users cannot perform completely the same actions every time, even the same user repeats the same motion (*e.g.* tap) [10]. Besides, those works may also be affected by different Apps (Application software) and achieve lower accuracy in practice. For instance, some Apps require users to long-press the screen to copy text, which changes the user's behavior and may cause misjudgment in the authentication.

In this paper, we propose *TouchSense*, an accurate and transparent scheme for user re-authentication. Our basic idea originates an observation that past literature [23] has proofed that biometric capacitance of human bodies contains a unique feature for individuals recognition. Specifically, the biometric capacitance stands for the capability of the human body to store charges, which is proportional to the number of cells and size of cell mass in the body, thus different for everyone. Upon this, *TouchSense* collects the user's biometric capacitance via finger touching and imperceptibly authenticates users in the whole duration of human-machine interaction (HMI). Compared to habit features, biometric capacitance shows better accuracy for two reasons. First, the biometric feature is more stable than behaviors as it is determined by the user's physical characteristics and will not change in a short time [7,16,36]. Second, biometric features are not affected by different Apps as user use.

Although the basic idea seems effective, we have to address a fundamental challenge in *TouchSense*. Off-the-shelf capacitive touch screens [1,13,14,29,30] detects the touch by sensing the biometric capacitance during user touching, but

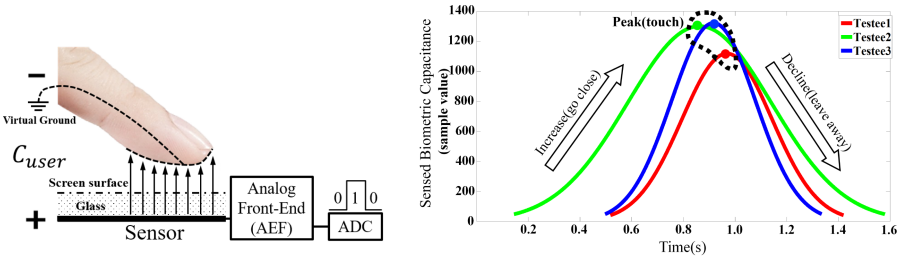


Fig. 1. Upon the capacitive sensors in a touchscreen, *TouchSense* continuously authenticates users according to their biometric capacitance during finger touching.

the accuracy is not adequate to achieve user authentication. The reason mainly stems from the fact that the accurate measurement of biometric capacitance requires the human standing with bare feet on a Styrofoam plate in a Faraday shield [11,18]. In practice, it is unrealistic to prepare such an experimental environment to measure the biometric capacitance.

We tackle this challenge by combining the biometric capacitance and touching behavior of the user to improve the accuracy. We show that both parameters can be obtained simultaneously by the touchscreen, as shown in Fig. 1. The peak value represents the biometric capacitance, while the rate of curve change is related to the user's operation habit (Fig. 1(b)). Specifically, when the user is going to touch the screen, the finger and the sensor will form a capacitor (C_{User}) in which the capacitance depends on both human biometric capacitance and the distance between the finger and the sensor under screen (Fig. 1(a)). Further, once the finger is touched on the screen surface, the distance can be treated as a constant value and thus the measurement result is only related to the biometric capacitance for different users. In addition, the finger movement speed results in the change to the distance and finally implies the user habits to touch the screen.

We also design an algorithm to avoid the impact of different Apps on user behaviors. The algorithm can distinguish and obtain the rising and falling edge of the sensed biometric capacitance which related to user behavior and exclude the holding period (finger keep on screen) which may be affected by different APPs. Finally, we build up a user-legitimate model to comprehensively evaluate the user's legitimacy in usage. This model accumulates the authentication results of each touching operation of the user and generates a legitimacy score, which increases with legal operations and decreases with illegal operations. Once the score drops below the threshold, the user will be considered an attacker and logged out to avoid further operations.

To show the feasibility of our design, we implement *TouchSense* on an SX9310 EVKA board [37] and evaluate it for 50 volunteers. To comprehensively evaluate the performance of *TouchSense*, we utilize the collected data set from 50 users and did mass simulations on 100,000 samples of the attackers and legitimate

users. For each sample, the simulation is up to 50,000,000 times. The results show an interesting result that *TouchSense* can identify 98% attackers within 6 s (10 operations) and logout all of them within 11 s (18 operations). However, for legitimate users, the misjudgment is only 0.321% in 16-minutes-usage (1,000 touching operations), and only 0.895% in 2.6-hours-usage (10,000 touching operations).

Contributions. We propose *TouchSense*, a novel secure scheme for accurate and transparent user re-authentication. *TouchSense* leverages human biometric capacitance to achieve imperceptible user re-authentication. Further, we combine the biometric capacitance and touching habits to enhance accuracy. We also eliminate the impact of APPs on user behavior to ensure the robustness of *TouchSense* in practical. Finally, we build up a user-legitimate model to describe the user's legality much more accurately in using.

The remainder of the paper is organized as follows. Section 2 discusses the background. Section 3 introduces our scheme design. Next, Sect. 4 discusses the evaluation setup and test results. Section 5 discusses future works and ethical concerns of this paper. Finally, we make a summary in Sect. 6.

2 Background

The widespread use of smartphones has not only enriched our lives but also raise new issues of security and privacy concerns. For example, smartphones are no longer just communication tools, but become as powerful as computers, which may store many personal information (*e.g.* location, account, photos, shopping preferences) in it. Hence, smartphones become more and more private and build up a relationship with users. Having a victim's private information, the attacker can launch an impersonation attack [38]. Such attacks could threaten the owner's property and even reputation security. Therefore, protecting private information on smartphones is a key issue that has to be settled urgently.

To protect the user's privacy, most smartphones have deployed conventional authentications schemes to unlock the devices, such as passwords, fingerprints, facial recognition, palm textures [43] and signatures [2,26,27] which authenticate users when they are logging in. However, the device is still vulnerable to attackers for the remainder of the session. Specifically, an attacker can easily access the privacy information if the owner forgets to lock the device and loses it in public places. Even the device is locked, the attacker can also leverage system flaws to circumvent the lock screen, which is reported to exist in both IOS [5] and Android [17] systems. Hence, the continuous protection provided by re-authentication is necessary for smartphones, as it will repetitively verify the current user's legitimacy during the whole system execution.

