Rhythm Authentication Using Multi-Touch Technology: A New Method of Biometric Authentication

Nakinthorn Wongnarukane

A Dissertation Submitted in Partial
Fulfillment of the Requirements for the Degree of
Doctor of Philosophy (Computer Science and Information Systems)
School of Applied Statistics
National Institute of Development Administration
2019

Rhythm Authentication Using Multi-Touch Technology:
A New Method of Biometric Authentication
Nakinthorn Wongnarukane

..... Major Advisor
(Assistant Professor Pramote Kuacharoen, Ph.D.)

The Examining Committee Approved This Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy (Computer Science and Information Systems).

..... Committee Chairperson
(Associate Professor Surapong Auwatanamongkol, Ph.D.)

..... Committee
(Assistant Professor Pramote Kuacharoen, Ph.D.)

..... Committee
(Associate Professor Ohm Sornil, Ph.D.)

..... Committee
(Associate Professor Pipat Hiranvanichakorn, Ph.D.)

..... Dean
(Assistant Professor Pramote Luenam, Ph.D.)

_____/_____/_____

ABSTRACT

Title of Dissertation	Rhythm Authentication Using Multi-Touch Technology:A New Method of Biometric Authentication
Author	Nakinthorn Wongnarukane
Degree	Doctor of Philosophy (Computer Science and Information Systems)
Year	2019

A keystroke authentication method has a relatively low cost, yet it is more powerful and easier to use than other biometric authentication methods. However, traditional keystroke authentication has many weaknesses and is easily exploited by malicious actors. Malicious attacks can include shoulder surfing attacks, eavesdropping attacks and key-logger attacks. When users try to access their computer or portable device by using a keystroke authentication method, the users must push the correct buttons with the correct rhythm in order to be authenticated. If the users make several failed authentication attempts, the system will lock their account. This results in, the users often employing a simple password and rhythm for accessing their account which further increases the risk of a malicious attack.

This research proposes a new method of a biometric authentication by using multi-touch technique on a touchpad which is embedded on a laptop computer combined with the concept of traditional keystroke authentication. The users can register their rhythm using their fingers on the touchpad as a biometric authentication method. An attacker will have difficulties conducting a shoulder surfing attack this is because the user has no need to type in their password and can use one hand to cover the other hand which is used to make their rhythm for the touch authentication. Furthermore, the process is extremely fast, thus further reducing the chance of a shoulder surfing attack. An eavesdropping attack is also rendered useless since the touchpad can register touch data which is inaudible. Even though some users may not be vigilant and make tapping sounds, an eavesdropper cannot know how many or which of fingers were used on the touchpad to make one beat.

The research results show that the proposed multi-touch rhythm authentication performs better than the traditional keystroke method and provides better security, usability, and faster authentication.

ACKNOWLEDGEMENTS

I would like to express my special thanks of gratitude to my advisor Assistant Professor Dr.Pramote Kuacharoen who gave me a golden opportunity to do this wonderful research on the topic Rhythm Authentication Using Multi-Touch Technology, which also helped me in doing a lot of Research and i came to know about so many new things I am really thankful to them. Secondly i would also like to thank my parents and friends especially Mr.Pokkhet Ratchakitprakarn who helped me a lot in finalising this project within the limited time frame.

Nakinthorn Wongnarukane

November 2019

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER 1 INTRODUCTION

Authentication is one of the most important issues in computer security management and information systems. Many countries have legislation regarding computer crime to control and regulate their citizens. Many criminals are usually arrested by tracing back to their IP address which is used when committing the crime. The criminals often use the victims' computers or accounts (username and password) to access social media such as Facebook and Twitter to conduct illegal activities. Therefore, the computer owners and the account owners must be aware of this issue and defend themselves against criminals who try to hack, attack, and use their computers to commit crimes. For this reason, many computer authentication methods are developed to detect and defend against the criminals from unauthorized access. The identity verification consists of three main methods (Kim and Hong, 2011: 188).

1) what you know (the secret, only the owner knows such as password and PIN).

2) what you have (the equipment that is used to access something such as an ID-card)

3) what you are the singularity, only owner has such as a fingerprint and their unique signature

From the above three methods, the mostly used method is password-based authentication (what you know) because it is the easiest and most convenient to use. Nowadays, a majority of web application systems such as Google deploy single sign-on (SSO) where the users sign in once and are able to use all products associated with their accounts (Mainka and Christian, 2015: 117-131). It is easier for the users to remember only one password. But, this system is susceptible to a higher risk from criminal attacks since the compromisation of one password results in multiple systems associated with the password being breached at the same time. One Time Password (OTP) (Subpratatsavee and Kuacharoen, 2015: 93-98) may be used to mitigate some of the risk as a two-factor authentication. However, some implementation of OTP has

already been hacked. Alternatively, a possession method can be used (what you have). It is based on items that users possess such as ID cards, RFID cards, and USB tokens. However, this method also has its problems, namely the fact that the users have to carry the authentication items around, that may be lost or stolen. Criminals may be able to obtain an authentication item which may lead to a security breach. The last method is based on what the users are, in the form of biometric data such as fingerprint, handwriting signature, or characteristics of the user behaviors.

Biometric authentication relies on the unique biological characteristics of individuals to determine identity which consists of two methods, namely, physical biometric and behavioral biometric. The physical biometric uses human body parts such as fingerprint, retina, face, DNA, and hand geometry while the behavioral biometric uses measurable patterns in human activities such as keystroke dynamic, voice ID, and gait analysis. The biometric authentication that is widely used is fingerprint authentication (Kant and Rajender, 2010: 1-9). Fingerprints can be used to access a personal computer, open a gate, access a smartphone, and much more. However, fingerprints can easily be forged or copied (Bhattacharyya, Ranjan, Alisherov and Choi, 2009: 13-28) which makes it unsafe to deploy by itself. Other biometric authentications are retina authentication (Galbally, Cappelli, Lumini, Gonzalez-de-Rivera, Maltoni, Fierrez and Maio, 2010: 725-732) hand-gesture authentication, heart rate authentication, human body print authentication as well as the characteristic behavior such as movement and keystroke authentication. These types of authentication are more secure than fingerprint authentication however, they are more expensive. If we need better security, we may have to pay more for the device. Some devices may not be convenient to use in real life. For example, retina authentication devices are very expensive and may be subjected to infection given the proximity between multiple eyes and the device.

