International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 3 | Mar -2017 www.irjet.net p-ISSN: 2395-0072

TRANSPARENT DEVELOPMENTAL BIOMETRIC BASED SYSTEM PROTECT USER REAUTHENTICATION USING SMARTPHONES

[*1] Mr. R.Arunkumar, [*2] M.Arunraj, [*3] R.Prabhakaran, [*4] Mr.V.Naveen

*[*1,2,3] B.Tech Student, Department of Information Technology Kingston Engineering College, Christainpettai, Katpadi, Vellore.*

*[*4] Assistant Professor, Department of Information Technology Kingston Engineering College, Christainpettai, Katpadi, Vellore.*

Abstract - ***With the emergence of smartphones as an essential part of daily life, the demand for user reauthentication has increased manifolds. The effective and widely practiced biometric schemes are based upon the principle of "who you are" which utilizes inherent and unique characteristics of the user. In this context, the behavioral biometrics such as sliding dynamics make use of on-screen sliding movements to infer the user's patterns. In this paper, we present Safeguard, an accurate and efficient smartphone user reauthentication (verification) system based upon on-screen finger movements.. The key feature of the proposed system lies in fine-grained on-screen biometric metrics, i.e., sliding dynamics and pressure intensity, which are unique to each user under diverse scenarios. We first implement our scheme through five machine learning approaches and finally select the support vector machine (SVM)-based approach due to its high accuracy. We further analyze Safeguard to be robust against adversary limitation. We validate the efficacy of our approach through implementation on off-the-shelf smartphone followed by practical evaluation under different scenarios.***

Key Words: *SVM, honeybeeandantcolony, FAR,FRR, reauthentication with smart phone, Mobile Verification.*

I. INTRODUCTION

Mobile Devices, especially smartphone, becomes the most common used tools for human daily accessing mobile social networks (MSNs) [1]-[3]. Thus, preserving users' privacy and sensitive data grows to the urgent need as smartphones store users' more private and sensitive information. At present, many smartphone authentication schemes are With the introduction of the Android operating system for mobile phones, an alternative to PIN-authentication on mo-bile devices was introduced and widely deployed for the first time. The password pattern, similar to shape-based authentication approaches. Despite its manifold advantages, this approach has major drawbacks, the most important one being security. Drawn passwords are very easy to spy on [11,36], which makes shoulder surfing, a common attack in public settings [27], a serious threat. Other attacks include the infamous smudge attack [1], in which finger traces left on the screen are used to extract the password. Due to its weak security properties, this authentication approach does not fully meet the requirement of adequately protecting the user's data stored on the device. Nowadays, not only private but also valuable business information is stored on the user's handheld [10]. Therefore, resistance to attacks is a major concern when designing respective authentication systems.

The biometric-based authentication has been widely used in many applications such as access control and continuous tracking systems [5]. However, the reauthentication is still challenging on smartphones. The reason lies in three-folds: 1) Current biometric approaches, such as fingerprints and retinal [6] scans, provide accurate one-time authentication but require specialized hardware, which may be nontransparent, expensive, or unavailable for smartphones. 2) A typical smartphone is equipped with a touch screen as the user interface. Thus, the classical behavioral biometrics on PCs, such as keystroke and mouse dynamics, cannot be applied on smartphones for user verification. 3) Existing reauthentication schemes for smart-phone are not practical for real applications because of low accuracy [7], limited application scenarios [8] or high system overhead [10]. Moreover, most existing approaches only address the one-time authentication when users unlock the screen of the smartphones.

.

We employ machine learning to devise an efficient system and use support vector machine (SVM) for optimal results. More in detail, the sliding movements of a legitimate user are mapped from touch screen in a passive and transparent way. Same information is also used to extract the angle metric, which is a unique feature of user's sliding movements. In parallel, the pressure-based metrics are calculated as the user applies his/her fingers on the screen. With these features, Safeguard "trains" a model of the user's behavioral biometrics consisting of on-screen pressure and the sliding movements (curve, angle, distance ratio). After the training, Safeguard records the new biometric parameters and compares them with the modeled profile of the user. After the classification, a decision is made based upon the similarity threshold on whether new parameters relate to a valid user or not. In essence, our key approach is to exploit the behavioral features by means of angle and pressure-based metrics which are the outputs of user's sliding movements and intensity of pressure on the touch screen. Both of these metrics are observed to be unique for each user and distinct from person to person.