As re-authentication schemes will keep running in the background as long as the user operating the phones, it should target not only accuracy but also to be user-transparent which can automatically run without user assistance. Previous re-authentication schemes are mainly base on the user's behavior or biometric

information. However, none of them satisfies both aforementioned requirements simultaneously.

2.1 Behavior-Based User Re-authentication

Behavior information is the process of body motion, which contains the habit information of the user [25] and can be transparently sensed during it's process (*e.g.* gait [12] can be sensed when the user is walking). Hence, many re-authentication works are concentrate on it. Zijiang *et al.* made an efficient user re-authentication scheme which bases on keystrokes [15]. This scheme can verify the user's identity transparently by their typing behavior when the user is using the on-screen keyboard, but the accuracy is inadequate (90% at 20 typing operations) for high-security requirements, and it can only work when the user is typing. Lingjun *et al.* designed another scheme [19] to verify the user by the on-screen operation gestures. The same as Zijiang's work, it achieves the advantages of user-transparency but also not accurate enough. There are also some other works using other behavior information for user re-authentication. Such as gait-based schemes [8,31,33,39,44], which can work when the user is walking, but in practical, walking may not the most state of users.

2.2 Biometric-Based User Re-authentication

Biometric information is the reflection of our body characteristics, just like the internal passwords of us, which is good at stability and uniqueness [6]. However, biometric information doesn't contain any motion, which has to be sensed under user assistance (*e.g.* touching fingerprint panel). Hence, biometric-based schemes may interrupt user operations and bother users during the re-authentication process. There are relatively little works study on it.

David *et al.* designed an excellent work, which leverages the user's facial information for re-authentication [4]. This scheme uses the front camera to continuously verify the current user's legitimacy, which is accurate and easy to build on the existing phones. However, this work needs the user to keep facing the front camera when using, which constrains user-behavior in practical. Moreover, this scheme may be fooled by a 3D mask [24], which is insecure under such attacks. Feng *et al.* design another work that continuously authenticates the user by voice [9], but the user has to talk to assist the re-authentication, which influences user experience.

In summary, both existing biometric or behavior based re-authentication schemes can only satisfy either accuracy or transparency. To meet both requirements simultaneously, *TouchSense* should combine both biometric and behavior information in the authentication. As biometric information has to be sensed under user motion, the solution is to find a biometric whose necessary extraction step happens to be the user's device operation behavior. In this way, it can be extracted during the user's operation transparently without any additional auxiliary actions, thus avoid disturbing the user. In this paper, we start from the user's finger touching operation and combines the associated biometric

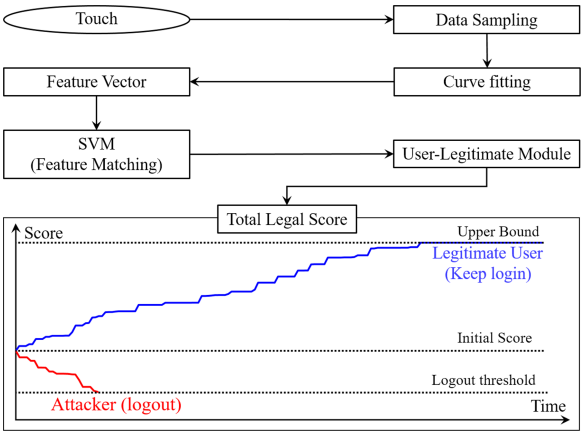


Fig. 2. *TouchSense* authenticates the user on every screen-touching operation and accumulates the results as the user's total legal score. If the score drops below the threshold, *TouchSense* will lock the device to protect private data on it.

information (biometric capacitance) and behavior information (touching behavior) to build up the re-authentication. Since the user needs to operate the device through finger touching, *TouchSense* can continuously work during the whole usage.

3 Scheme Design

The system design of *TouchSense* is shown in Fig. 2. When the user touches the screen, *TouchSense* scans the sensor and obtains the capacitance data. Then *TouchSense* fits the curve base on the sampled data and generate the user's feature vector according to the parameters of the curve function. After that, *TouchSense* compares the generated feature vector with the stored database to authenticate the user through Support Vector Machine (SVM). Finally, *TouchSense* builds up the user-legitimate module base on the authentication results during usage and generates the user's legal score. The score dynamically updates during user operation, which rises with legal operations or drops with illegal operations. If the score drops below the logout threshold, *TouchSense* logs out the user as an attacker, and locks the smartphones to avoid the user's further operation.

3.1 Data Sampling

The first step of *TouchSense* is data sampling. *TouchSense* periodically scanning the sensor during user operation to obtain data for the authentication. To eliminate the impact of different APPs on user behavior, we design a novel algorithm to combine the rising and falling edge together, and remove the holding period,

