ISSN: 0974-2115
www.jchps.com Journal of Chemical and Pharmaceutical Sciences

Android based unlocking mobile waving pattern with waving pattern for emergency support system

VijayaKumar M, R.Senthamarai, K.Pushpavathi
Faculty of Computer Applications, Aarupadai Veedu Institute of Technology, Paiyanoor, Chennai-603104.
Vinayaka Missions University, Salem, India
***Corresponding author: E-Mail:kumarvijay7794@gmail.com**

ABSTRACT

Screen locking/unlocking is important for modern smart phones to avoid the unintentional operations and secure the personal stuff. Once the phone is locked, the user should take a specific action or provide some secret information to unlock the phone.

The existing unlocking approaches can be categorized into four groups: motion, password, pattern, and fingerprint. Existing approaches do not support smart phones well due to the deficiency of security, high cost, and poor usability. We collect 200 users' hand waving actions with their smart phones and discover an appealing observation: the waving pattern of a person is kind of unique, stable and distinguishable. In this paper, we propose Open Sesame which employs the users' waving patterns for locking/unlocking. The key feature of our system lies in using four fine-grained and statistic features of hand waving to verify users. Moreover, we utilize support vector machine (SVM) for accurate and fast classification. Our technique is robust compatible across different brands of smart phones, without the need of any specialized hardware. Results from comprehensive experiments show that the mean false positive rate of Open Sesame is around 15 percent, while the false negative rate is lower than 8 percent.

KEY WORDS: Smart phone, security, privacy, authentication, accelerometer.

1. INTRODUCTION

Nowadays, smart phones are no longer the devices that are only used to call or text others. They become prevalent with much more powerful functions. Acting as pocket PCs, smart phones can be used to deal with complicated tasks such as sending/receiving e-mails, shopping, mobile payment, etc. Screen locker is a fundamental utility for smart phones to prevent the device from unauthorized use. For example, the Apple iPhones and Android phones can lock themselves automatically after being idle for a short time. It can protect the privacy of users as well as prevent unintentional operations. Classical screen lockers have been proposed long time back.

(1) The most widely used one is Slide-to-Unlock. The user can unlock his/her phone through sliding his finger across a defined trajectory. This method is too simple to protect user's privacy.

(2) PIN, the most common method used by traditional digital device, is always adopted on smart phones for unlocking smart phones. However, due to the relatively small screen and frequent unlocking request, it is inconvenient to set long and complex PIN on phones. For example, there are only four numbers allowed to be set as unlocking PIN in iPhone's default setting. Such a short and simple PIN can often be easily guessed (Florencio, 2007; Bonneau, 2012).

(3) The user can pre-define a graphical password, like connecting at least four circles shown in the screen. Being similar to the PIN, simple graphic passwords are easy to be peeked and guessed, while the complex pattern may confuse the user and make inconvenience. To enhance the security as well as the flexibility, many biometric authentication methods (Park, 2010), are introduced for screen lockers. The secrets of these methods cannot be easily spied and reproduced since they identify the user based on her natural features. The biometric measures are grouped into two main categories (Yampolskiy, 2008) physiological biometrics and behaviour biometrics. Physiological biometrics leverage the physiological features of human beings to identify the user, including recognitions of face (Akkermans, 2005), voice (Jain, 1999), fingerprint (Phillips, 2000), ear (Akkermans, 2005), and so on. However, we find that (i) performances of these solutions are heavily influenced by external factors. For example, the face acquirement by the camera is severely affected by the illumination, resulting in the failure to identify user at night. Similarly, it is hard to distinguish the voice from the ambient interference in an extremely noisy environments, like subway or restaurant. Any authentication method must be adapted to all kinds of conditions. (ii) Unlocking operation is a very frequent operation, of which energy consumption should be carefully considered.

Related Work: This section reviews the related work. Accelerator based authentication. A work parallel to ours is that Conti et al. propose to adopt the movement the user performs when answering a phone call to authenticate the user of a smartphone, which utilizes two kinds of components, accelerometer and orientation sensors, in smart phones (Conti, 2011).