Keystroke authentication (Maiorana, Campisi, González-Carballo, and Neri, 2011: 21-26) may be an alternative since it is not costly and easy to implement. Using the keystroke authentication, the system detects the rhythm of the users pressing and releasing their fingers on the keyboard. Keystroke authentication does not require any additional peripheral devices. Although the keystroke authentication is less expensive and easier to use than other methods of biometric authentication, it is susceptible to

shoulder surfing attacks. For example, if an attacker has enough time to detect the victim's rhythm and password, the attacker can impersonate the victim. Every authentication method has advantages and disadvantages depending on the situation and how it is deployed. This research presents another method of authentication and verification for users by using a touchpad to produce a rhythm print as the biometric authentication method. Doing so we are trying to solve the weaknesses of the traditional keystroke authentication without increasing cost while retaining its ease of user.

This paper describes how to use rhythm authentication without using a password. Since many devices have an embedded touchpad or a touch-screen and support a multi-touch system, they can be used as an input device for authentication. The multi-touch system is used to detect the rhythm and the number of fingers that is used to press and create a rhythm when they tap their fingers on a touchpad or screen of a smartphone. Attackers would have difficulties trying to conduct shoulder surfing attacks because users do not expose their fingers during tapping by covering the tapping hand with the other hand. Tapping actions can be made so that it does not produce any sound as lightly touching on the touchpad or the touch-screen is enough thus making it difficult for attackers to perform an eavesdrop attacks. Even if a tapping action produces audible sound, the attacker has no way of knowing how many and what fingers were used to produce the sound.

1.1 Objective of the Study

To propose a new method of biometric authentication that performs better than the traditional keystroke method, while providing higher security, usability, and faster authentication.

To compare the expected security of the proposed method with another biometric authentication based on keystroke authentication and to apply the proposed method to real-life computer/device authentication.

1.2 Scope of Study

This study focuses on how to develop a new method of biometric authentication in the part of behavioral biometric by applying the multi-touch input

feature on devices such as laptops and smartphones. The limit of this research is to allow only ten fingers to touch the touchable device at the same time.

CHAPTER 2 RELATED WORKS

2.1 Biometric Authentication

The term biometrics is made up of the word Bio, which means living things and the word Metrics, which means features that can be measured or evaluated. When bringing the meaning of both words together Biometrics means technology using certain features or behaviors of living things that are unique features (Aleksandra, 2012: 2-3). Biometric is a combination of biological and medical technologies and computer technology. By measuring individual physical characteristics and behavior characteristics that are unique to each person for identification purposes. The physical characteristics of most people will not change over time, while human behavior may change over time, whether it is speech, signature, keyboard use. Thus, authenticating a person using their physical characteristics is more reliable However, the use of physical biometric systems is at risk of infection because the vital organs must be attached to the authentication device. On the other hand, using biometric behavior types, results in a low risk of infection because it does not require sensitive organs (such as eyes) to touch the authentication device. Biometric can be divided into two major categories, namely

- Physiological Biometrics, for example
    - Fingerprint
    - Facial Recognition
    - Hand Geometry
    - Finger Geometry
    - Ear Shape
    - Iris and Retina in the eyes
    - Human Scent
- Behavioral Biometrics, for example
    - Keystroke Dynamics
    - Gait Recognition
    - Voice Recognition

- Signature

The process of checking or identifying a person with biometric identification, regardless of the use of any particular style, will have the same procedure as follows.

- The user of the system must provide samples of the biometric characteristics to be used, or the initial registration before using the system.
- Biometric samples that were collected in the first step will then be converted and stored as a template to be used for comparison.
- When the user wants to use the system, it will be checked by comparing the biometric data of the user with the stored template, checking the similarities and measurements from the examination.

The results of the examination or identification of this user can be made in four cases:

- Correct Accept: allow users with access to the system.
- Correct Reject: deny those who do not have the right to use the system.
- False Accept: allows people who do not have access to the system for the number of False Accept if calculated in percentage, this will be called False Accept Rate (FAR)
- False Reject: Deny users with system rights. The number of False Reject if calculated in percentage is called False Reject Rate or FRR.

The advantages of using biometrics are:

- No need to use password recognition or hold any pass cards, making it convenient, fast and also help increases security by preventing loss of pass cards, or stolen passwords.
- Difficulty to counterfeit and steal Biometric data makes users unable to deny responsibility. For example, in the case of using a password or a pass card through the owner of the card, it may be claimed that the password or card were illegally used by a malicious actor. By using biometric authentication or identification, users cannot deny responsibility.

One of the most popular of biometric authentication method is fingerprint. The human fingertips generally see fingerprints that are composed of two lines of ridges and furrows, in which both lines are alternated throughout.

Characteristics on fingerprints and various marks on the fingerprint, can be divided into two types as follows:

- various characteristics of common lines such as straight lines, curves, points, cracks, cross lines, circular lines, broken lines, lake lines, two lines meet
- Some special characteristics such as
    - Bifurcation is one border that has been separated into two lines or more than two lines.
    - Divergence is a border that runs parallel to or almost parallel and splits out.
    - Minutiae point is the point on the end of the line, stop or split line

The major features that show the differences of each fingerprint, are 4 items, namely:

- Type Line
- Delta
- Core
- Pattern Area

Figure 2.1 shows the different points of a fingerprint.

Figure 2.1 The point name fingerprint pattern

Fingerprint analysis of a general person, begins by analysing a person's finger, to find important and unique characteristics. The first process is reading the fingerprint and storing it into a database. The data that was read or scanned must first be processed before being stored. This data / information will be stored as a model or code for each user. The process of pre-processing will make the image that has been scanned more completed because the machine scanned image, are sometimes unclear. So, in this process, many actions are performed, namely eliminating interference, adjusting the darkness, brightness, converting the images into two levels (0 and 1), thinning out the lines, adjusting the images after converting the images into two levels, evaluating the threshold of the image adjustment in two levels etc. The extent of the preprocessing more or less depends on the quality of fingerprints that was obtained; naturally more preprocessing is required for lower quality samples.

Once the fingerprint has been processed, it will then be stored in a database. The images or template that was stored on the database will then be compared with fingerprints that are being scanned when the user tries to authenticate using a fingerprint scanning device. Figure 2.2 shows the process of template creation and storing of the result on a database.

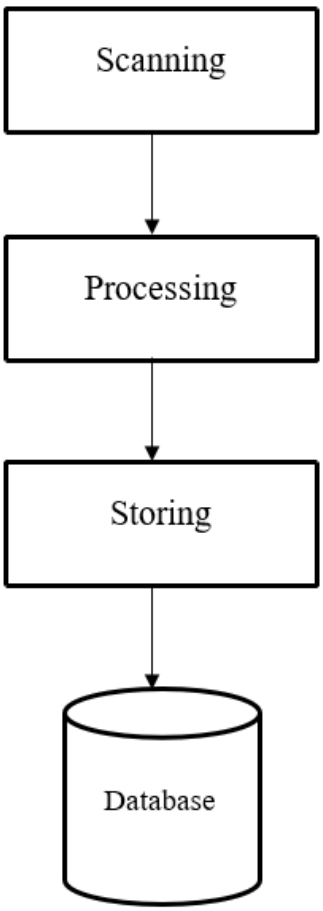


Figure 2.2 The process of fingerprint template creation

When the user needs to be using the system, which requires fingerprint authentication. The user must scan his/her finger on the scanner device. Figure 2.3 shows the process of template or image comparison between the scanner and the template or image from the database.