Silent Sense is designed as a pure software-based framework, running in the background of smartphone, which nonintrusive explores the behavior of users interacting with the device without any additional assistant hardware. The main idea of Silent Sense for user identification comes from two aspects: (1) how you use the device; and (2) how the device reacts to the user action. While using mobile devices, most people may follow certain individual habits unconsciously. Running as a background service, Silent Sense exploits the user's app usage and interacting behavior with each app, and uses the motion sensors to measure the device's reaction. Correlating the user action and its corresponding device reaction, Silent Sense establishes a unique biometric model to identify the role of current user.

II. RELATED WORK

The underlying principle of Biometric-based user authentication focuses on "who you are" which differs from conventional user authentication approach that mainly relies on "what you have" or "what you know." Thus, a biometric-based approach is based on the inherent and unique characteristics of a human user being authenticated. Biometric-based reauthentication approaches have been widely studied for PCs [8]-[7]. However, we can find only few such implementations on smartphones, which are either limited by coarse accuracy or restrict application scenarios or gestures. Therefore, in this section, we focus on the state-of-the-art on smartphones. We first present some smartphone applications that perform the one-time identification and reauthentication.

User Behavior Modeling: To characterize individuals' unconscious use habits accurately, the user behavior model should contain multiple features of both user's action and device's reaction. In addition, the connection between features may not be neglected for identification, such as different interaction coordinates on the touch screen may cause different reaction vibrations of the device.

Identification Strategy: To establish the user behavior model, for the owner there are abundant behavior information. For a guest, the collected behavior information may be very limited. In addition, in motion scenarios, some interacting features will be swamped by the motion from the perspective of sensory data, which greatly increases the difficulty of accurate identification. Thus it is challenging to distinguish users with limited information effectively, even if the interacting features are partially swamped.

Balance Among Accuracy, Delay and Energy: Nonstop observation with sensors provides identification with small delay and high accuracy, but may cause unwanted energy consumption for the mobile device when the current user is the owner or a guest is using an insensitive app, *e.g.*, playing a game. But intrusion may happen when the sensors are off and the risk increases with the detection delay. A well designed mechanism is required to decide the observation timing to reduce energy consumption while guarantee the identification accuracy and delay.

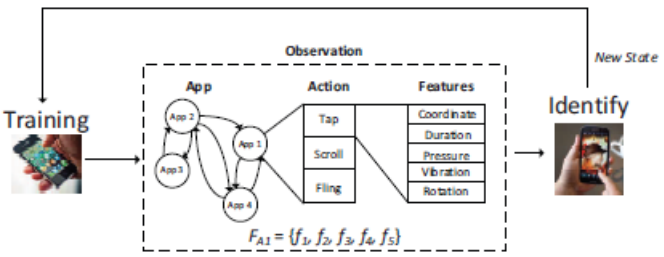

Fig : System Framework

The framework model consists of two basic phases: *Training* and *Identification*, as shown in Figure 1. The training phase is conducted to build a behavior model when the user is interacting with the device, and the identification phase is implemented to distinguish the identity of the current user based on the observations of each individual's interacting behaviors. When a guest user is observed, privacy protection mechanism will be triggered automatically. After the guest leaves and the owner returns, privacy protection will be reset for the owner's convenience

Initially, we assume that the device has only the owner's information, *e.g.* a newly bought phone. The framework trains the owner's behavior model by retrieving two types

of correlated information, the information of each touch screen action and the corresponding reaction of the device when the user is static. In the motion scenario, instead of the reaction information, motion features will be detected. With the owner's behavior model, the current user will be identified through one-class SVM classification. One-class SVM provides a judgment whether the observed features belong to the owner (true of false). However the identification accuracy by one observation is usually not high enough for an identity conclusion, continuous consistent judgments will increase the accumulated confidence for this judgement.When the accumulated confidence is high enough, a conclusion is ready and the newly observed features will be added to the owner or guest dataset according to the conclusion to update the model.

