Security Enhanced Rhythm Authentication using Relative Finger-Tip Positions

N. Wongnarukane, P. Kuacharoen
Department of Computer Science, Graduate School of Applied Statistics
National Institute of Development Administration, Bangkok, Thailand
nakinthorn.n@gmail.com

***Abstract*—The rhythm authentication algorithm uses the concept of a traditional keystroke authentication and a multi-touch technology based on a touchable device. Three measurements are used in the rhythm authentication method which consists of holding times, latency times, and the number of fingers per beat. These measurements are compared with the user templates. The user is authenticated if a percentage of error is in a predetermined range. However, using only three measurements is not enough. If the attacker is familiar with the victim, the rhythm authentication can be attacked by shoulder surfing or eavesdropping. Additionally, only a percentage of similarity between the user template in database and user's input for classifying is not a reliable algorithm. In this research, we propose a security-enhanced rhythm authentication using relative finger-tip positions and the KNN algorithm for classification to prevent shoulder-surfing attacks.**

***Index Terms*—Rhythm Authentication; Multi-touch; Biometric Authentication; Keystroke.**

I. INTRODUCTION

The rhythm authentication [1] is a novel method of the biometric authentication which was published in 2017. The rhythm authentication uses the concepts of the traditional keystroke authentication and the multi-touch technology based on a touchable device such as a touchpad on a laptop, a smartphone screen, and the like. Three measurements are used in the rhythm authentication method which consists of the holding times, the latency times and the number of fingers per beat. The method uses a percentage of error for classifying the user template. The previous work has weak points because many people are careless when they tap their fingers on the device for authentication. As a result, eavesdropping and shoulder surfing can be a significant threat.

The number of fingers per beat and rhythm feature in the rhythm authentication algorithm can prevent shoulder surfing attacks. However, if the attacker is familiar with the victim, the rhythm authentication may be compromised. The only percentage of similarity between the user template in the database and the user input, which are used to classify the user, is not a reliable algorithm.

In this research, we propose a security-enhanced rhythm authentication using relative finger-tip positions and the KNN algorithm for classification to prevent shoulder-surfing attacks. This paper consists of five sections. The next section provides the background information and related work that are relevant to the paper. The design and implementation of the proposed enhancement will be described in the third section. The fourth section presents the experimental results, and the last section concludes the paper.

II. BACKGROUND AND RELATED WORKS

The rhythm authentication using the multi-touch technology called Rhythmprint is a novel method of biometric authentication. The method can increase security by preventing shoulder surfing attacks and eavesdropping attacks. The Rhythmprint uses the traditional keystroke authentication concept together with the multi-touch technology. The Rhythmprint process consists of two parts, namely, the registration part and the authentication part. The registration part is used for creating a new user template and saving it to a database. The user must tap his/her fingers on a touchable device to make rhythm. Each beat of the rhythm can be produced by one or more fingers. The system records the holding time, the latency time and the number of fingers of each beat. Finally, the user template will be generated and saved into the database. The registration process is shown in Figure 1.

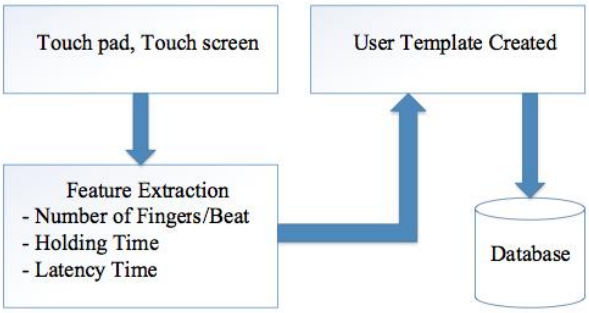

Figure 1: Rhythmprint registration

The authentication process is started when the user needs to access his/her laptop or a touchable device. The user must touch or tap on a touchpad on a laptop or a device screen with the same rhythm which consists of a sequence of beats. The number of fingers per beat must be identical to one which was registered as previously described. The authentication process is shown in Figure 2.

From Figure 2, the system extracts data from the user to create a new template and compares it with the template stored in the database after the user touched or tapped his/her fingers on a touchpad of a laptop or a device screen.