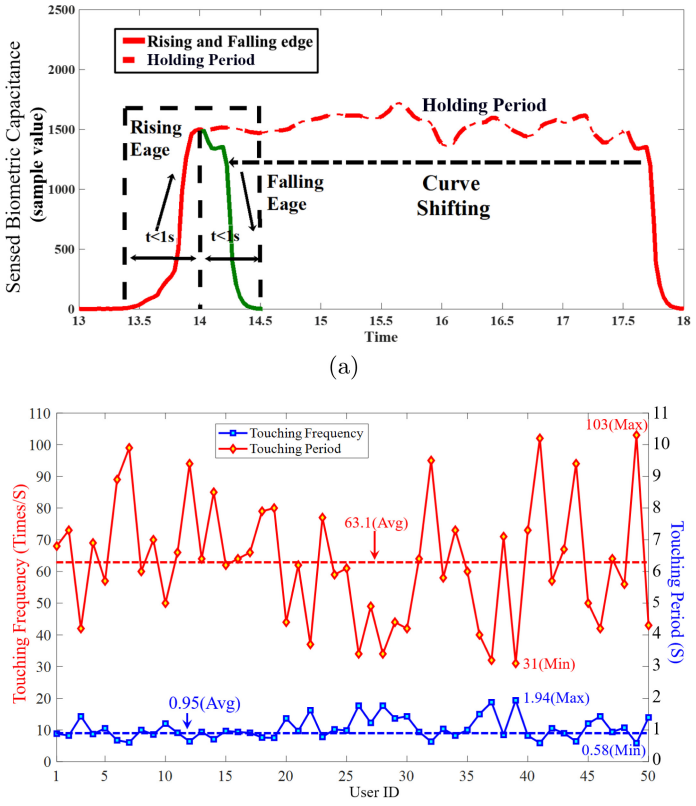


Fig. 3. (a) We only extract the rising and falling part of the sensed data and removing the holding period to avoid influence from different Apps. (b) Ignore the holding period, the touch frequency of 50 testers is 31–103 times per minute, and the period of each touch is 0.58–1.94 s.

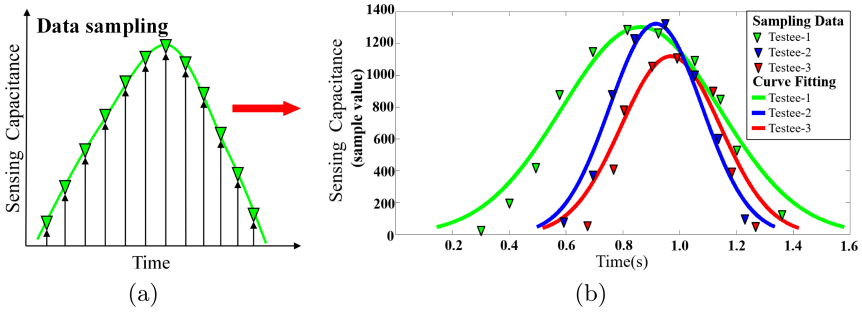


Fig. 4. Data sampling (a) and curve fitting (b) for 3 testers.

Algorithm 1. BIOMETRIC-CAPACITANCE-EXTRACT

```
▷ The variable initialization
New File String
New Array P[length][3]
time ← 0, value ← 0, trend 50 ← 0, trendflag ← 0
▷ The data processing
for String.readLine()! = null do
    timenext ← String[0]
    valuenext ← String[1]
    if |timenext − time| > timethreshold then
        Continue
    else
        trendflag ← 1
        P ← timenext, valuenext, trendflag
        Continue
    end if
    if valuenext − value > 0 and trend 50 == 0 then
        trendflag = 0
        P ← timenext, valuenext, trendflag
        Continue
    end if
end for
▷ Calculate the value
for for i ← 0 to length[P] do
    if flag ← P[i][3] == 1 then
        count ← count + 1
        time ← time + P[i][0]
        value ← value + P[i][1]
    end if
    timeavg ← P[i][0]/count
    valueavg ← P[i][1]/count
    Quick − Sort(P)
end for
```

as shown in Fig. 3(a). Specifically, the holding period might be influenced by specific APPs (*e.g.* sliding, long pressing, have longer holding period than tapping), which may lead to misjudgment. This algorithm can identify the trend of the sensed biometric capacitance. If the value does not change significantly over a period of time, it will be defined as the holding period and deleted, and the portions that rise rapidly (finger go close to the screen) and fall (finger move away from the screen) will be recorded, as shown in Algorithm 1 (page 9). To set the appropriate length of the edge decision window, we analyzed the operating habits of 50 users, as shown in Fig. 3(b). Among 50 users, the operating frequency is 31–103 times, and the duration of each operation is 0.58–1.94 s. Hence, we set 1 s as the length of the decision widow to obtain the rising and falling edge of the curve (2 s in total).

3.2 Curve Fitting

After the data sampling, *TouchSense* gets the discrete capacitance value, shown in Fig. 4(a). These discrete points contain both biometric capacitance and touching behavior information of the user. The increasing or decreasing of the value depends on the user's touching behavior (*e.g.* finger moving speed, tapping frequency) and the peak value represents the biometric capacitance In this step, *TouchSense* uses the Gaussian function to fitting the curve base on these sampled data, and we can see the obvious differences between different users, shown in Fig. 4(b). The Gaussian function has three coefficients: (1) the peak value of the curve: a, (2) the abscissa value in the center of the curve: b, (3) the half-width of the curve: c. *TouchSense* utilizes these three coefficients and builds up a three-dimensional feature vector for the user, which represents the user's identity under the current finger touching. The formula of the Gaussian function is shown below.

$$f(x) = ae^{-\frac{(x-b)^2}{2c^2}} \tag{1}$$

3.3 Feature Vector Extraction

To obtain the accurate value for user feature vector, we need to solve Eq. (1) and get the optimal value of those three coefficients, which minimizes the variance between the fitted curve and sampled value.

We use logarithm and simplify it as follows:

$$ln(y_i) = \left\{ ln(a) - \frac{b^2}{2c^2} \right\} + \frac{2x_i b}{2c^2} - \frac{x_i^2}{2c^2} \tag{2}$$

Let $ln(y_i) = Z_i$, $ln(a) - \frac{b^2}{2c^2} = b_0$, $\frac{2b}{2c^2} = b_1$, $-\frac{1}{2c^2} = b_2$, and bring these equations into Eq. 2 to get the fitting function:

$$Z_i = b_0 + b_1 x_i + b_2 x_i^2 = \begin{pmatrix} 1 & x_i & x_i^2 \end{pmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} \tag{3}$$

The sampled capacitance from the user's single operation is defined as

$$[X, Y] = [(x_1, y_1), (x_2, y_2) \cdots (x_i, y_i) \cdots] \tag{4}$$

Where x_i is the sampling time of the sensed biometric capacitance and y_i is the sampled value. We import all the sampled data into function (3) and get the following parameter matrix:

$$\begin{bmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix} \tag{5}$$

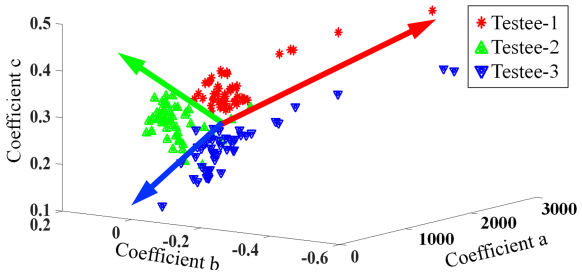


Fig. 5. Feature vector sets for three users. The coefficient a is the peak value of the Gaussian curve, b is the abscissa value at the center of the Gaussian curve, c is the half-width of the Gaussian curve.