Initially, without only the owner's behavior data, a one-class SVM model is trained to identify a new observation O_i and provide the judgment Ji whether this action belongs to the owner or not, *i.e.*, J_i = *true or* J_i = *false*. Lacking of ground truth, it is difficult to determine the correctness of the judgment. To achieve high identification accuracy, we adopt the SVM model's credibility for each judgment Ji as the *confidence* of the framework on J_i. Let this confidence be "*i(Ji*), which indicates the probability the framework considers that $_{Ji}$ is correct for the current observation Oi. Using one-class SVM, the judgment of one observation usually is not accurate enough to make a identity conclusion. Obviously, more observations leads to higher conclusion accuracy. Specifically, let $\{J_1, J_2, \cdots, J_k\}$ be a sequence of consistent judgments, *i.e.* $J_1 = J_2 = \cdots = Ji$, to continuous observations $\{_{O1,O2,} \ldots {}_{,Ok}\}$. Based on the judgment sequence, an identity conclusion I1,k can be made, *i.e.* $I_{1,k} = J$k. Then the *conclusion confidence* will be cumulated as

$$P_{1,k} = P(J_1, J_2, \cdots, J_k) = (1 - \pi (1 - "i(J_i))) - (1).$$

Which indicates the probability this framework considers that the identity conclusion $I_{1,k}$ is correct. Then the identification delay d_k for a conclusion $I_{1,k}$ is defined as the number of observations taken to achieve this conclusion. With the number of observation increases, the framework will be more confident to provide a correct conclusion, meanwhile the delay will increase. Note that, an inconsistent judgment will interrupt the sequence, and the conclusion confidence needs to be cumulated from scratch.

III. PREVIOUS IMPLEMENTATIONS

At present, many smartphone authentication schemes are. in practice including PIN and sliding pattern unlock which are the most common. These mechanisms, however, give one-time security and cannot protect users' privacy in run-time. For example, an adversary can operate a user's phone once the PINis stolen or compromised, and can easily abuse the owner's Private information. In addition, the verified users can still fall victim to both session hijacking and leakage of the secret information. Thus, to guarantee the user authenticity, more frequent user verification is needed. However, currently practiced user verification and reauthentication methods are not suitable for frequent verification, as they are not passive or transparent to users. Like, frequently answering predefined secure questions is too obtrusive and inconvenient to be acceptable. This necessitates a user-re authentication scheme which should be transparent and easily implementable in run-time without any user involvement and system overhead.

Existing smartphones are equipped with sensors such as GPS, microphone, accelerometer, magnetic field, gravity, temperature, and gyroscope. These sensors provide a side channel to compromise the privacy of smartphone owners as they can record some biometric characteristics [23]. By accessing the camera and microphone of a smartphone, its location can be successfully identified [2] Cai and Chen [6] presented an effective method to infer the sliding unlock pattern by observing finger smudges on a smart phone's screen. An interesting attack called Tap prints in [5] exploits the gyroscope readings and guesses the on-screen positions. The basic idea is to use the gyroscope to record vibrations caused by user pressing on different positions of the screen. Through machine learning, this attack can effectively recognize user typing on the virtual keyboard of a smartphone with the equal error rate (EER) not less than 30%.

Drawback in previous implementation Users tend to choose short passwords or Passwords that square measure straight- forward to recollect. Sadly, these passwords is simply guessed or broken. Another technique is statistics, like fingerprints, iris scan or identity verification has been introduced however not nonetheless wide adopted. square measure graphical positive. Identification schemes that are planned that square measure proof against shoulder-surfing, however they need their own drawbacks like usability problems or taking longer for users to login.