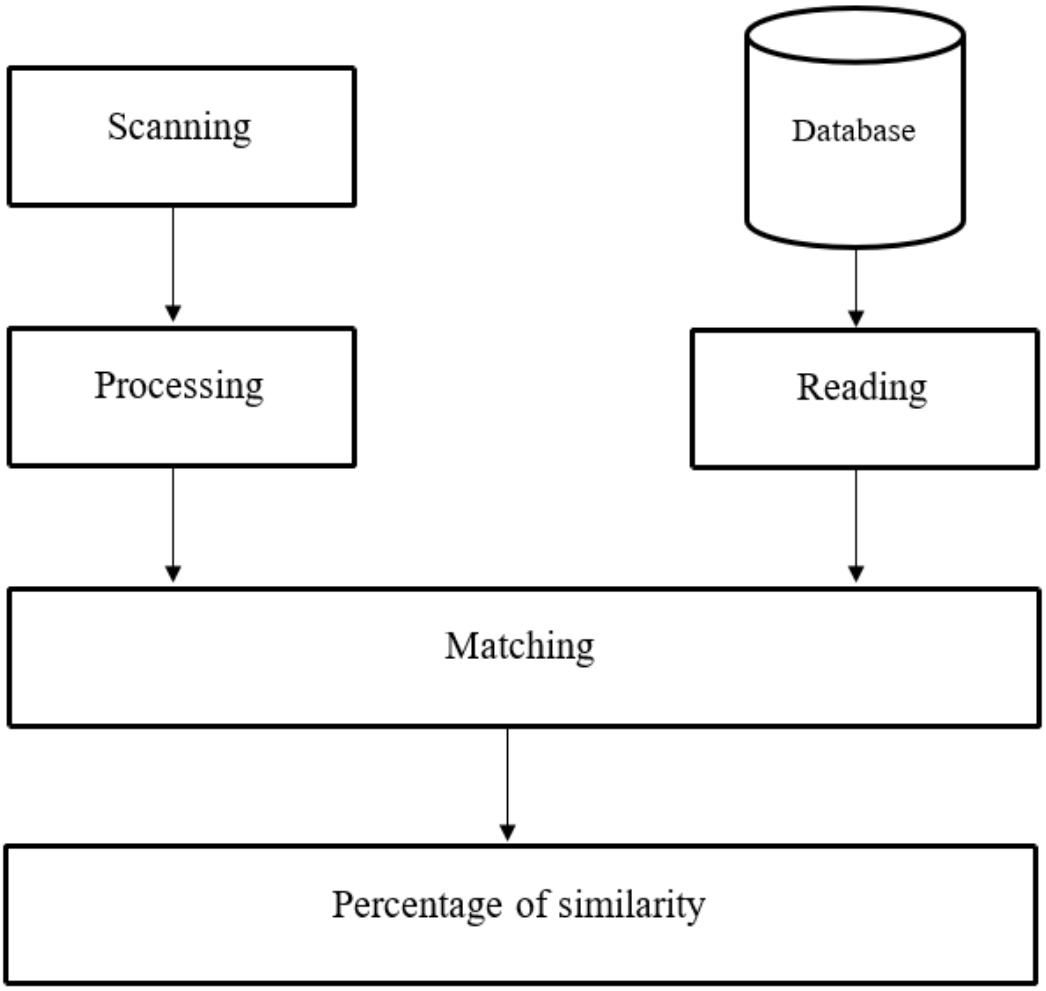


Figure 2.3 The process of fingerprint authentication

From Figure 2.3 shows the process of comparing fingerprints that have been scanned by the scanner device and replicate the process of template creation. the fingerprint images or templates from database will fetched and compared with the scanned image; the characteristic differences percentage are then calculated.

In addition to fingerprints, identity verification is also possible with other biometric authentication types that derives data from the use of human organs such as Hand Geometry, Finger Geometry, Iris and Retina. But at present, such authentication methods are no longer safe because the attacker can easily attack the system in a short amount of time. For example, fingerprints can easily be faked in many ways (Galbally, Cappelli, Lumini, Maltoni and Julian Fierrez, 2008: 1-4). Therefore, the system developers need to add extra measures to increase the system security (Espinoza,

Champod, and Margot, 2011:41-49) such as the use of passwords along with the use of fingerprints. Facial Recognition, which uses the camera to compare the template in the database with face detected from the camera, are still prone to errors, for example, the analysis of the twin brothers.

In addition, the human body naturally changes over time, such as the nature of the sound that changes throughout one's life, or the fingerprints or hands that changes due to accidental damages. Therefore, a physical security biometric authentication system is often used in conjunction with other authentication systems for accuracy, this is called two factor authentication.

When compared with Rhythmprint authentication, which is an algorithm to verify identity in the form of two factor authentication within itself. When the user needs to use the system, the user must tap the rhythm on the touchpad with the correct rhythm, with the correct hand, with the correct fingers and with the correct number of fingers used per beat to touch each stroke.

For this reason, it can be defined that identity verification in the form of physical security biometric authentication (what you are?) are not as secure compared to Rhythmprint authentication, which consists of three main methods of identity verification (what you are, what you have, what you know).

2.2 Keystroke Authentication

Keystroke authentication measures the manner and rhythm in which each individual touchtype. The user must enter their password with the correct rhythm to verify themselves (Shanmugapriya and Padmavathi, 2009: 115-119). The methodology of keystroke authentication is simple. The user must type his/her password on a keyboard with the previously registered rhythm, the system extracts features of the user password and rhythm, and then compares them with user's template in the database. The performance of biometric authentication can be measured in three factors namely:

1) False Acceptance Rate (FAR)
2) False Rejection Rate (FRR)
3) Equal Error Rate (EER)

FAR is the measure of the possibility that the biometric system incorrectly permits an access attempt by an unauthorized user. FAR is typically calculated by dividing the number of false acceptances by the number of identification attempts. In contrast, FRR is the measure of the likelihood that the biometric system incorrectly denies an access attempt by an authorized user. FRR can be computed by using the ratio of the number of false rejections and the number of identification attempts. When FAR and FRR are equal, it is called EER. For creating the keystroke template, various measurements can be taken. However, these are the common measurements, namely, holding time, latency time, and pressure. The time which starts when users push their fingers on the button of the keyboard and ends when they release their finger from the button, is called holding time. The time when users switch their finger from the current button to a new button call is latency time as shown in Figure 2.3.