Simplified as:

$$Z_{n\times 1} = X_{n\times 3} B_{3\times 1} + E_{n\times 1} \tag{6}$$

To minimize the sum of squared errors of the calculation results, we can find the least squares solution of the B matrix according to the principle of least squares:

$$B = \left(X^T X\right)^{-1} X^T Z = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} ln(a) - \frac{b^2}{2c^2} \\ \frac{2b}{2c^2} \\ -\frac{1}{2c^2} \end{bmatrix} \tag{7}$$

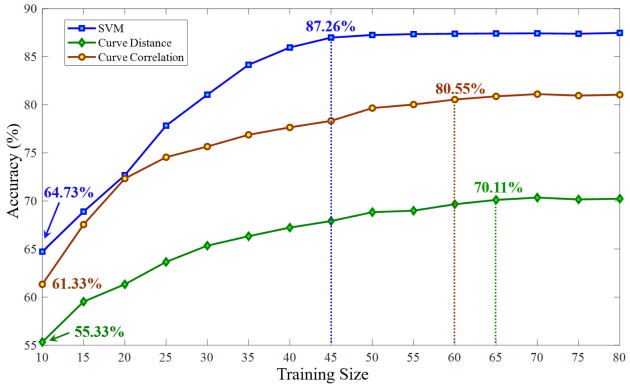
By further solving the equation, we can get the optimal value of those three parameters (a, b, c), and define it as the user's three-dimensional feature vector for authentication.

To verify the rationality of feature vectors, we trained the data from 50 testers, and each tester has 80 groups of data tests to build their own feature vector set. Among them, the data sets of three users are shown in Fig. 5. We can see that the feature vectors for the different users are merged in different areas, which makes it possible to classify different users.

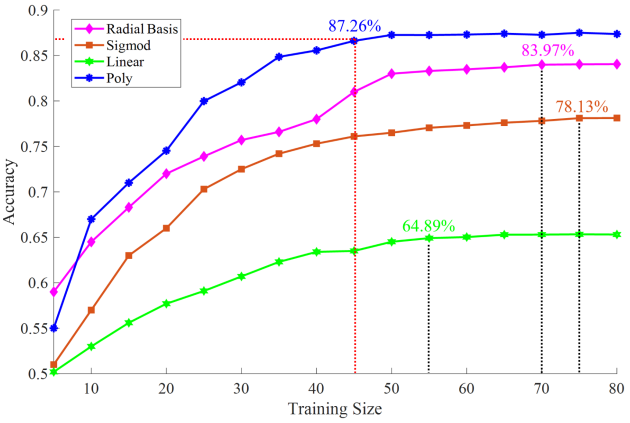
3.4 Feature Matching (Authentication)

In this step, *TouchSense* will compare the generated feature vector with the stored data to verify the current user's legality. To find a suitable algorithm for user authentication in *TouchSense*, we test common classification and matching algorithms, including curve distance difference, curve correlation and Support Vector Machine (SVM), and found that SVM results best, as shown in Fig. 6(a)

The SVM is a supervised learning model which has associated learning algorithms to analyze the data for classification. It needs a kernel function to work, and different kernel functions will bring different performance. We conduct many



(a)



(b)

Fig. 6. The authentication accuracy with different classification algorithm (a) and SVM results best with polynomial kernel (Poly) (b)

experiments and finally find that the polynomial kernel achieves the best results, as shown in Fig. 6(b). The equation of polynomial kernel is shown below.

$$K\{x, y\} = \left(ax^T y + c\right)^d$$

x means the abscissa values, y means the ordinate value, T represents the matrix transpose, a, c and d is the artificial constants.

We can see that the performance of Polynomial Kernel gradually stable after 45 groups of data training. Hence, for better user experience, *TouchSense* only needs to collect 45 groups of data as the user's data set. By this configuration, *TouchSense* achieves 87.26% accuracy and provides 16.12% FAR (False Accept Rate) and 9.36% FRR (False Reject Rate) in a single finger touching.

Algorithm 2. User-Legitimate Module

Input : Authentication result of each finger touching operation.
Output : User's legitimacy score and system decision (Keep user login or lock screen).

Start
$Score = 100$
Operation times : $i = 0$
for $Score \geq 0$ **do**
  $i++$
  if *Authentication result*$[i] = illegal$ **then**
    $Score = Score -$ *plenty points* X
    if $(i \geq 2)$ **and** (*Authentication result*$[i-1] = illegal$) **then**
      $Score = Score -$ *extra plenty points* E
    end if
  else
    $Score = Score +$ *correction points* Y
  end if
  if $Score >$ *upper bound* S **then**
    $Score =$ *upper bound* S
  end if
  if $Score < 0$ (*Threshold*) **then**
    Log the user out (*Lock the screen*)
  end if
end for

3.5 User-Legitimate Model

Finally, *TouchSense* builds up the user-legitimate model to comprehensively evaluate the user's legitimacy. This model converts the authentication results of each finger touching into corresponding scores and accumulates it after the user logging in. The total score can effectively represent the user's legitimacy, and it increases if the result is legal, or decrease if the result is illegal. *TouchSense* works in the background without bothering the user until his score drops below the threshold. Then *TouchSense* regards the user as an attacker and locks the screen to ensure device security. Hence, we can set suitable parameters to quickly logout intruders while providing certain fault tolerance for legitimate users. By this, *TouchSense* can rapidly reduce misjudgment and improve accuracy, as shown in Algorithm 2.