One-Time Identification: This method enhances the security of sliding unlock pattern on smartphones by employing the intensity of pressure on touch screen [13]. Shahzad *et al.* [14] present an approach for unlocking smartphone by gestures. Zheng *et al.* [12] also present an approach for enhancing PIN with behavior biology. Basically, these approaches first build models for different behaviors on touch screen, and then make use of classification algorithms for user identification. However, these approaches can provide enhanced security only for one time while the user is unlocking the smartphone. They do not provide continuous authentication during user operations which makes them a nontransparent process. Moreover, each model for a particular sliding movement on touch screen, say sliding up, needs to be trained

separately which incurs extra training overheads and restricts the application to limited behaviors. Vu *et al.* [28] develop a special hardware token device which makes use of specific charging and discharging of capacitors

IV. SYSTEM IMPLEMETNATION

Proposed the transparent biometric-based reauthentication Approach, called Safeguard, which is used to verify the users based upon their passive observable behaviors, i.e., the sliding curves made by the user as smartphone is used and pressure intensity applied on the touch screen. Both, sliding movements and touch screen pressure intensity are unique and Distinguishable parameters of each user and form the core of our design methodology due to two reasons. the First security check we match the angle of the mobile. In this, firstly user has to match the correct angle for unlocking the mobile that is X, Y, Z co-ordination. And this is calculated on the base of rotation of mobile. The option is available to choose the number of angles is included in the password. The user wants to select 1 angle, 2 angle or all the 3 angles it depends on the user. When all the three angles are selected, then security will be very high. In the particular mobile rotation the corresponding change in X, Y and Z value. And in the second step two security checks are done at one time. In second security check we use the password pattern. The Password pattern is like draw-a secret shape on the screen. a user unlocks the mobile first time, we match time taken to draw with initial time (when user changes the pattern password), and it store his/her average time of initial and first time taken to draw patterns.

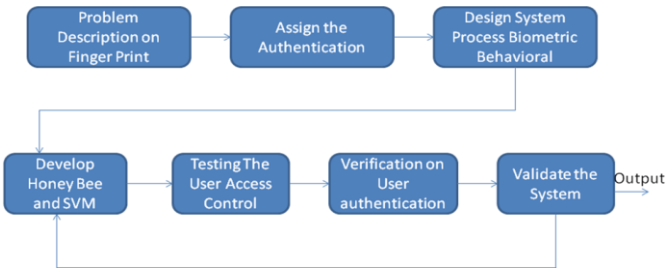

Fig : System Model

Advantages of Biometric identification can provide extremely accurate, secured access to information; fingerprints, retinal and iris scans produce absolutely unique data sets when done properly. Current methods like password verification have many problems (people write them down, they forget them, they make up easy-to-hack passwords). Automated biometric identification can be done very rapidly and uniformly, with a minimum of training. Your identity can be verified without resort to documents that may be stolen, lost or altered.

SYSTEM ARCHITECTURE

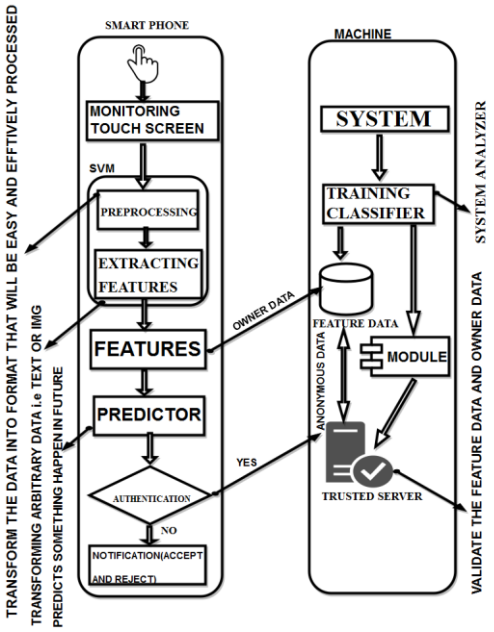

Fig : System Architecture

Key Contributions

In this paper, we make following five key contributions.

- We proposed, implemented, and evaluated a gesture based authentication scheme for the secure unlocking of touch screen devices.
- We identified a set of effective features that capture the behavioral information of performing gestures on touch screens.
- We proposed an algorithm that automatically segments each stroke into sub-strokes of different time duration where for each sub-stroke the user has consistent and distinguishing behavior.
- We proposed an algorithm to extract multiple behaviors from the training samples of a given gesture.
- We collected a comprehensive data set containing 15009 training samples from 50 users and evaluated the performance of GEAT on this data set.

Algorithm1: Honey Bee and Ant Colony Algorithm

Step1.Initialise population with random solutions.

Step2. Evaluate fitness of the population.

Step3. While (stopping criterion not met)

//Forming new population.