Waving Characterization: In this section, we introduce the sensor used for waving sensing, real trace collection, and analysis on the data.

Waving Sensing: For precisely characterizing user's waving actions, selecting appropriate sensors is necessary. As the tremendous growth of MEMS technology, there are many powerful sensors equipped in our smart phone today, such as camera, microphone, proximity sensor, accelerometer, gyroscope, and magnetic sensor etc. In our system,

the selected sensor should be able to depict the hand waving. In addition, it should be energy-efficient, stable, cheap, and compatible for wide deployment in most kinds of smart phones. Obviously, the first three sensors cannot capture the phone's motion. The gyroscope sensor is attractive because it is designed for measuring or maintaining purpose, based on the principles of angular momentum. Unfortunately, this kind of sensor is not a standard equipment in most smart phones due to its high price. The magnetic sensor is usually used for compass, but it tends to be interfered by the mental objects under special environment, like inside the car or subway. In our approach, we finally select the three-axis accelerometer as our feature detecting sensor. The accelerometer allows smart phones to detect the motion performed on them. The accelerometer in smart phones measures the acceleration of the phone relative to freefall. A value of 1 indicates that the phone is experiencing 1 g of acceleration exerting on it. ONE g of acceleration is the gravity, which the phone experiences when it is stationary. The accelerometer measures the acceleration of the phone in three different axes: X, Y, and Z.

Data Collection: For investigating the uniqueness of hand waving, we collect the waving action data from 200 distinct smart phone users. For each specific user, he is asked to shake the smart phone for more than 10 seconds and repeat for three times. Note that there is no special restriction on user's waving actions. He can shake the smart phone arbitrarily in each trail. Indeed, we aim at taking insight into the hand waving action but not the motion pattern.

The data is collected in two sampling modes: fast and normal modes. In the fast mode, the accelerometer samples every 10 to 20 milliseconds, corresponding to the acceleration value change rate. There are 100 users' traces collected using this mode. In the normal mode, the sampling interval is 200 milliseconds and 100 users' traces are sampled. Clearly, using normal sampling mode of accelerometer loses some data, but saves energy. We will compare these two modes in the evaluation section. All the raw waving action are recorded as a sequence of tuples represented as ðxt, yt, ztÞ, where x, y, z donate the acceleration along the x-axis, y-axis and z-axis respectively, and t donates the time. As a result, we totally collect 600 files containing 389; 373 raw tuples.

Waving Measurement: To show the uniqueness of hand waving in intuition, we display four users' traces in Fig. 1. The traces are illustrated in a 3D acceleration space, short for A-Space, where the raw tuple ðxt; yt; ztÞ are connected in time order. Both the Figs. 1a and 1b are generated from two trails of a volunteer. We can see that the two shapes are very similar. The last three figures come from three distinct persons. Fig. 1c is plot as a circle, Fig. 1d resembles a river, while the shape in Fig. 1e is in the shape of crescent. From these figures, we can observe that the hand waving biometrics are unique for a certain user. A given user presents very simple shape results on different trails. Moreover, different users have clearly different results. The challenge here is how to measure the hand waving represented in A-Space. We should transform the A-Space representation into a parameterized and comparable feature vector. For this purpose, we define the waving function to measure the global geometric properties of the waving shapes, which is formally given by:

f ¼ SðAÞ; (1)

where A ¼ fðxt0; yt0; zt0 Þ; ðxt1; yt1; zt1 Þ . . . ; ðxtn; ytn; ztnÞg. A is a set of raw waving tuples collected during t0 and tn. The waving function considers A as input and outputs a feature vector f. A good waving function should have the following

properties: Efficiency. Since shape function will be performed in the smart phone, it should be simple enough to be fast and efficiently function.

Invariance. In most time, the smart phone is working in mobile environments. The waving function should be insensitive to the position or direction change of smart phones. Robustness. Although the waving data generated by one person is similar, there always exist many noises and the sampling time is variable. Hence, the waving function should be robust to noise, blur, cracks, and dust in the waving. For meeting above four requirements, we propose four waving functions, S1; S2; S3; S4, as follows: _ S1. The centroid C is computed first and then two random points A and B in the A-Space are chosen. The angle ffACB among these three points are measured. The selection of random points is repeated for N times. At a result, N angles output and the corresponding PDF of these angles is reported as the feature vector.