The number of fingers per beat of the new template is compared with the stored data in the database. Subsequently, the algorithm compares the holding time and latency time by defining an error range which is not more than 0.1 second per beat. The experiment result shows that the Rhythmprint can prevent shoulder surfing attacks because it does not have any target button to press. The user can touch or tap his/her fingers

on a device without looking at the touchpad or screen. Therefore, the user can use the other hand to cover while tapping on a touchpad or touchscreen.

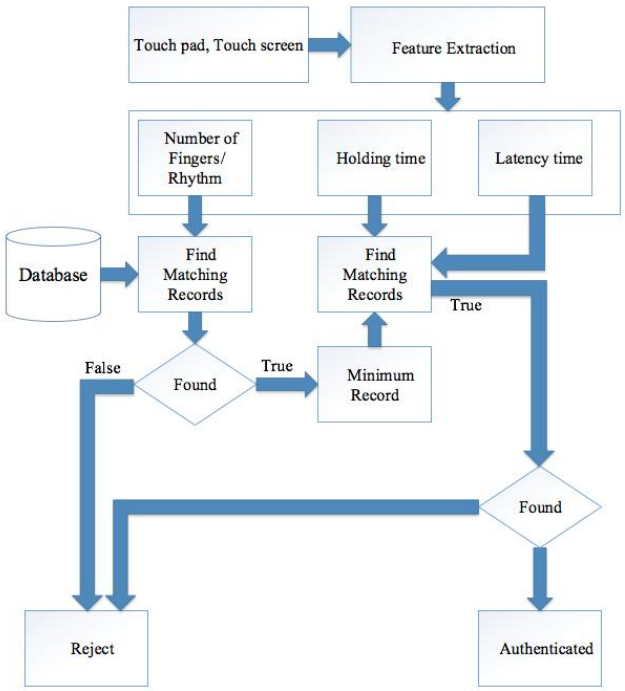

Figure 2: Rhythmprint authentication

Eavesdropping attacks are rarely successful because when the user taps his/her fingers on a touchpad or touchscreen, it does not make any sound. Although some users are not careful when they tap their fingers on a device and the tapping sound can be heard, the attacker does not know the number of fingers in the sequence of beats that user taps on the device.

The Rhythmprint provides stronger security. However, there is a weak point for shoulder surfing attack prevention. When the user taps on a device for authentication, many users do not use the other hand to cover the hand which is tapping. If the attacker is familiar with the user and the user's rhythm is easy to detect, the attacker will be able to learn the number of fingers per beat by shoulder surfing.

Before the Rhythmprint was proposed, many algorithms and methods of the keystroke authentication on a touchable device were published. The keystroke authentication method on a smartphone with a touchscreen device using a virtual keyboard was proposed, Huang et al. [2] use password by the input button, holding time, latency time because the user cannot use the other hand to cover the touching hand. It can be easily attacked by shoulder surfing. It is not different from the traditional keystroke authentication as it only changes from the button keyboard to the virtual keyboard.

In 2015, a keystroke authentication on a smartphone by using a password input, holding time, latency time and pressing pressure was published [3]. This method has weak points as the previous work. However, it includes pressing pressure. In real life, the pressures cannot be accurately used as an authentication attribute since the user makes different pressures while doing different activities such as sitting, standing or lying down.

PassChords is a multi-touch authentication using the touch screen on iPhone and other features including the number of fingers (limit to four fingers) and the fingers [4]. First, the login process starts when the user places four fingers, specifically, fore, middle, ring and little fingers, on the screen. The system will detect which fingers are pressing by using the location on the screen. The systems will respond to the user by vibration if all four fingers tap on the screen. Second, the user must tap his/her fingers on the screen four times by trying to press each finger in the same location. The system will count the number of fingers that press down on the screen each time. Finally, the user is authenticated if the sequence of tapping and the number of fingers per tap are identical to the one stored in the database. PassChords was developed for blind people, who lock/unlock their smartphone by VoiceOverPINs, and the authors claimed that it is no longer secure. The experiment shows that PassChords are nearly 75% faster than VoiceOverPINs and more secure because PassChords method does not produce audio feedback and does not show anything on the screen. Therefore, aural and visual eavesdropping is more difficult. However, it is easy to detect the correct number of fingers per tap. This is not different from the traditional password. The attacker remembers only how many fingers were tapped each time just like a digit password. Although the system can take more than four taps for the authentication, it is not different. This method is very easy to use, but it is a high risk as the digit password.