Step4. Select sites for neighbourhood search.

Step5. Recruit bees for selected sites (more bees for best **e** sites) and evaluate fitnesses.

Step6. Select the fittest bee from each patch.

Step7. Assign remaining bees to search randomly and evaluate their fitnesses.

Step8. End While.

Algorithm Implementation

For all i such that i is a valid path number do

If obpa (i) < cwnd(i) then

Transmit on path i until obpa (i) = cwnd(i);

If more data is pending transmission in queue then

Choose next path i;

End

End

End

If no path was available for transmission and data is pending transmission

Then

Find path j that has the minimum ratio of obpa (j)

Cwnd (j) over all paths;

Transmit one MTU of data on path j;

End

Algorithm 2: SVM Supported Vector Machine

Data: PathSet ← set of valid path numbers sorted in ascending order of the number of packets missing on the respective paths

Suppose we choose the Threshold (seen below) that is close to some sample Now suppose it have a new point that should be in class "-1" and is close to . Using our classification function his point is misclassified! -The SVM idea is to maximize the distance between The hyperplane and the closest sample point.

Let j be the first path from PathSet;

While data is pending transmission do

If (obpa(j) < cwnd(j)) or (missing(j) + sentAlready(j) < missing(j +1))

Then

transmit on path j;

End

Else

Choose path j + 1;

If j + 1 is not a valid path number then

Break out of the loop;

End
End

A) Smartphone Sensors :

Most modern smart phones have built-in sensors which can measure motion and environmental and positional environment the devices are subject to. They provide several facilities such as providing accurate and precise raw data, observing the position in three dimensions, and measuring any possible changes in the surrounding environment sufficiently close to the device.

- **Motion sensors** These sensors measure acceleration forces and rotational forces along three axes. This category includes accelerometers, gravity sensors, gyroscopes, and rotational vector sensors.
- **Environmental sensors** These sensors measure various environmental parameters, such as ambient air temperature and pressure, illumination, and humidity. This category includes barometers, photometers, and thermometers.
- **Position sensors** These sensors measure the physical position of a device. This category includes orientation sensors and magnetometers.

Sensor Coordinate System

In general, the sensor framework uses a standard 3-axis coordinate system to express data values. For most sensors, the coordinate system is defined relative to the device's screen when the device is held in its default orientation (see figure 1). When a device is held in its default orientation, the X axis is horizontal and points to the right, the Y axis is vertical and points up, and the Z axis points toward the outside of the screen face. In this system, coordinates behind the screen have negative Z values. This coordinate system is used by the following sensors:

Coordinate system (relative to a device) that's used by the Sensor API.

- Acceleration sensor
- Gravity sensor
- Gyroscope
- Linear acceleration sensor
- Geomagnetic field sensor

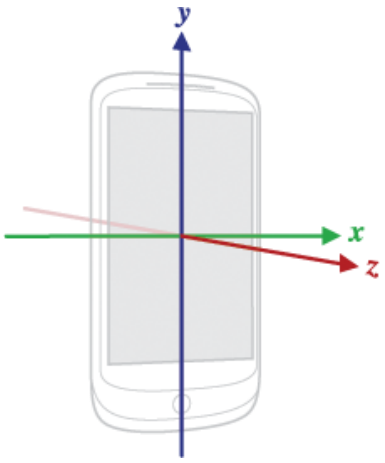

Fig: Sensor Detection

The most important point to understand about this coordinate system is that the axes are not swapped when the device's screen orientation changes—that is, the sensor's coordinate system never changes as the device moves. This behavior is the same as the behavior of the OpenGL coordinate system. Another point to understand is that your application must not assume that a device's natural (default) orientation is portrait. The natural orientation for many tablet devices is landscape. And the sensor coordinate system is always based on the natural orientation of a device.