S2. This waving function is similar to the S1. The difference is that all of these three points are randomly selected. One angle among the three angles formed by these three points is recorded. As the result, the corresponding PDF of these angles is given for the feature vector. S3. While both S1 and S2 concentrate on the angle parameter, the other two waving functions, S3 and S4, focus on the distances among the points. S3 randomly selects N points and calculates the Euclidean distance between the centroid and these N selected points. Finally, the corresponding PDF of distances is calculated as the feature vector. S4. Randomly selects N pair of points and calculates their euclidean distance.

Waving Matching: Keeping in mind that our goal is to determine whether the screen should be unlocked according to a given waving action and the pre-defined one. We formalize the similarity of two waving actions by means of the distance between their feature vectors. Since the feature vectors are PDF of distributions, we divide the whole range of PDF into discrete bins and the average value is calculated regarding to each bin. As a result, the discretized

PDF, f ¼ ½p1; p2; . . . ; pn_, is considered the feature vector where pi denotes the probability of falling into the ith bin. Definition 1 (Similarity). Given two arbitrary feature vectors, f1 ¼ ½p1; p2; . . . ; pn_, and f2 ¼ ½q1; q2; . . . ; qn_, their similarity is defined as
Dðf1; f2Þ ¼X
n
i¼1 jpi _ qij;
where Dðf1; f2Þ 2 ½0; 2_. The smaller similarity means two features are very close and vice versa. We select six users randomly and each user conducts three trails. The waving function S4 is employed here to measure the handwaving. As a result, there are 3 _ 6 ¼ 18 features after using by S4. Their similarity are plotted as a visualized similarity matrix in Fig. 3. In the matrix, the darkness of each elements ði; jÞ is proportional to the magnitude of the computed similarity between the ith and jth features. Darker elements represent better matches, while lighter elements indicate worse matches. The matrix is symmetric. Definition 2 (Self similarity). The self-similarity is the distance of two feature vectors extracted from two hand waving generated by a same user. Especially, if the two features come from a same waving instance, they are equal and their similarity equal zero. Obviously, the elements lying in the diagonal line are the darkest because their distances equal 0. For each user, there are 3 _ 3 ¼ 9 elements for self-similarity measurement. From the figure, we can see that the self-similarity always maintains an acceptable darkness and is fully distinguishable from other users' features.

Open sesame: In this section, we present our unlocking method for smart phone called Open Sesame.

Overview: Open Sesame consists of four components: sensing, filter, fetcher, classifier, and matcher. Sensing. This component is straightforward used to record the user's hand waving action data. Filter. In practice, we find that there always exist some silent periods when no waving or very low level sensing data is detected. For better feature extraction, we use filter component to wipe out the silent periods. Fetcher. The filtered raw tuples is feeded into fetcher component in which four waving functions are applied to fetch the waving features. Classifier. To discriminate the authorized users and unauthorized users, the support vector machine is employed in our system for classification. Matcher. In the last component, the extracted feature is used to determine whether it matches the predefined one.

Fetcher: After the filter component, we need to generate the feature vector of the user's hand waving action. According to Section 2.3, the field set of the acceleration points can be treated as one single input of waving function, and the waving function can be applied to this input to generate the feature vector. However, using the field set as an input has two shortcomings. First, the amount of acceleration points in a field set is large, usually more than 1,000. In order to generate a representative feature vector for the waving action data, an extremely large number of feature vectors are required. In this way, the system overhead is high and affects the normal operation of the smart phone. Second, to unlock the smart phone, the user is required to shake his smart phone for a period to generate same amount of waving data. However, it is inconvenient to ask the user to shake the smart phone for such a long time period to generate more than 1,000 acceleration points for each time he wants to unlock his phone. Therefore, the amount of acceleration points selected as an input needs to be reduced.