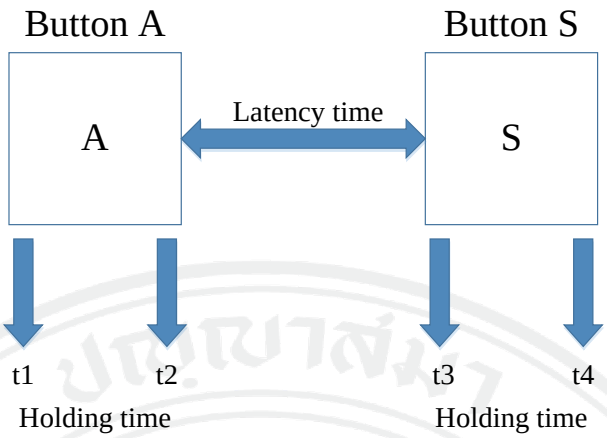


Figure 2.4 The measurements of the holding time and the latency time

From Figure 2.4, we can calculate the holding time:

Holding time of A = t2-t1

Holding time of S = t4-t3

We can also calculate the latency time:

Latency time of A switch to S = t3-t2

A keystroke authentication method on a smartphone with a touchscreen device using virtual keyboard is presented (Huang, Lund and Sapeluk, 2012: 1342-1347). The holding time, latency time and password are used to produce the keystroke pattern template by applying the following equations.

$$\text{Pattern Holding time} / \alpha < \text{Attempt Holding time} < \text{Pattern Holding time} \times \alpha$$

and

$$\text{Pattern Latency time} / \alpha < \text{Attempt Latency time} < \text{Latency time} \times \alpha \text{ Attempt}$$

Holding time and Attempted Latency time comes from when the users try to access the system by typing in their password with the same rhythm that was previously registered. Attempted Holding time value and Attempt Latency time must satisfy the above conditions in order to be authenticated. The paper states that $\alpha = 4$ is the best choice.

Using a virtual keyboard for keystroke authentication on small screen devices is difficult since the buttons on the virtual keyboard are very small; this often results in an error caused by a nearby button being pressed. An auto-correction feature is usually favorable when typing on a virtual keyboard. However, for authentication, the users have to enter the correct password within a time limit based on the registered rhythm which is challenging to accomplish. Thus, the users opt to use a simple password and a simple rhythm which is vulnerable to attacks. Although using a tablet or a larger screen device which provides a larger virtual keyboard reduces typing errors, shoulder surfing attacks become more problematic.

Furthermore, a long password may be inconvenient since the users must enter it many times a day due to screen locking feature. The users cannot use the other hand to cover the screen when typing on a virtual keyboard because the user must look at keyboard.

2.3 Touchscreen Authentication

Antal et al. (Antal, Szabó and László, 2015: 820-826) proposed another keystroke authentication on smartphones with the Android operating system. Three methods which consist of holding time, latency time and pressure of fingers on the smartphone screen are used. A new virtual keyboard was developed in their research and a piece of software called Weka was used as a classification tool. Euclidian distance was used in their experiment. However, this research has a similar weakness for using a virtual keyboard as previously mentioned. Moreover, it is difficult to detect the correct pressure when the users attempt to authenticate while doing various activities such as walking, standing, and sitting.

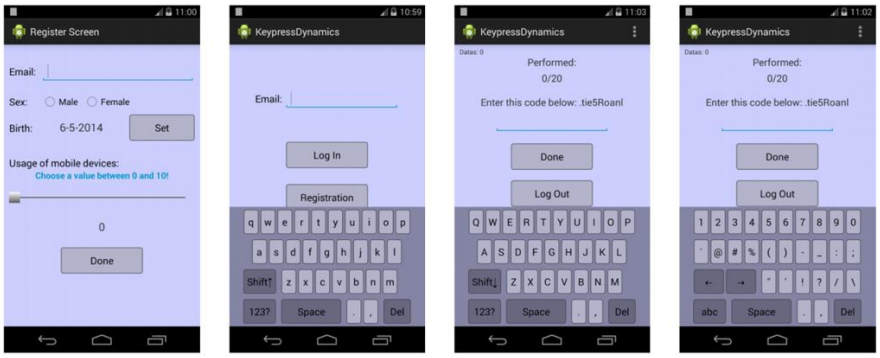


Figure 2.5 The application screen of Antal et al. research

Saevanee and Bhatarakosol (Saevanee and Bhatarakosol, 2008: 82-86) proposed a keystroke authentication using a touchpad of a laptop and the features includes holding time, latency time, and pressure. The proposed method divides the touchpad into grids representing a numeric keypad. The experiments were conducted by gathering data while participants entered their 10-digit phone number 30 times continuously. Features were extracted using k-nearest neighbors (k-NN) algorithm and the Euclidian distance. The authors claimed that using only the finger pressure with the k-NN analytical method can identify the users with an accuracy rate of 99%. Even though the result seems impressive, measuring pressure is impractical as discussed earlier .

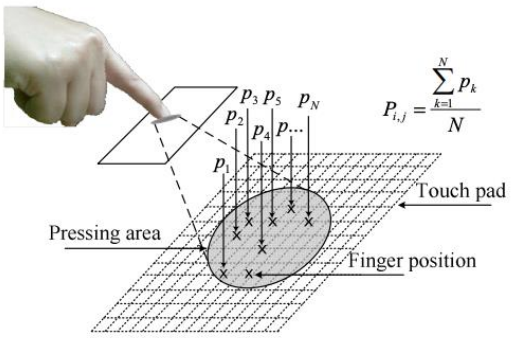


Figure 2.6 Calculation method of the finger pressure value of Saevanee and Bhatarakosol research

Azenkot et al. (Azenkot, Rector, Ladner and Wobbrock, 2012: 159-166) proposed a multi-touch authentication method using the touchscreen on an iPhone and the features includes the number of finger (limit to four fingers) and which fingers were

used. They call their method is PassChords. First, the login process starts when the users place four fingers on the screen, the system will detect which fingers were used by location on the screen (fore, middle, ring and little finger) and respond to the users with a vibration if all four fingers were placed on the screen. The users must then tab 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, if the sequence of tab and number of fingers per tab are matching in the database, the user is authenticated. PassChords was developed for blind people, who lock/unlock their smartphone by VoiceOverPINs, which they deemed no longer secure. The experiment shows that PassChords authentication are nearly 75% faster than VoiceOverPINs and more secure, because PassChords produce no audio feedback and doesn't show anything on the screen, therefore making aural and visual eavesdropping more difficult. Moreover, it is easy to detect the correct number of fingers per tab therefore it is similar to a traditional password. Attacker only remembers how many fingers were tapped each time just like a digit password. The limitation with PassChords authentication however, is the high chance of replicated patterns even if the system wasn't limited to 4 fingers.