This algorithm has four parameters: penalty points X for illegal operation, correction points Y for legal operation, extra penalty points E for continuously illegal operation, and upper bound S. When logging in, the user will get an initial score: 100, and the score will decrease by X points for illegal operation or increase by Y points for legal operation. For continuously illegal operation, the user will deduct extra penalty points E. After the user's i-th operation, the expectation of the legitimacy score can be expressed as

$$S(i) = MAX(100 - iPX + i(1-P)Y - (i-1)P^2E, S) \tag{8}$$

$S(i)$ is the expectation of user score after the i-th operation, and P is the average probability for the user to be judged illegal in each operation. If $S(i)$ drops below the threshold, *TouchSense* will lock the system and stop the user's $(i+1)$ th operation. We did mass calculation and find the optimal values for those four parameters: $(X, Y, S, E) = (25, 67, 568, 40)$. By this configuration, 59% (calculated by $(1 - FAR)^3$) of the attackers will be logged out within three operations, and most legitimate users will increase their score to the upper bound within ten operations, which ensures the security while hardly bother the legitimate users.

4 Evaluation Results

To evaluate the feasibility of *TouchSense*, we built an experimental platform on the SX9310 evaluation board and did comprehensive experiments and simulations base on the data collected from 50 users.

4.1 Experimental Platform Setup

To ensure the validity of experiments, we need to choose the appropriate hardware to deploy *TouchSense*. Capacitive-screens are vital interfaces for users to interact with smartphones, which relying on biometric capacitance to detect the touch. However, we can not directly deploy *TouchSense* on existing capacitive-screens because the screen can not obtain the accurate value of the human body capacitance, which need for authentication. Specifically, capacitive-screens are aimed at the localization but not authentication. To precisely detect the touching coordinate, most capacitive screens are based on Projected Capacitive Touch (PCT) technologies[13,22,29], which comprised of millions of micro capacitors by the mixed array [1,30,42], but only use a single threshold ADC to sense the touch. When the user touches the screen, the finger forms an inductive

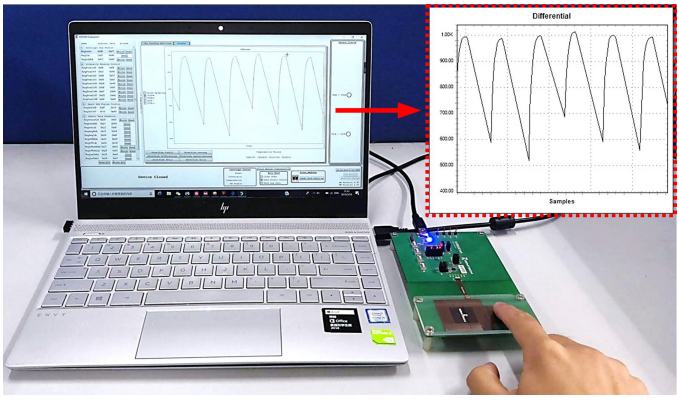


Fig. 7. The test platform for *TouchSense*. We use a laptop to observe the working states of our scheme. The sensed biometric capacitance is shown in the window real-timely.

capacitance with the touch sensor below the screen, which values proportional to the user's biometric capacitance and inversely proportional to the distance between the finger and the sensor. The touch takes effect if the value exceeds the threshold. So it can precisely detect the location where the user touches but fail to obtain the specific value of biometric capacitance. Hence, we need to find a suitable device embedded with a sensitive sensor and high-resolution ADC to deploy *TouchSense*.

We find the chip SX9310, which has up to $0.08fF$ ($1fF = 10^{-15}F$) resolution for capacitive sensing [37]. We use the evaluation board as the hardware to deploy *TouchSense*, and carry out experiments to verify the feasibility. This board has an SX9310 chip to gather the data and an MSP430F2132 chip to analyze the results. We can also use a laptop to monitor the working status of *TouchSense*, shown in Fig. 7.

Table 1. Biometric capacitance extract accuracy

UI	Gender	Temperature	Motion	Actual motions	Valid detection	Missed detection
1	Male	High (30 °C)	Tapping	48	48	0
9	Female	Low (10 °C)	Holding	50	50	0
12	Male	High (30 °C)	Holding	45	45	0
26	Female	Low (10 °C)	Tapping	48	46	2
39	Male	High (30 °C)	Tapping	44	44	0
42	Female	Low (10 °C)	Holding	46	46	0
Biometric capacitance detection accuracy of 50 users: 99.2%						

Legend: UI = User ID

We test the platform for 50 users, including tapping and holding, and find that the extraction accuracy of the biometric capacitance is 99.2%; among them, the test results of 6 users are shown in Table 1.

Tapping and holding represent user operation for different Apps. Tapping fits most Apps in which the user tap the screen with negligible holding time. Holding represents some special APPs, which require users to long-press the screen, such as copy text or screenshots. Correspondingly, the finger stays on the screen for a long time, and the trend of the sensed biometric capacitance is completely different. The results show *TouchSense* achieves good robustness in different user APPs.