IR Ring [5] is an amazing method. It proposes how to prevent the touch or input on the multi-touch display from an unauthorised user. If the user needs to touch fingers on the multi-touch display, the user must wear a ring called IR Ring. IR Ring was developed from a small circuit board with an infrared module. IR Ring identifies the user who is wearing it and returns the location of the hand on the screen. Only the person wearing the IR Ring can input actions on the screen. Although this method is pretty good, it is not different from USB token or NFC tag (what you have factor). The ring may be lost or stolen by criminals, and it is costly. Furthermore, the ring needs to be recharged.

The tap authentication on the smartphone can be performed more quickly, and it is more secure than other methods if there is no visual feedback from the screen when hiding the device under the Table [6]. The authors researched the use of draw pattern, PIN and tap authentication methods in their experiment. The experiment was split into two parts. In the first part, the user could look at the screen when he/she tried to authenticate with three methods (draw pattern, PIN and tap) for 30 times per method. In the second part, the user could not look at the screen, when he/she tried to authenticate with three methods the same as first part for 30 times per method. Shoulder surfing attacks were conducted while the users are performing authentication. The results of the experiment for both parts are shown in Table 1.

Table 1 shows that the tap authentication can be performed in the least amount of time if the user does not look at the screen (no visual). The experiment shows that if the user tries to authenticate his smartphone in public (visual), the attacker can successfully replicate victim's pattern and code after only five times of shoulder surfing attacks for drawing a pattern or tapping to unlock the device. However, the attacker can learn the user's PIN after shoulder surfing for only nine times. Although the authentication on the smartphone without visual feedback from the screen increases security and prevents shoulder surfing attacks, more errors occur. Therefore, hiding the device under the table is impractical.

Table 1
The Completion Time of Unlock Task and Error

Method	Time (visual)	Time (no visual)	Error (no visual)
Draw Pattern	2.81s	30.32s	2.84 times
PIN	3.82s	43.14s	3.53 times
Tap	3.73s	6.18s	0.42 times

Table 1 also shows that the tap authentication method has the lowest error rate when compared with other methods. The tap authentication in their research uses a simple keystroke pattern like PassChords, but it does not include the multi-touch feature. When the user presses/releases a finger on the screen, the system converts the press action to 1 and release action to 0 and adds up all holding times. The authors use the Hamming distance for the classification.

However, the author proposed that their authentication method is effective for being fast and easy to use, but it has a very high risk. In real life, people rarely authenticate themselves using their smartphone under the table every time. A shoulder-surfing attack can be successfully performed when the user authenticates in public only five times or less.

BoD is a back-of-device authentication based on portable touchscreen device [7]. The user can choose one of two candidates: "BoD Pattern Unlock" (draw the pattern to unlock) or "BoD Shapes" (draw the shape to unlock) features for the authentication service. When the user needs to unlock the phone, the user must draw the pattern or the shape at the back of the phone for authenticating.

The back of the phone must be embedded with a portable touchscreen device and connected to the smartphone by wireless. This research is successful in preventing shoulder surfing attacks, but it has a high false acceptance rate (FAR) because drawing on the back of the smartphone is very confusing and requires a high level of concentration.

To improve the performance of BoD, Leiva and Català [8] proposed a method to resolve a FAR by adding a new feature called "BoD Tabs". The user only has to tap his/her finger on the back of the phone to unlock. It is simple and able to reduce FAR and prevents shoulder surfing attacks. Although BoD and BoD Tabs can prevent the shoulder surfing attacks, they require a peripheral device, specifically a portable touchscreen. Furthermore, it is impossible to use in real life. Nowadays, people need their smartphones to be thin, lightweight and have a long life battery. Peripherals should be already embedded from the manufacturer.

In 2013, the pin authentication scheme allowing multi-touch key input was proposed [9]. This method combines PIN and the multi-touch feature of the smartphone. They use PIN (4 digits) for the authentication by using a virtual numeric keyboard (0-9) as a traditional PIN, but they allow the user to use more than one finger when touching the digit button on the screen. Their experiment shows the time of the single touch is lower than the multi-touch. The method is simple and easy to use but does not increase security because the way to unlock is so slow. The users have to see the numbers on the screen when they need to unlock their phone, and they must carefully input the correct PIN and correct pattern. This method also has a high risk of shoulder surfing attacks.