Fig : Finger Print Sensor

. **Image Authentication:**

In this type the authentication user select a points from the given image. In the time of login the application allow in the system when the selected region and previously selected regions are match. The Fingerprint Manager (and its Support Library counterpart, Fingerprint Manager Compact) is the primary class for using the fingerprint scanning hardware. This class is an Android SDK wrapper around the system level service that manages interactions with the hardware itself. It is responsible for starting the fingerprint scanner and for responding to feedback from the scanner. This class has a fairly straightforward interface with only three members:

- **Authenticate** – This method will initialize the hardware scanner and start the service in the background, waiting for the user to scan their fingerprint.
- **Enrolled Fingerprints** – This property will return true if the user has registered one or more fingerprints with the device.
- **Hardware Detected** – This property is used to determine if the device supports fingerprint scanning.

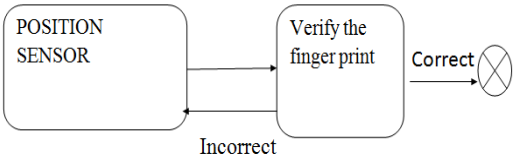

Fig : Image Authentication

Draw line Authentication:

In this type the authentication user draw a line by connecting two points from the given image. For this system will take the starting and ending point for Authentication. During login only allow in to the system when both the lines are equal.

1. ***Data Structuring and Processing***

Sliding metrics: a raw sliding movement can be represented as a series of successive points. From here, we can extract the tuples of Cartesian coordinate pairs, pressure intensity, size, and timestamp at each point. We record the biometric values after each time interval of roughly 17 ms, which serves as a single sample point

1) ***Direction:*** For any two consecutive points *A* and B,we record the direction along the line *AB* from the first point to the second. The direction is defined at the angle *a* between the line *AB* and the horizontal axis.
2) ***Angle:*** For any three consecutive points *A*, *B*, and *C* ,the angle of the curvature is *Z A B C*, i.e., the angle between lines *ABB* and *Bs C*.
3) ***Distance ratio:*** For any three consecutive points *A*, *B*, and C, consider the length of line *AC*. The distance ratio is the ratio of the length of the

perpendicular distance from midpoint B to AC to AC.

4) ***Sliding duration:*** The time duration between a finger touches and leaves the screen.
4) ***Sliding distance:*** The Euclidean distance between the locations where a finger touches and leaves the screen.
5) ***Sliding velocity:*** The ratio of sliding duration and sliding distance.

2. Data Feature Exploration

At first glance, it seems that we should use all available features to identify users. However, including too many features can worsen performance in practice because of their varying accuracies and potentially conflicting signatures.

3. Preprocessing

1) Data Grouping and Filtering: The sliding movements of a single user may have multiple shapes. In addition, the effects such as screen vibration and finger joggling bring outliers and incur deviations from the ideal slide curvature. In Safeguard, we employ data processing to classify the valid sets of sliding curvatures and to filer the redundant samples. As a result, the filtered samples, called the effective samples, are used for classification in a more precise and efficient way.

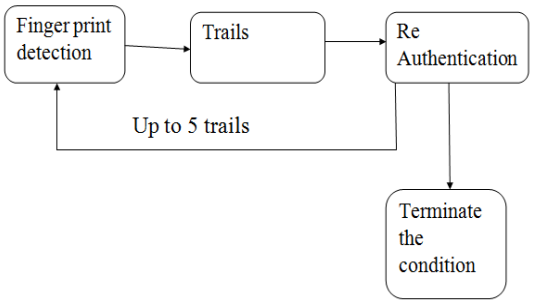

Fig : Draw line Authentication

CONCLUSION

In order to prevent unauthorized usage, we have proposed a re-authentication system using user finger movement. The system performs continuous reauthentication and does not need human assistance during re-authentication. We carried out all our data collections and experiments on Motorola Droid phones, which are equipped with low-end touch screens. Therefore, some metrics may be dropped due to the smartphone's hardware limitations. For example, we left out the pressure related metrics because the touch screen did not provide accurate pressure measurements. The metrics may need to be carefully tested or even re-designed before deploying our system on another smartphone platform. For example, pressure may become useful and provide more user information on some smartphone platforms. However, our work provides a guideline for the metric design on other platforms and our methodology can still be adopted. Our work shows that using gestures to construct an unobservable continuous re-authentication on smartphones is practical and promising. We have discussed biometric feature design and selection for finger movement. We have demonstrated the effectiveness and efficiency of our system in extensive experiments.