Figure 2.7 PassChords detection and authentication

IR Ring (Roth, Schmidt and Güldenring, 2010: 259-262) is an amazing device that prevents touch or input on a multi-touch display from an unauthorized user. If the user needs to touch their fingers on a multi-touch display device, the user must wear the ring (IR Ring) first. The IR Ring was developed from a small circuit board with an infrared module. The IR Ring is identifying the user who wears it and returns the location of the hand on the screen. Only the person wearing the IR Ring can input any action onto the screen. Although, this method is pretty good, it is no different from a USB token or an NFC tag (what you have?) therefore the ring is at risk of being lost or

stolen by criminals; furthermore, it is costly to replace and constantly needs to be recharged, making it inconvenient.

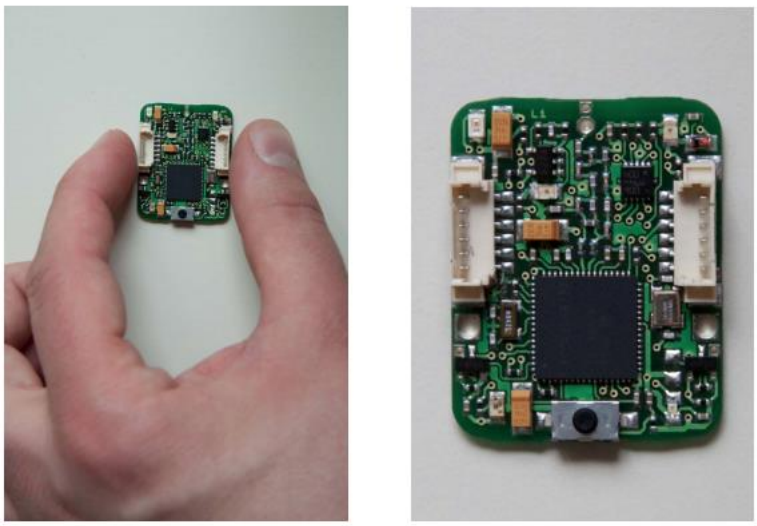


Figure 2.8 IR ring circuit board

If there is no visual feedback from the screen (under the table) the tap authentication on smartphones is the fastest and more secure than other methods (Marques, Guerreiro, Duarte and Carriço, 2013: 33-39). The researchers uses pattern, PIN and tap authentication method in their experiment. The experiment was split into two parts. First, the users can look at the screen while he /she tries to authenticate with three previously mentioned methods (draw pattern, PIN and tab) 30 times per each method. Second, the user repeats the process but this time they cannot look at the screen, while he/she tries to authenticate. For both parts of the experiment, the fake attackers will try shoulder-surfing attacks while the users tries to authenticate. They experiment was shown in Table 2.1.

Table 2.1 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
Tab	3.73s	6.18s	0.42 times

As shown in If there is no visual feedback from the screen (under the table) the tap authentication on smartphones is the fastest and more secure than other methods (Marques, Guerreiro, Duarte and Carriço, 2013: 33-39). The researchers uses pattern, PIN and tap authentication method in their experiment. The experiment was split into two parts. First, the users can look at the screen while he /she tries to authenticate with three previously mentioned methods (draw pattern, PIN and tab) 30 times per each method. Second, the user repeats the process but this time they cannot look at the screen, while he/she tries to authenticate. For both parts of the experiment, the fake attackers will try shoulder-surfing attacks while the users tries to authenticate. They experiment was shown in Table 2.1.

Table 2.1 the tap authentication is fastest if the user is not looking at the screen (no visual). The experiment shows that if the user tries to authenticate his smartphone in the public 30 times (visual), the shoulder-surfing attacks when authenticating with a draw pattern or tap unlock were successful 5 times and 9 times for PIN authentication. Although authenticating on a smartphone without visual feedback from the screen does indeed increase security by preventing shoulder-surfing attacks, it is evidently more prone to errors. On that note, Table 2.1 further show that the tap authentication method has the lowest error rate (0.42 times) when compared with the other methods. The tap authentication of their research is using a simple keystroke pattern, just as Azenkot (Azenkot, Rector, Ladner and Wobbrock, 2012: 159-166) researched, excluding the multi-touch feature. When the user presses/releases their fingers on the screen (they call on/off) it will convert each press to 1 and release to 0 and sum up all the holding time. They represented their pattern as follows:

(64-bit array (0 or 1), total time, pattern length)

They use Hamming distance for classification. However, although what they proposed is the best performing method of the three, being the fastest, easiest and arguably the most secure, it is still prone real life risks. In real life, it is unlikely that anyone authenticates for the usage of their smartphone under the table every time. This means that there is still a 16.7% chance of a malicious shoulder-surfing being successful for tap authentication in real life scenario.

In 2013, De Luca et al. (De Luca, Von Zezschwitz, Nguyen, Maurer, Rubegni, Scipioni and Langheinrich, 2013: 2389-2398) proposed a back-of-device authentication device called BoD. The user can choose one of two modes: “BoD Pattern Unlock” (draw the pattern for unlock) or “BoD Shapes” (draw the shape for unlock) for authentication service. When the user needs to unlock the phone, the user must draw the pattern or draw the shape at the back of the phone for authentication. The back of the phone was embedded with a portable touchscreen device which was connected to the smartphone wirelessly. This research was successfully in preventing shoulder surfing attacks but it has a high false acceptance rate (FAR). This is mainly because, drawing on the back of a smartphone is very confusing and needs a high level of concentration. Because De Luca’s (De Luca, Von Zezschwitz, Nguyen, Maurer, Rubegni, Scipioni and Langheinrich, 2013: 2389-2398) method has a high FAR, Leiva et al. (Leiva and Català, 2014: 63-66) proposed a set of improvements to De Luca’s (De Luca, Von Zezschwitz, Nguyen, Maurer, Rubegni, Scipioni and Langheinrich, 2013: 2389-2398) researched problem. Leiva (Leiva and Català, 2014: 63-66) added a new feature called “BoD Taps”. The authentication process is that the users tap the back of the phone using their fingers to unlock. As expected, It was simple and successful in reducing FAR and preventing shoulder surfing attacks. However, such a peripheral device again, has its flaws. For starters, a portable touchscreen is costly and impractical to use in real life. Nowadays, people want a smartphone which is thin and light where all the needed functions should already be embedded within the device; such a peripheral is simply too much of a compromise in terms of usability.