K-Nearest Neighbors (K-NN) is widely used in the pattern recognition because it is simple and easy to understand [10]. The principle of K-NN is to compare the similarity between the interesting data with the set of stored data to find what the class of the interesting data should be. If the distance between interesting data and the stored data is lowest, the interesting data are in the same class as the stored data. Euclidean Distance algorithm is used to find the distance matrix of K-NN as the following equation.

$$distance = \sqrt{\sum_{k=1}^{n}(p_k - q_k)^2}$$

where: p = Point of data of interest attribute
q = Point of data set in the same attribute
n = Number of attributes

After calculating the distance metric, we can answer what the class of the interest data from the minimum distance.

III. Design and Implementation

Our proposed method called Rhythmprint Authentication includes the concept of Hand Geometry. The objectives are to provide better security, usability, and faster authentication to the rhythm authentication using multi-touch technology proposed earlier. We design a new user template to include four attributes:

- Distance between fingers
- Holding time
- Latency time
- Number of fingers

We add a new attribute called 'distance between fingers' to the user template. The distance between fingers is applied from hand geometry [11]. According to research, each person has different hand geometry. Therefore, the distance between fingertips of the user is different when his/her fingers touch on an input device.

The next attribute is the holding time. It is the time when the user pushes his/her fingers on the input device and when the user releases his/her fingers from the input device.

The third attribute is the latency time. It is the time when the user releases his/her fingers from the input device and when the user pushes his/her fingers on input device again. The last one is the number of fingers which are used to tap. To verify our method, we designed an experiment including two processes which are the template creation process and authentication process.

A. Template Creation

The user must create his/her own template to use for authentication. The flow of user template creation method is shown in Figure 3.

When the user enters each beat to create the rhythm, we collect the holding time, latency time, number of fingers and distance between fingertips. The number of fingers and distance between fingertips per beat is very important because the holding time and latency time can distinguish only the person who creates his/her owner of the rhythm. However, different users may choose the same rhythm. If the attacker can only hear when the victim enters the rhythm to authenticate, the number of fingers and distance between fingertips protect the user's rhythm. Figure 4 shows the step of the template created for the first beat of the rhythm and how we collected the data for each beat.

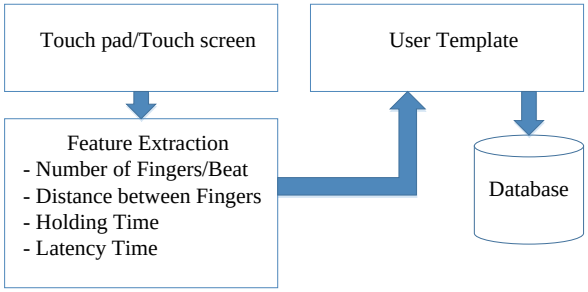

Figure 3: The user template creation process

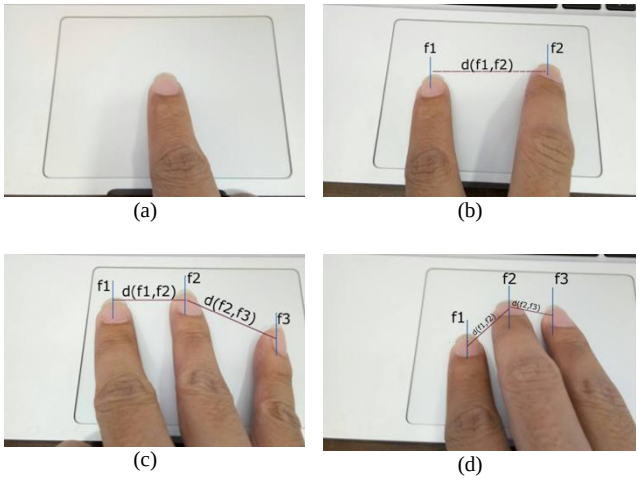

Figure 4: The user attempts to log in to the system.

Figures 4(a) - 4(d) show when the user enters his/her fingers on the touchpad to create the first beat by a different number of fingers and different fingers. In Figure 4(a), the user presses only one finger on the touchpad to create one beat. In Figure 4(b), the user presses two fingers on the touchpad to create the first beat. In Figure 4(c), the user presses three fingers on the touchpad to create the first beat. In Figure 4(d), the user also presses three fingers on the touchpad to create the first beat. The required values are collected. In table 2, the data for the first beat from the Figure 4 (a)-(d) are created as soon as the user touches the touchpad.