According to our observation, the waving action of user always shows the property of repeating. In fact, the input waving action can be regarded as a series of small repeating waving actions which are very similar. Therefore, we can select a continuous sequence of acceleration points with a reasonable amount as an input to the waving function. Feature vectors can be generated from these small inputs with low data loss. We generate the feature vectors as follows: we first select a window with size w, where w is much smaller than the size of the field set of data. From the field set of data, we select an acceleration point Pk and form the input with the subsequence of w continuous acceleration points fPk; Pkþ1; Pkþ2; . . . ; Pkþw_1g. Then we apply the waving function on this input and deliver the PDF of the feature vectors to describe the feature of the waving action.

3.4 Classifier The feature classifier is designed to generate a standard to discriminate authorized user and unauthorized users with the feature vectors of the input waving action data. In Open Sesame, the support vector machine, SVM for short, is selected as the classifier. The SVM classifier is used to classify a group of linear-inseparable training tuples into two classes. Training tuples for SVM input is donated as fv; yg, where v is the attribute vector used to describe the attributes of the training tuple, and y is the label of the training tuple, which represents the actual class it belongs to. The basic idea of SVM is to transform these attribute vectors of training tuples into a higher dimensional space to make the training tuples linear-separable. Then the training tuples can be separated into two classes by a hyperplane.

Matcher: The matcher component is performed when the user activates the authentication interface of Open Sesame and wants to unlock the smart phone. The user shakes the smart phone to input his waving action as the authentication data. Feature vectors of the input waving action is generated and used to verify whether the user is the authorized user. If so, the access query is accepted and the smart phone is unlocked. If not, the access query is denied and the smart phone keeps locked. The most important requirement is that the feature matching phase has to be processed

within a short time period, say 1 or 2 seconds. The reason is that users always expect the unlocking process to be fast and convenient. If the feature matching time is long, the inconvenience overweighs the security of our approach and the users may decide to give up our system. To reduce the response time, two aspects need to be considered. The first issue is to reduce the amount of repetition when doing authentication. This can be achieved by reducing the false negative rate (FNR) of authentication, which is going to be discussed in the experiment section. The second issue is to reduce the waving time in the matcher component. As we designed in the fetcher component, by using a small waving function input with window size w, the waving time can be reduced to the time period for collecting w acceleration points. Since w is much smaller than the size of the field set of acceleration points. Therefore, the waving time can be reduced to a tolerant range.

Implementation and Evaluation: In this section, we present the implementation of Open-Sesame and evaluate its performance.

Implementation App: We implement Open Sesame in Android-based smart phones. The version of Android system is 2.3.3. the app is developed with Android-SDK using Java SE.

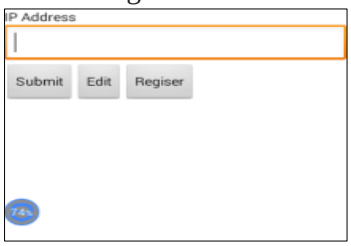

Figure.1.

With this app, the user's hand waving data is collected and analysed by the smartphone. Specifically, the interfaces shown in Figs. 5b and 5c are used to notice whether the unlocking access is success or not. We use the open source library tool, LIBSVM, to perform the classification of SVM. LIBSVM is an integrated software for support vector classification. The version we used is LIBSVM-3.12. During our experiments, we use the default kernel function (Gaussian Radial Basis Function) and find the best setting of parameters Cost and g for the kernel function via the cross-validation when generating the training model.