Table 2. Cross-validation for 50 users

UI	S													
	UI													
	1	2	3	4	5	6		45	46	47	48	49	50	FIR (%)
1	**0.38**	0.12	0.15	0.07	0.12	0.19	...	0.08	0.14	0.11	0.16	0.12	0.18	0
2	0.12	**0.41**	0.21	0.14	0.11	0.11	...	0.17	0.31	0.25	0.17	0.24	0.19	0
3	0.15	0.13	**0.44**	0.15	0.14	0.19	...	0.10	0.26	0.33	0.22	0.12	0.23	0
4	0.21	0.18	0.16	**0.42**	0.26	0.23	...	0.22	0.26	0.06	0.04	0.04	0.07	0
5	0.14	0.13	0.15	0.19	**0.35**	0.12	...	0.11	0.02	0.01	0.01	0.02	0.05	0
6	0.13	0.17	0.11	0.21	0.18	**0.36**	...	0.25	0.06	0.04	0.04	0.05	0.21	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
45	0.17	0.21	0.14	0.21	0.08	0.24	...	**0.41**	0.04	0.09	0.11	0.08	**0.42**	**2**
46	0.09	0.18	0.05	0.03	0.02	0.01	...	0.06	**0.47**	0.34	0.24	0.18	0.14	0
47	0.03	0.18	0.08	0.00	0.01	0.01	...	0.25	0.04	**0.45**	0.20	0.10	0.05	0
48	0.07	0.03	0.11	0.00	0.01	0.00	...	0.17	0.21	0.40	**0.42**	0.15	0.09	0
49	0.16	0.21	0.14	0.04	0.02	0.04	...	0.23	0.09	0.34	0.26	**0.44**	0.18	0
50	0.05	0.07	0.09	0.02	0.03	0.03	...	0.18	0.07	0.07	0.04	0.03	**0.42**	0
Average accuracy of 50 users: 99.6%, false identification rate of 50 users: 0.4%														

Legend: S = Similarity; UI = User ID

4.2 Test Results

First, we cross-validated the data set of 50 users, as shown in Table 2. The first row and column show the ID of each user, and the value in the table means the similarity score between two users. The last column shows the FIR (False Identification Rate). In real experiments, we find that there is only one of 50 users who falsely identified. Hence the average FIR is 0.04%. From these results, we also observe that for correctly matched users, the similarity scores are ranged from 0.38–0.47 and which is below 0.37 for most false matched users. Hence, we can set 0.375 as the similarity threshold for *TouchSense* to identify legitimate users and attackers.

To further evaluate the performance of *TouchSense*. We did mass simulations on 100,000 samples of attackers and legitimate users based on the data set collected from 50 users. The FAR (False Accept Rate) and FRR (False Reject Rate) in different operation times are respectively shown in the Fig. 8 and 9.

The FAR curve (Fig. 8) shows the rate of attackers who falsely accepted by *TouchSense* in different operation times, represents the performance of security during usage. For 100,000 samples, the simulation result shows that only 2.19% of attackers will be left after 10 operations (9.5 s), and none of them will be left after 18 operations (17 s). Hence, *TouchSense* identify 97.8% attackers within 10 s, and locks all of them after 17 s. If an attacker stole a phone which deployed *TouchSense*, he knows the password, and he wants to steal 1 US dollar through E-bank payment, he has only 0.064% probability to succeed because he has to take at least 18 operations (Open software and find the payment function: at least 2 steps. Input the attackers account: at least 8 steps. Input the victim's password: at least 6 steps. Pay money: at least 2 steps).

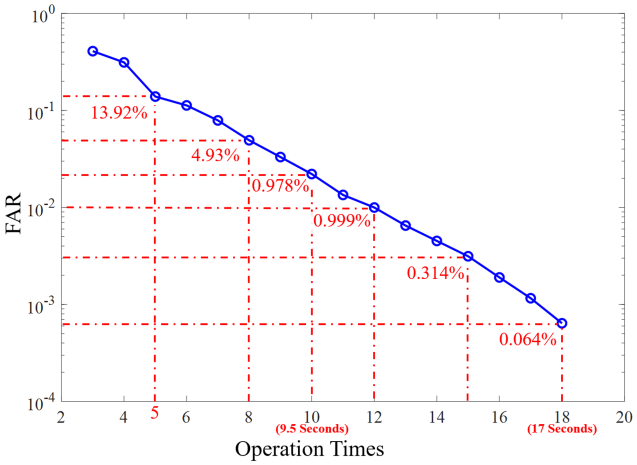


Fig. 8. The FAR (False Accept Rate) of attackers for different operation times. The FAR rapidly reduces with the increase of operations. For 10 operations the FAR is only 2.19%, and it goes to 0 after 18 operations.

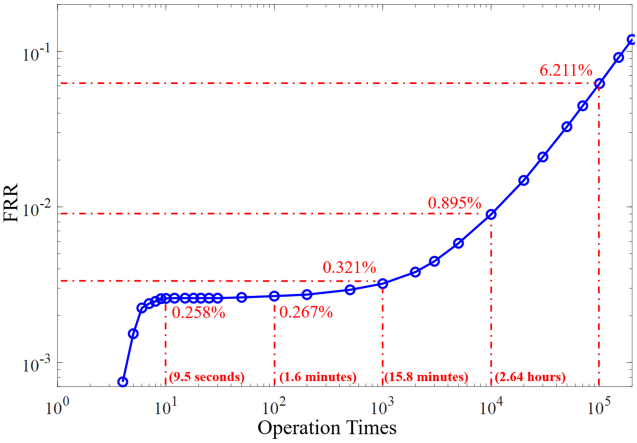


Fig. 9. The FRR (False Reject Rate) of legitimate users for different operation times. The FRR is very low in short usage, and it increases slowly with more operations as the legitimate users who misjudged by *TouchSense* will be accumulated during the usage. However, it hardly affects the majority of legitimate users, as even in 2.64-h of long-time continuous usage, the FRR keeps below 0.9%.

The FRR curve (Fig. 9) shows the rate of legitimate users which falsely rejected by *TouchSense*. The lower the FRR, the fewer legitimate users are misjudged and logged out by *TouchSense*. From the simulation results, we can see there are only 0.258% legitimate users falsely rejected by *TouchSense* for 10 operations (9.5 s), and for 100 operations (1.6 min), the FRR is only 0.267%. As user re-authentication is a repetitive process, despite in every operation, the legitimate users have very little chance to be logged out due to misjudging, the number will also be accumulated during usage. Hence the FRR increases slowly with more operations, for 1,000 operations (15.8 min), the FRR growth to 0.312%, and for 10,000 operations (2.64 h), the FRR is only 0.895%. Moreover, for the legitimate users who unfortunately be misjudged and excluded from the system, only need to re-login, and he can use the device freely like before. In summary, *TouchSense* can quickly identify and lock attackers but hardly affect legitimate users in daily use.