Table 2
The Sample of Second Beat Creation

Method	Value (Figure)			
	4a	4b	4c	4d
Holding time (ms)	0.053	0.083	0.072	0.063
Latency time (ms)	0	0	0	0
Number of fingers	1	2	3	3
Distance between finger tips	0	{{0}, {1.34}}	{{0},{1.30}, {1.23}}	{{0},{1.30}, {2.11}}

For the first beat (first touch), latency times are zero. However, holding times, the number of fingers, and fingers are different. In Figure 4 (c) and (d), the numbers of fingers are the same, but fingers are different. Therefore, the distances between fingertips are different. Table 3 shows the data for the second beat. Latency times are the times between the first beat and the second beat.

Table 3
The Sample of Second Beat Creation

Method	Value (Figure)			
	4a	4b	4c	4d
Holding time (ms)	0.067	0.096	0.101	0.072
Latency time (ms)	0.096	0.087	0.088	0.071
Number of finger	1	2	3	3
Distance between finger tips	0	{{0}, {1.36}}	{{0},{1.28}, {1.24}}	{{0},{1.34}, {2.06}}

After collecting the data, we can create the user template and save it to the database.

B. *Authentication*

The user can access his/her device such as a laptop, smartphone and other touchable devices by tapping his/her fingers on the touchable input of the target device with the same rhythm, the same sequence of number fingers per beat and the same hand to create a new template, namely a login template. The login template will be used to compare to the templates in the database by using K-NN algorithm. Figure 5 shows the flow of authentication process.

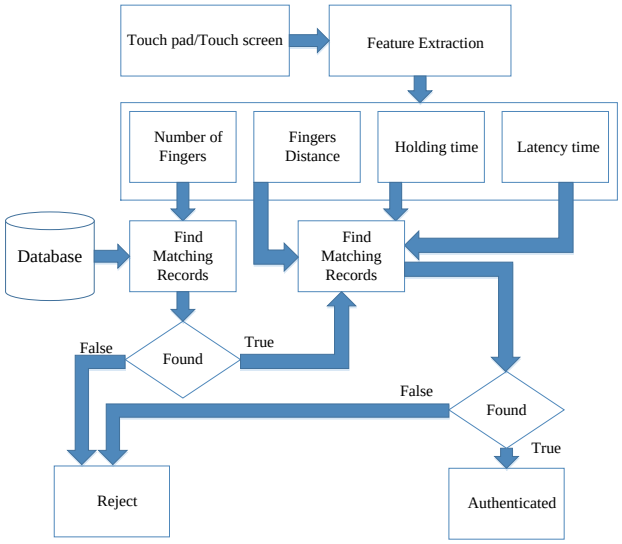

Figure 5: The authentication process

C. *Classification*

The authentication process to find a matching record in figure 5 can be separated into two steps. First, when the feature extraction is completed, we will obtain four attributes from the user input including the number of fingers, the distance between fingertips, holding time and latency time per beat. The number of fingers per beat can be used to classify records in the database directly.

After classifying, we will obtain the minimal records. Second, the system will classify the login template by using the holding times, latency times and the distance between fingertips against the template in the minimal records from the previous state using K-NN algorithm. If we can find the minimum distance calculated from Euclidian distance algorithm, the user is authenticated.

IV. Experimental Result

To verify our proposal, we designed and developed software using our algorithm on the laptop by using Java programming language for the software development. Mac Book Pro by Apple Inc. is the laptop in our experiment. We

recruited 100 participants to test our program consisting of 34 men and 66 women whose age is between 16-61 years old. We used four measurements which are the holding times, latency times, number of fingers/beat and distances between fingertips to create the user template and K-NN to classify. FAR and FRR will be recorded in this phase.

In the experiment, we asked the users to create the rhythm with our fix rhythm to prove whether the distance between fingertips (hand geometry) can improve the performance of rhythm authentication. Our fix rhythm was created from 10 beats. We tapped this rhythm on the table to make an example of sound for users to listen ten times. After that, the user could use one or more fingers per beat for creating similar rhythm like our example rhythm.