REFERENCES:

[1] M. Dong *et al.,* "Quality-of-experience (qoe) in emerging mobile social networks," *IEICE Trans. Inf. Syst.,* vol. 97, no. 10, pp. 2606-2612, 2014.

[2] M. Dong *et al.,* "Qoe-ensured price competition model for emerging mobile networks," *IEEE Wireless Commun.,* vol. 22, no. 4, pp. 50-57, Aug. 2015.

[3] K. Wei *et al.,* "Camf: Context-aware message forwarding in selfish mobile social networks," *IEEE Trans. Parallel Distrib. Syst.,* vol. 26, no. 8, pp. 2178-2187, Aug. 2014.

[4] Y. Zhang *et al.,* "The security of modern password expiration: An algorithmic framework and empirical analysis," in *Proc. 17th ACM Conf. Comput. Commun. Secur. (CCS),* 2010, pp. 176-186.

[5] A. A. E. Ahmed and I. Traore, "Anomaly intrusion detection based on biometrics," in *Proc. IEEE Inf. Assur. Workshop,* 2005, pp. 425-453.

[6] C. W. Shanley *et al.,* "Remote retinal scan identifier," U.S. Patent 5,359,669, Oct. 1994.

[7] T. Feng *et al.,* "Tips: Context-aware implicit user identification using touch screen in uncontrolled environments," in *Proc. 15th ACM Workshop Mobile Comput. Syst. Appl.* , 2014, p. 9.

[8] P. Saravanan *et al.,* "Latentgesture: Active user authentication through background touch analysis," in *Proc. 2nd ACM Int. Symp. Chin. CHI,* 2014, pp. 110-113.

[9] M. Frank *et al.,* "Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication," *IEEE Trans. Inf. Forensics Security,* vol. 8, no. 1, pp. 136-148, Dec. 2012.

[10] C. Bo *et al.,* "Silentsense: Silent user identification via touch and movement behavioral biometrics," in *Proc. 19th ACM Annu. Int. Conf. Mobile Comput. Netw. (Mobicom),* 2013, pp. 187-190.

[11] L. Li *et al.,* "Unobservable re-authentication for smartphones," in *Proc. 20th Annu. Netw. Distrib. Syst. Secur. Symp. (NDSS),* San Diego, CA, USA, Feb. 24-27, 2013.

[12] N. Zheng *et al.,* "You are how you touch: User verification on smart-phones via tapping behaviors," in *Proc. IEEE 22nd Int. Conf. Netw. Protoc. (ICNP),* 2014, pp. 221-232.

[13] A. De Luca *etal.,* "Touch me once and i know it's you!: Implicit authentication based on touch screen patterns," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.* , 2012, pp. 987-996.

[14] M. Shahzad *et al.,* "Secure unlocking of mobile touch screen devices by simple gestures: You can see it but you can not do it," in *Proc. 19th ACM Annu. Int. Conf. Mobile Comput. Netw. (Mobicom),* 2013, pp. 39-50.

[15] T. Buch et al., "An enhanced keystroke biometric system and associated studies," in Proc. CSIS Res. Day, New York, NY, USA, 2008.

[16] F. Monrose and A. D. Rubin, "Keystroke dynamics as a biometric for authentication," Future Gener. Comput. Syst., vol. 16, no. 4, pp. 351-359, 2000.

[17] F. Bergadano et al., "User authentication through keystroke dynamics," ACM Trans. Inf. Syst. Secur., vol. 5, no. 4, pp. 367-397, 2002.

[18] F. Monrose et al., "Password hardening based on keystroke dynamics," Int. J. Inf. Secur., vol. 1, no. 2, pp. 69-83, 2002. [19] H. Saevanee and P. Bhattarakosol, "Authenticating user using keystroke dynamics and finger pressure," in Proc. 6th IEEE Consum. Commun. Netw. Conf., 2009, pp. 1-2.

[20] A. A. E. Ahmed and I. Traore, "A new biometric technology based on mouse dynamics," IEEE Trans. Dependable Secure Comput.,vol.4, no. 3, pp. 165-179, Jul./Sep. 2007.

[21] N. Zheng et al., "An efficient user verification system via mouse move¬ments," in Proc. 18th ACM Conf. Comput. Commun. Secur. (CCS), 2011,