5 Discussion

The future works and ethical concerns of *TouchSense* are discussed below.

5.1 Feature Works

Due to the limitation of hardware, this paper only verifies the feasibility of *TouchSense* by one-finger touching, as the platform only deploys one sensor to sense the touch. In future works, we can use sensor arrays to authenticate users by more behaviors (*e.g.* multi-finger touches), and further improve accuracy by fusing the data collected from multi-sensors. The basic idea is to build sensing points which evenly distributed on the screen, shown in Fig. 10. The maxim interval to place sensing points (L_{Max}) can be defined as

$$L_{Max} = (2 \cdot L_2^2 - 2 \cdot L_1^2)^{\frac{1}{2}}$$

L_2 is the maximum sensing range of the sensing points, and L_1 is the maximum distance between the user's finger and the screen during the touching. In most cases, the user's fingers are close to the screen in using ($L_1 \approx 1$ cm by the test). Therefore, we only need to deploy a few sensing points to cover the whole screen. For SX9310 ($L_2 \approx 2$ cm by tested), the maximum deploy interval L_{Max} is 24.5 mm, and only 12 sensing points are required to cover a 6-inch screen (92 × 122 mm). Also, we can use low-power approaches to optimize power consumption of sensor array [20,32].

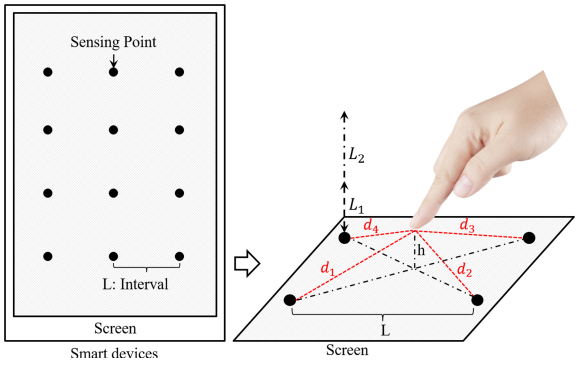


Fig. 10. The distribution schematic of sensing points. L_1: Maximum operation distance for most users ($h \leq L_1$). L_2: Maximum sensing range for the sensor. If the maximum distance from the finger to the sensing point $MAX(d_1, d_2, d_3, d_4) \leq L_2$, all the user motion trajectories can be captured.

5.2 Ethical Concerns

In this paper, the collected user features (*e.g.* biometric capacitance, finger touching behavior) will be only used and stored locally without appearing on Internet. Thus, we believe that our work does not involve ethical issues and user privacy leakage.

6 Conclusion

In this paper, we describe our approach towards *TouchSense*, a new method to re-authenticate the user accurately and transparently. *TouchSense* senses and authenticates the user by the combined information (touching behavior and biometric capacitance) as long as the user operates the smartphones. To complete our design, we build up User-Legitimate Module to comprehensively evaluate the user's legitimacy by stitch the authentication results of each finger touching. The experimental results show that *TouchSense* achieves 87.26% accuracy and offers 16.12% FAR and 9.36% FRR in a single touch. Further, the simulation result indicates that *TouchSense* identifies 98% attackers within 10 s (10 touching operations) and logs out all of them within 17 s (18 touching operations). However, for legitimate users, the misjudgment is only 0.321% in 16-minutes-usage (1,000 touching operations), and only 0.89% in 2.6-hours-usage (10,000 touching operations). Moreover, *TouchSense* is also designed in light-weight, which not require intensive computations and power. With the merits of high-security, user-transparency, low power consumption, and continuous security, we foresee that *TouchSense* can be wildly deployed on smartphones in the future.

Acknowledgments. The authors would like to thank the editors and anonymous reviewers for their comments and feedback, which is helpful to the publication of *Touch-*

Sense. This work is supported by the National Natural Science Foundation of China (61872061).

References

1. Barrett, G., Omote, R.: Projected-capacitive touch technology. Inf. Display **26**(3), 16–21 (2010)
2. Carrizo, C., Ochoa, C., Massuh, L.: Gesture-based signature authentication, 22 March 2016. US Patent 9,292,731
3. Clancy, T.C., Kiyavash, N., Lin, D.J.: Secure smartcard-based fingerprint authentication. In: Proceedings of the 2003 ACM SIGMM Workshop on Biometrics Methods and Applications, pp. 45–52. ACM (2003)
4. Crouse, D., Han, H., Chandra, D., Barbello, B., Jain, A.K.: Continuous authentication of mobile user: fusion of face image and inertial measurement unit data. In: 2015 International Conference on Biometrics (ICB), pp. 135–142. IEEE (2015)
5. Darlene, M.: Easy way to bypass passcode lock screens on iPhones, iPads running iOS 12 (2018). https://www.computerworld.com/article/3041302/4-new-ways-to-bypass-passcode-lock-screen-on-iphones-ipads-running-ios-9.html
6. Dascalescu, A.: Biometric authentication overview, advantages and disadvantages (2019). https://heimdalsecurity.com/blog/biometric-authentication/
7. Delač, K., Grgić M.: A survey of biometric recognition methods (2004)
8. Derawi, M.O., Nickel, C., Bours, P., Busch, C.: Unobtrusive user-authentication on mobile phones using biometric gait recognition. In: 2010 Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), pp. 306–311. IEEE (2010)
9. Feng, H., Fawaz, K., Shin, K.G.: Continuous authentication for voice assistants. In: Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking, pp. 343–355. ACM (2017)
10. Frank, M., Biedert, R., Ma, E., Martinovic, I., Song, D.: Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. IEEE Trans. Inf. Forensics Secur. **8**(1), 136–148 (2013)
11. Fujiwara, O., Ikawa, T.: Numerical calculation of human-body capacitance by surface charge method. Electron. Commun. Japan (Part I: Commun.) **85**(12), 38–44 (2002)
12. Gafurov, D.: A survey of biometric gait recognition: approaches, security and challenges. In: Annual Norwegian Computer Science Conference, pp. 19–21 (2007)
13. Gray, T.: Projected capacitive touch basics. Projected Capacitive Touch, pp. 5–17. Springer, Cham (2019). https://doi.org/10.1007/978-3-319-98392-9_2
14. Grivna, E.L.: Capacitance touch screen, 7 January 2014. US Patent 8,624,845
15. Hao, Z., Li, Q.: Towards user re-authentication on mobile devices via on-screen keyboard. In: 2016 Fourth IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb), pp. 78–83. IEEE (2016)
16. Sampathkumar, K., Balaji Sr., P.: Biometric methods - a secure survey. SSRN Electron. J. (2009)
17. Jason: How to reset android lock screen password and pin (2019). https://www.lifewire.com/reset-android-lock-screen-password-2740708
18. Jonassen, N.: Human body capacitance: static or dynamic concept? [esd]. In: Electrical Overstress/Electrostatic Discharge Symposium Proceedings, 1998, pp. 111–117. IEEE (1998)