Metrics: We evaluate OpenSesame in terms of the authentication accuracy. The authentication accuracy is measured via the following metrics: _ False negative rate. The probability that an authorized user is treated as an unauthorized user. This rate is indeed the ratio of the number of incorrect authentications conducted by an authorized user to the number of his authentication attempts. _ True positive rate (TPR). The probability that an authorized user is successfully verified. This rate derived from the ratio of correct authentication times of an authorized user to the number of his authentication attempts. _ False positive rate (FPR). The probability that an unauthorized user is treated as an authorized user. This rate is obtained from the ratio of the incorrect authentication times of an unauthorized user to the number of his authentication attempts. Note that FNR and TPR are related to the convenience of users when they use our system, where the authorized user can successfully unlock the smart phone by a single try. The FPR reflects the security of the Open Sesame, where the unauthorized user should be denied to unlock the smart phone. 4.3 Experiment Setup for investigating the uniqueness of hand waving, we collect the waving action data from 200 distinct smart phone users. The subjects producing these datasets are randomly selected in different public places, including railway station, university library, and stadtpark. When collecting the waving action data, three smart phones from different brands are used. For collecting each specific users hand waving data, he is asked to act with the following instruction: The user first randomly selects one of the three smart phones we provided, and holds this smart phone, which is running our data collection app, in his accustomed way. Then he pushes the button of 'start' on the screen and begins to wave the smart phone until the hint sound is played by the smart phone. This waving process lasts for more than 10 seconds. The user repeats the above action for three times to terminate the data collection. Note that there is no special restriction on users waving actions. He can wave the smart phone arbitrarily in each trail. Indeed, we aim at taking insight into the handshaking action but not the motion pattern.

Overall, 389,373 raw tuples are captured from 200 distinct users, with an average 1,947 raw tuples per user. Each user performs the hand waving for three trails while each trail persists 10 _ 20 seconds. For each user, the training data will be extracted from the first two trails, while the testing data will be retrieved from the last one. Therefore, there is no overlap between the training data and testing data. The classification is based on self and non-self-discrimination. For a given user, the training data is composed of negative samples belonging to this user, and an equal number of positive ones from others.

Impact of Waving Functions: There are four waving functions to parameterize the A Space representation of hand waving. In this experiment, we select 30 users' hand waving and maintains the window size as 50 tuples.

The similar observation is obtained on FPR, as shown in Fig. 6b. This shows that the distance-based waving functions perform better than the angle based ones. We further focus on the distance-based waving functions. S3 and

S4 have close FNRs and FPRs. However, the variance of S4 is smaller than that of S3, which means S4 is more stable than S3. 4.5 Impact of SVM Window size is an important factor. For capturing enough windows, we require the users to shake their phones in an acceptable time period. A large window size will prolong the waving time period for unlocking and seriously affect user experiences. But a small window size will influence the identification accuracy. We change the windows size from 5 to 50 with the increment of 5 and employ S4 for testing. The result is shown in Fig. 7. The average FNR decreases from 20 to 8 percent and the average FPR reduces from 42 to 18 percent as the window size increases. This shows that the larger window helps improve the accuracy. This is because that more raw tuples are extracted in a larger window and the user's hand waving is better characterized. The number of training tuples also affect the accuracy. As illustrated in Fig. 7, FNR is approximately reduced by 50 percent, i.e. from 15 to 8 percent, when window size is 50. This reduction is even obvious with small window size. On the other hand, the average FPR only reduces from 20 to 15 percent taking 5 percent off when window size is 50. This shows that FPR is less sensitive to the number of training tuples.

Impact of User Motion: As mentioned before, our approach should be insensitive to the user's motions because the smart phone is mainly used in mobile environment. Clearly, the user motion will introduce many noises. In this experiment, we test the relationship between the speed of user's motions and the accuracy. Five user's motions are considered: stationary, walking slow, walking fast, running, and taking a vehicle. The result is shown in Fig. 9. From the figure, we can see that as the speed growing from 0 to 5 m/s, the FNR is steady around 11 percent, with a standard deviation of 2:0 percent. This indicates that the motion of users makes a very limited effect on our approach. Besides, the FPR is also invariant when the speed of user's motion increases. The false positive rate is around 15 percent with a standard deviation of 2:5 percent. It can be further obtained from Fig. 9 that, the FNR has an slightly increase, about 7 percent, when the speed of user increases from 0 to 5 m/s. This can be understood because the faster motion will increase vibration in his smart phone leading to more noisy. However, these motions has very limited effect on the accuracy.

Figure.2.