Figure 2.9 BoD prototype device

Takada and Kokubun (Takada and Kokubun, 2013: 307-310) proposed a new method of mobile authentication. This method combines PIN with the multi-touch

feature of a smartphone. They are using PIN (4 digit) for authentication by virtual numeric keyboard (0-9) like a traditional PIN, but they allow the user to use more than one finger when inputting the PIN. and their experiment show the time of single touch is lower than multi-touch. Takada and Kokubun (Takada and Kokubun, 2013: 307-310) researched is simple and easy to use, but not increase security anymore. Because of users have to seeing on the screen when they need to unlock their phone and they must carefully input for correct PIN and correct pattern, it slowly. From above reason Takada and Kokubun (Takada and Kokubun, 2013: 307-310) method is easy to shoulder suffering attack like Azenkot (Azenkot, Rector, Ladner and Wobbrock, 2012: 159-166) proposed.

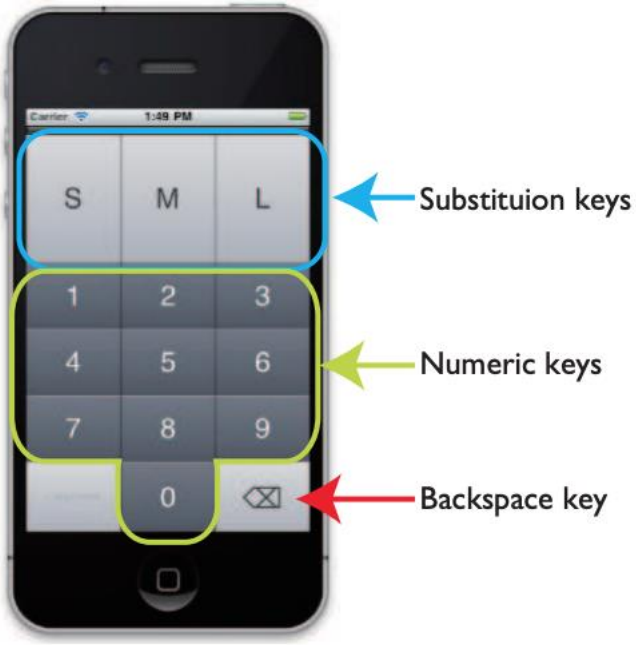


Figure 2.10 User interface of Takada and Kokubun research

2.4 K-Nearest Neighbors

K-Nearest Neighbors (K-NN) is a widely used pattern recognition method because it is simple to use and easy to understand. The principle of K-NN is to compare the similarity between the interest data with the set of stored data to find what the class of the interest data should be, if the interested data's distance is closest with any datasets of stored data, the class of the nearest data is the class of interest data. To find the

distance matrix K-NN, a Euclidean Distance algorithm is used with the following equation:

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

Where p is the interest point of attribute, q is the point of data set in the same attribute and n is number of attributes. After calculating the distance metric, we can answer what the class of interest data is by the minimum amount of distance.

2.5 Two-Factor Authentication

Two-factor authentication is two-step authentication which applies to systems that require additional security. Normally, access to various systems will generally require a one step identity authentication process ie, by entering the user ID and password enable to access the system. Two-factor authentication, in addition to using the user's ID and password as the first authentication process, must use a One Time Password (OTP) as the second authentication process (ALOUL, ZAHIDI and EL-HAJJ, 2009: 641-644). An OTP is usually delivered through the following method:

- Short Message Service (SMS)
- Programs or applications (Google Authenticator App)
- Token device

Using Two-Factor Authentication allows access to various systems to be more secure, because even if the attacker knows the username and password, they will still be unable to access the system without an OTP. Two-factor authentication is often used with these systems for identity verification: Internet banking system, social media (Facebook, Twitter) and email (Gmail, Outlook).

Figure 2.11 below shows the simple process of Two-factor authentication on website or application using SMS on a mobile phone.