Each user must enter our fix rhythm for ten times with one's own user's sequence and some fingers per beat. After all, users already created the template; we would have 1,000 records of 100 user templates in our database.

After obtaining 1,000 templates of 100 users, we selected 50 of 100 users to authenticate oneself by trying to access our laptop with our fix rhythm for ten times. The result of the user authentication for 500 times of 50 users is shown in Table 4.

Table 4
The Fix Rhythm Authentication Result

FAR	FRR	Valid
0.4%	3.2%	96.4%

The result of FAR and FRR in our experiment was showed in Table 3. The percentage of the correctness is 96.4% of valid authentications and only 3.2% of FAR and 0.4% of FRR.

V. CONCLUSION

The Rhythmprint authentication with the relative finger-tip distance (hand geometry) will increase the security of Rhythmprint. It can defend shoulder surfing attacks, and it is more convenient to use. This is because users do not need to use the other hand to cover the tapping hand to prevent shoulder surfing attacks. Although attackers can see which fingers and number of fingers that the user taps for each beat, the attackers cannot impersonate the user. In the experiment, we asked all of the volunteers to use the same rhythm to create their templates, but the FAR is still low. The shape and size of fingers are unique to an individual. As a result, the distance between fingertips when making taps are unique. Eavesdropping attacks are unlikely because tapping on the touchpad does not make loud sounds. For these reasons, we can conclude that the Rhythmprint authentication with hand geometry provides strong security than Rhythmprint. Even if the attacker knows the rhythm, the number of fingers per beat and which the fingers used for each beat, the feature of distance between fingertips cannot be easily copied due to the hand geometry. Moreover, the result of the experiment in section 4 shows that if users were asked to use a predefined rhythm for creating their template, the FAR and FRR are still low. However, if we allow the user to use his/her own rhythm, the percentage of accuracy score will be lower than the results in Section 4. This is because when the user taps on the touchable device with a poor rhythm, he/she must have to think about the number of fingers of the next beat all the time until the last beat. The fix rhythm makes the user do something unnaturally. Nevertheless, when the user chooses his/her own good rhythm and is familiar with it, the false acceptance and false rejection will be unlikely.

REFERENCES

[1] N. Wongnarukane and P. Kuacharoen, "Rhythm Authentication Using Multi-touch Technology: A New Method of Biometric Authentication," *Lecture Notes in Computer Science (LNCS) Springer, Cham.*, 2017, pp. 390-399.

[2] X. Huang, G. Lund and A. Sapeluk, "Development of a typing behaviour recognition mechanism on Android", In *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*. *IEEE*, June 2012.

[3] M. Antal, Z.S. László and L. Izabella, "Keystroke dynamics on android platform," *in Procedia Technology*, 2015.

[4] S. Azenkot, K. Rector, R. Ladner and J. Wobbrock, "PassChords: secure multi-touch authentication for blind people," in *Proc. of the 14th international ACM SIGACCESS conference on Computers and accessibility*, October, 2012.

[5] V. Roth, P. Schmidt and B. Güldenring, "The IR ring: authenticating users' touches on a multi-touch display," in *Proc. of the 23nd annual ACM symposium on User interface software and technology*, October, 2010.

[6] D. Marques, T. Guerreiro, L. Duarte and L. Carriço, "Under the table: tap authentication for smartphones," in *Proc. of the 27th International BCS Human Computer Interaction Conference*, September, 2013.

[7] A. De Luca, E. Von Zezschwitz, N.D.H Nguyen, M.E. Maurer, E. Rubegni, M.P. Scipioni and M. Langheinrich, "Back-of-device authentication on smartphones," in *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, April, 2013.

[8] L.A. Leiva and A. Català, "BoD taps: an improved back-of-device authentication technique on smartphones," in *Proc. of the 16th international conference on Human-computer interaction with mobile devices & services*, September, 2014.

[9] T. Takada and Y. Kokubun, "Extended pin authentication scheme allowing multi-touch key input," in *Proc. of International Conference on Advances in Mobile Computing & Multimedia*, December, 2013.

[10] D. T. Larose, "Discovering Knowledge in Data: An Introduction to Data Mining," *John Wiley & Sons, Inc., Hoboken, NJ, USA*., 2004.

[11] D. Zhang and V. Kanhangad, "Hand Geometry Recognition," *Encyclopedia of Cryptography and Security Springer US*, 2011, pp. 529-531.