Impact of Phone Diversity: Nowadays, there are plenty of smart phone brands, such as iPhone, MOTO, SAMSUNG, HTC, etc. To promote the OpenSesame to smart phone users, one crucial issue is whether the OpenSesame can be well adapted to different brands of phones. The most effective factor on different smart phones is the type of accelerometer equipped. For different types of accelerometers, the level of sensitivity is different. Hence, the waving data collected is in equivalent.

4.9 Impact of Smart Phone's Orientation although the waving habit may be similar for an identical user, the postures of users when waving the smart phone can change the orientation of the phone. In this section, we evaluate Open Sesame with variant phone's postures. In this experiment, three user's postures are tested: _ standing. Waving phone when standing on the ground. We consider the standing as a normal posture. _ Lying. Waving phone when lying on the bed. The waving orientation is rotated 90 degrees upward. _ On-the-side. Waving phone when sleeping on the user's left side. The waving orientation is rotated 90 degrees to the left. Our approach should be insensitive to the rotation. We transform the waving from A-Space to feature PDF, shown in Fig. 11b, by means of waving function S4. As we expected, the difference of these three PDFs is very slight. In details, the distance between standing posture (the normal posture) and the lying posture/on-the-side posture are 0.172 and 0.173, respectively. We believe these distances are small enough for the trails to be treated as coming from an identical user.

2. CONCLUSION

In this paper, we propose a novel behavioural biometric-based authentication approach called Open Sesame for smart phone. We design four waving functions to fetch the unique pattern of user's hand waving actions. By applying the SVM classifier, the smart phone can accurately verify the authorized user with the pattern of hand

waving action. Experiment results based on 200 distinct users' hand waving actions show that the Open Sesame reaches high level of security and robustness, and achieves good user's experience.

5. ACKNOWLEDGEMENT

This paper is done for MCA Department of Computer Applications Aarupadai Veedu Institute of Technology College Vinayaka Missions University, Salem, India.

REFERENCES

Akkermans AH, Kevenaar TA, and Schobben DW, Acoustic ear recognition for person identification, in Proc, IEEE 4th Workshop Automat. Identification Adv, Technol, 2005, 219–223.

Bonneau J, The science of guessing: Analyzing an anonymizedcorpus of 70 million passwords, in Proc. IEEE Symp. Security Privacy, 2012, 538–552.

Conti M, Zachia-Zlatea I and Crispo B, Mind how you answer me!, Transparently authenticating the user of a smartphone when answering or placing a call, in Proc. 6th ACM Symp. Inf., Comput.Commun Security, 2011, 249–259.

Florencio D and Herley C, A large-scale study of web password habits, in Proc. 16th Int. Conf. World Wide Web, 2007, 657–666.

Jain A, Hong L and Kulkarni Y, A multimodal biometric system using fingerprint, face and speech, in Proc, 2nd Int. Conf. Audio Video-Based Biometric Person Authentication, 1999, 182–187.

LiKamWa R, Priyantha B, Philipose M, Zhong L, and Bahl P, Energy characterization and optimization of image sensingtoward continuous mobile vision, in Proc. 11th Annu. Int. Conf. Mobile Syst, Appl, Serv, 2013, 69–82.

Monrose F, Reiter MK and Wetzel S, Password hardening based on keystroke dynamics, Int. J. Inf. Security, 1, 2002, 69–83.

Park HA, Hong J.W, Park J.H, Zhan Z and Lee D.H, Combined authentication-based multilevel access control inmobile application for daily life service, IEEE Trans. Mobile Comput, 9(6), 6, 2010, 824–837.

Phillips PJ, Martin A, Wilson CL, and Przybocki M, Anintroduction evaluating biometric systems, Computer, 33(2), 56–63, 2000.

Sae-Bae N, Ahmed K, Isbister K and Memon N, Biometric-rich gestures: A novel approach to authentication on multi-touch devices, in Proc. SIGCHI Conf. Human Factors Comput. Syst, 2012, 977–986.

Yampolskiy R.V and Govindaraju V, Behavioural biometrics, Asurvey and classification, Int. J. Biometrics, 1, 81–113, 2008.