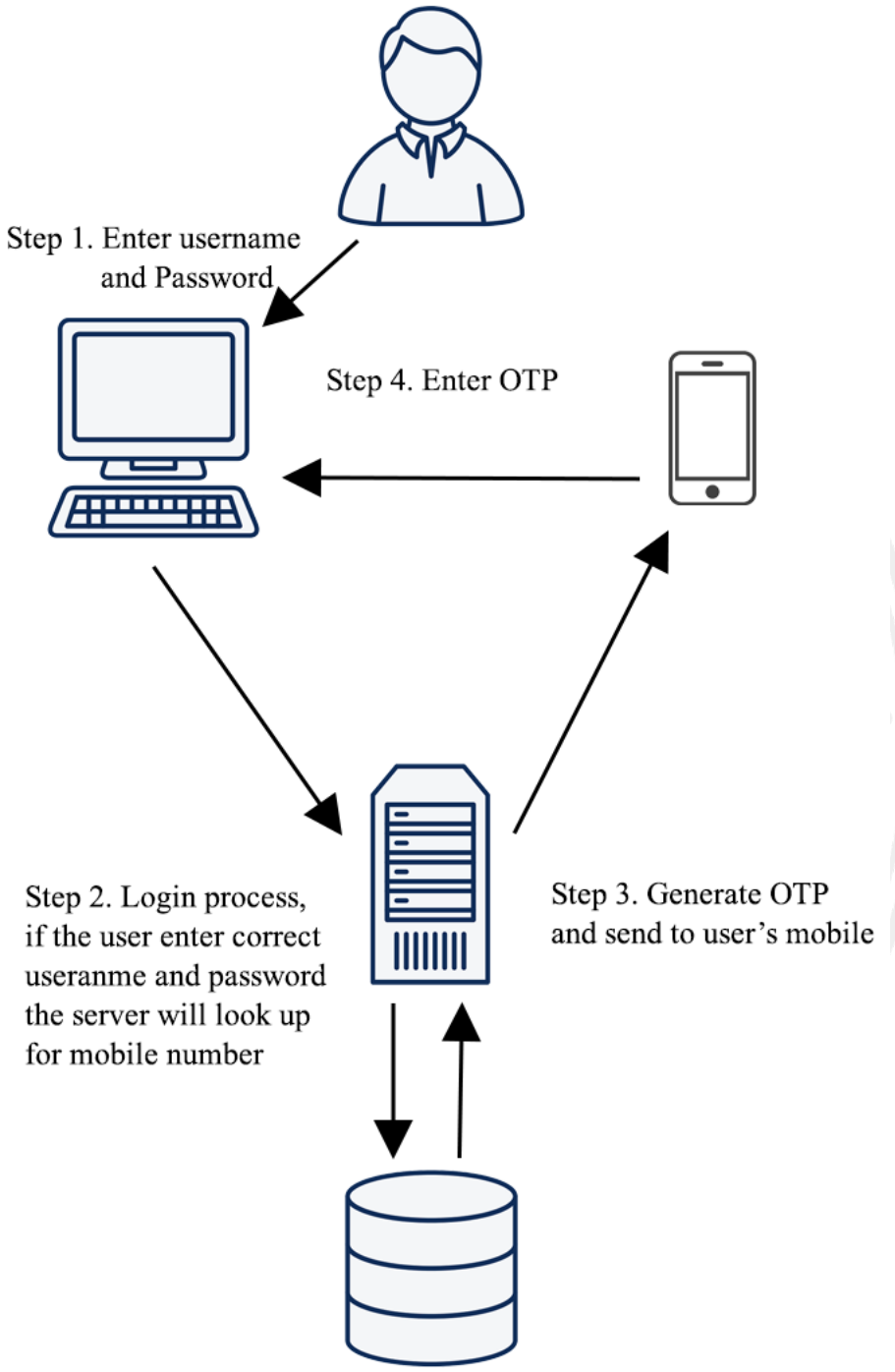


Figure 2.11 The simple process of Two-factor authentication on website or application base on SMS of mobile phone.

CHAPTER 3 METHODOLOGY FRAMEWORK

From related work, we found that a keystroke dynamic authentication can be applied to a touchpad and a touch-screen devices in a similar fashion to keystroke dynamic authentication on a traditional hardware keyboard. Although previous work used a pressure feature, measuring pressure for authentication is not suitable. For this reason, we propose a new method of authentication using keystroke dynamic method which provides an accurate verification, flexibility, and usability. Our proposed method is called a *Rhythm Authentication Using Multi-Touch Technology* in the short, "*Rhythmprint". Rhythmprint* utilizes multi-touch features including four attributes namely, holding time, latency time, distance between fingers and number of fingers per beat. Users can verify themselves by simply touching on a touchpad on a laptop computer or a touch-screen of a smartphone with their fingers with the enrolled rhythm. Each beat can be made by one or more fingers. It can prevent shoulder surfing attacks because users do not need to look at their keyboard when tapping or touching a touchpad or a touch screen. Therefore, they can use the other hand to cover the tapping hand. The attackers will have difficulties performing an eavesdropping attack since a touchpad can obtain event data when the users' fingers make contact with it without making any audible sound. Even when an audible tapping sounds is heard, an eavesdropper has no way of knowing how many fingers and what fingers were used to tap on the touchpad to make one beat. To prove our proposed authentication method, we designed the experiment including two methods which are the template creation method and authentication method.

3.1 Template Creation

Users must register before using the authentication system. To register, users has to make a rhythm by tapping on the touchpad or a touch-screen. We designed a user template to include these four attributes:

- Distance between fingers

- Holding time
- Latency time
- Number of fingers

The distance between fingers is applied from hand geometry (Zhang and Kanhangad, 2011: 529-531). According to this research, each person has a different hand geometry. Therefore, the distance between the fingertips of each user is different when his/her fingers touches an input device. The next attribute is the holding time. It is the time measured when the user pushes his/her fingers on the input device until the user releases his/her fingers from the input device. The third attribute is the latency time. It is the time measured 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 the rhythm.

The registration user interface records the holding time, distance between finger, and number of figures per beat. For the subsequent taps, latency times are also recorded. After that, the keystroke extraction system creates a template of the user and stores it in the biometric database.

The process of the user registration for template creation is shown in Figure 3.1 below.

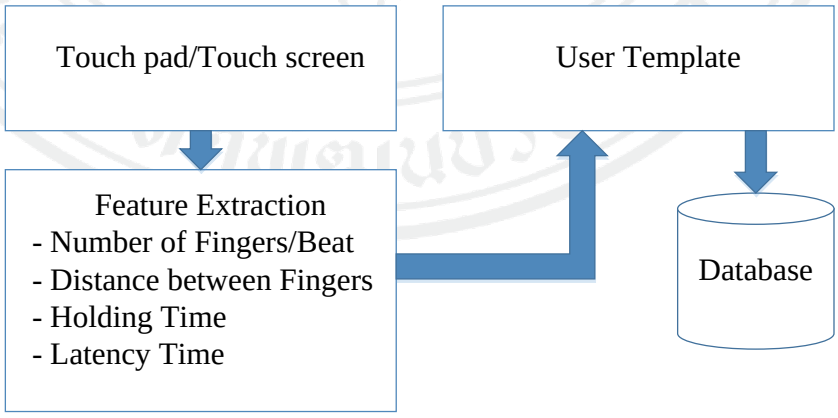
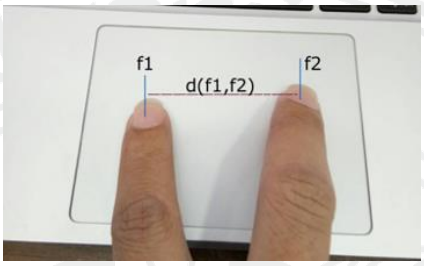


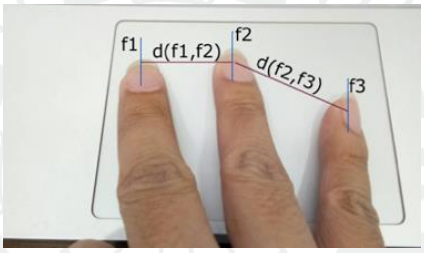
Figure 3.1 The template creation of Rhythmprint

When the user enters each beat to create the rhythm, we collect the holding time, latency time, number of fingers and distance between each fingertip. The number of fingers and distance between fingertips per beat are very important because the holding time and latency time can determine the true owner of the rhythm even if the same rhythm is chosen by more than 1 users. 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 from malicious eavesdropping attack. Figure 3.2 shows the process of the template creation for the first beat of the rhythm and how we collected the data for each beat.

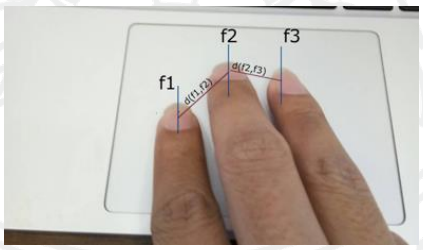
(a)



(b)



(c)



(d)

Figure 3.2 The template creation for the first beat

Figures 3.2 (a) – 3.2 (d) shows when the user enters his/her fingers on the touchpad to create the first beat by using a different number of fingers and different fingers. In figure 3.2 (a), the user presses only one finger on the touchpad to create one beat. In figure 3.2 (b), the user presses two fingers on the touchpad to create the first beat. In figure 3.2 (c), the user presses three fingers on the touchpad to create the first beat. In figure 3.2 (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 3.2 (a)-(d) are created as soon as the user touches the touch pad.