19. Li, L., Zhao, X., Xue, G.: Unobservable re-authentication for smartphones. In: NDSS, vol. 56, pp. 57–59 (2013)
20. Li, S., Lu, L., Hussain, M.J., Ye, Y., Zhu, H.: Sentinel: breaking the bottleneck of energy utilization efficiency in RF-powered devices. IEEE Internet Things J. **6**(1), 705–717 (2018)
21. Li, Y., Hu, H., Zhou, G.: Using data augmentation in continuous authentication on smartphones. IEEE Internet Things J. **6**(1), 628–640 (2019)
22. Liu, S.Y., Wang, Y.J., Lu, J.G., Shieh, H.P.D.: 38.3: one glass solution with a single layer of sensors for projected-capacitive touch panels. In: SID Symposium Digest of Technical Papers (2014)
23. Lorenzo, A.D., Andreoli, A., Battisti, P., Talluri, T., Yasumura, S.: Total body capacitance correlates with total body potassium. Ann. New York Acad. Sci. **904**, 259–262 (2010). (In vivo body composition studies)
24. Lovejoy, B.: 3D mask or photo fools airport and payment face-recognition, but not face id (2019). https://ww.9to5mac.com/2019/12/16/3d-mask/#aprd
25. Mahfouz, A., Mahmoud, T.M., Eldin, A.S.: A survey on behavioral biometric authentication on smartphones. Inf. Secur. Tech. Rep. **37**, 28–37 (2018)
26. Maiorana, E., Campisi, P., Neri, A.: Biometric signature authentication using radon transform-based watermarking techniques. In: Biometrics Symposium, 2007, pp. 1–6. IEEE (2007)
27. Maiorana, E., Campisi, P., Neri, A.: Template protection for dynamic time warping based biometric signature authentication. In: 2009 16th International Conference on Digital Signal Processing, pp. 1–6. IEEE (2009)
28. Matsumoto, T., Matsumoto, H., Yamada, K., Hoshino, S.: Impact of artificial "gummy" fingers on fingerprint systems. Electron. Imaging **4677**, 275–289 (2002)
29. Mi, D.: Single-layer projected capacitive touch panel and method of manufacturing the same (2017)
30. Mo, M., Li, H., Zhang, J.: Capacitance touch screen with mesh electrodes, 15 March 2012. uS Patent App. 13/226,902
31. Muaaz, M., Mayrhofer, R.: An analysis of different approaches to gait recognition using cell phone based accelerometers, pp. 293–300 (2013)
32. Nesa, N., Banerjee, I.: SensorRank: an energy efficient sensor activation algorithm for sensor data fusion in wireless networks. IEEE Internet Things J. **6**(2), 2532–2539 (2018)
33. Nickel, C., Busch, C., Rangarajan, S., Möbius, M.: Using hidden Markov models for accelerometer-based biometric gait recognition. In: Proceedings - 2011 IEEE 7th International Colloquium on Signal Processing and Its Applications, CSPA 2011, March 2011. https://doi.org/10.1109/CSPA.2011.5759842
34. Primo, A., Phoha, V.V., Kumar, R., Serwadda, A.: Context-aware active authentication using smartphone accelerometer measurements. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 98–105 (2014)
35. Qi, M., Lu, Y., Li, J., Li, X., Kong, J.: User-specific iris authentication based on feature selection. In: 2008 International Conference on Computer Science and Software Engineering, vol. 1, pp. 1040–1043. IEEE (2008)
36. Rathgeb, C., Uhl, A.: A survey on biometric cryptosystems and cancelable biometrics. EURASIP J. Inf. Secur. **2011**(1), 3 (2011)
37. Semtech: Semtech sx9310, ultra-low power smart proximity sensor for SAR (2017). https://www.semtech.com/uploads/documents/sx9310.pdf
38. Sharma, K.: How dangerous are impersonation attacks? (2018). https://cybersecurity.att.com/blogs/security-essentials/how-dangerous-are-impersonation-attacks

39. Thang, H.M., Viet, V.Q., Thuc, N.D., Choi, D.: Gait identification using accelerometer on mobile phone, pp. 344–348 (2012)
40. TycoyokeI: How to fool a fingerprint security system as easy as ABC (2019). https://www.instructables.com/id/How-To-Fool-a-Fingerprint-Security-System-As-Easy-/
41. Yan, J., Blackwell, A.F., Anderson, R., Grant, A.M.: Password memorability and security: empirical results. In: IEEE Symposium on Security and Privacy, vol. 2, no. 5, pp. 25–31 (2004)
42. Zhang, S.: Main construction of capacitive touch screen (2019). https://www.vtouchscreen.com/news/main-construction-of-capacitive-touch-screen-28265581.html
43. Zhang, Y.-B., Li, Q., You, J., Bhattacharya, P.: Palm vein extraction and matching for personal authentication. In: Qiu, G., Leung, C., Xue, X., Laurini, R. (eds.) VISUAL 2007. LNCS, vol. 4781, pp. 154–164. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76414-4_16
44. Zhong, Y., Deng, Y., Meltzner, G.S.: Pace independent mobile gait biometrics, pp. 1–8 (2015)