Table 3.1 The sample of first beat creation

	Value (Figure)			
Method	3.2 (a)	3.2 (b)	3.2 (c)	3.2 (d)
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 the fingers used were different. In figure 3 (c) and (d), the number of fingers used were the same, but fingers are different, therefore, the distances between finger tips are different. After collecting the data, we can create the user template and save it to the database.

3.2 Authentication Process

The user can access his/her device such as a laptop, smartphone and other touchable devices by tapping his/her fingers on a touchable input of the target device; naturally, the users will have to use the same rhythm, the same sequence and number fingers per beat using the same hand. This is also known as a login template. The login template will compared with the templates stored in the database by using the K-NN algorithm.

Figure 3.3 below shows the flow of the authentication process.

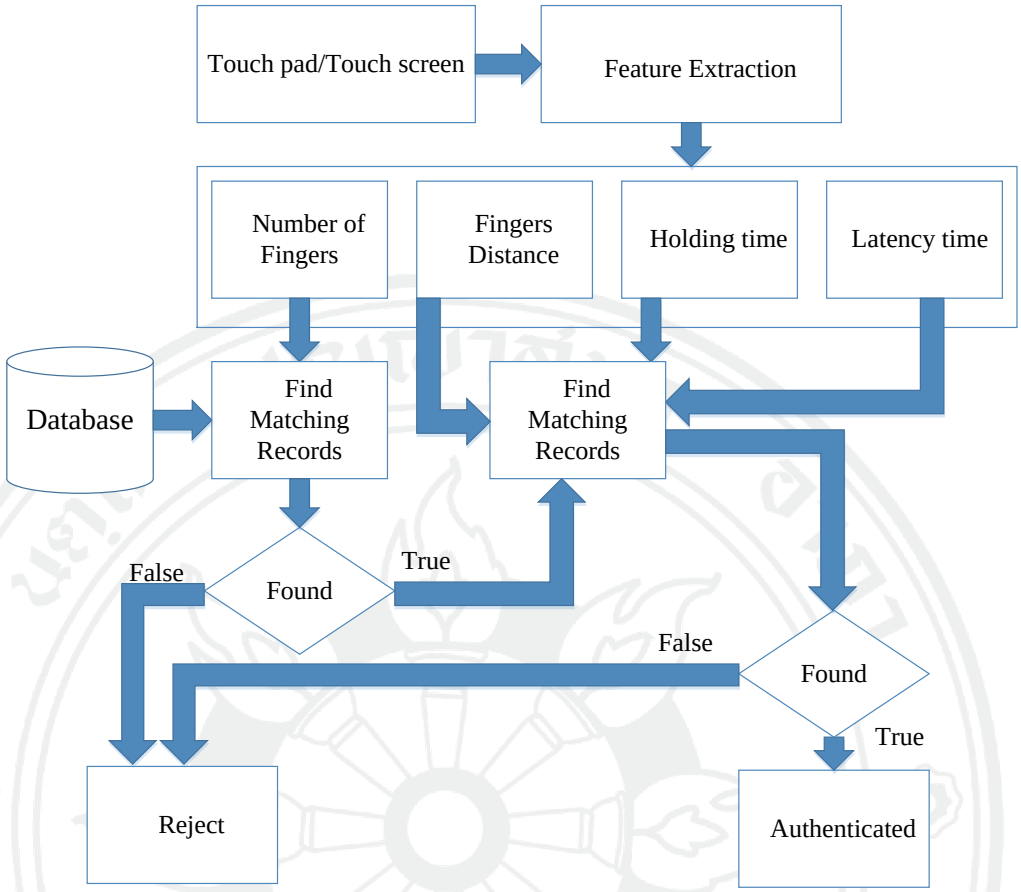


Figure 3.3 The authentication process

The user must enter the rhythm to register the template in the creation process first. This enrolled template will be used to authenticate the user. The authentication process is quite simple, it begins when the user taps or touches his/her fingers on the touchpad with the correct rhythm to login. After the system receives the rhythm from user, the authentication process will start.

First, we calculate the number of beats and the number of fingers used for each beat. Table 3.2 below shows an example of the raw recorded data of the user rhythm used to enters the system by touching on his/her touchpad to authenticate.

Table 3.2 The dataset of user which enter the system for authentication

Number of Fingers/Beat	Number of Beat	Holding Time (s)	Latency Time (s)	Distance between Finger
1, 2, 3, 3	4	0.09400000 000005093, 0.08800000 000064756, 0.07999999 999992724, 0.07999999 999992724	0.00000000 00000000, 0.43199999 99997890, 0.23599999 99998763, 0.16799999 99996653	0, {{0}, {1.34}}, {{0}, {1.30}, {1.23}}, {{0}, {1.30}, {2.11}}

Table 3.3 shows the sample stored data pattern in the database. In terms of the number of fingers per beat, we can directly compare the data with those stored on the database for simple filtering in the first phase.

Table 3.3 Example dataset of stored data in database

ID	Number of Fingers/Beat	Number of Beat	Holding Time (s)	Latency Time (s)	Distance between Finger
1	1, 2, 3, 3	4	0.09400000 000005093, 0.10400000 000026921, 0.07999999 999992724, 0.08800000 000064756	0.00000000 000000000 0.40799999 999944700, 0.22400000 000016007, 0.15999999 999985448	0, {{0}, {1.23}}, {{0}, {1.19}, {1.16}}, {{0}, {1.46}, {2.63}}
2	1, 2, 3, 3	4	0.17799999 999988358, 0.16799999 999966530, 0.15200000 000004366, 0.15200000 000004366	0.00000000 000000000 0.51200000 000062570, 0.50799999 999981080, 0.53600000 000005820	0, {{0}, {1.68}}, {{0}, {1.10}, {1.02}}, {{0}, {1.66}, {2.09}}
3	1, 2, 3, 4	4	0.09400000 000005093, 0.09600000 000045839, 0.11200000 000008004,	0.00000000 000000000 0.36799999 999948340, 0.15999999 999985448,	0, {{0}, {1.05}}, {{0}, {1.80}, {1.98}}, {{0}, {2.60}, {1.99}, {2.48}}

			0.09600000 000045839	0.17599999 999947613	

After filtering by using the number of finger per beat, we will get the minimum records that was shown in Table 3.4

Table 3.4 The set of stored dataset in database after filtering by the number of finger/beat

ID	Number of Fingers/Beat	Number of Beat	Holding Time (s)	Latency Time (s)	Distance between Finger
1	1, 2, 3, 3	4	0.09400000 000005093, 0.10400000 000026921, 0.07999999 999992724, 0.08800000 000064756	0.00000000 000000000 0.40799999 999944700, 0.22400000 000016007, 0.15999999 999985448	0, {{0}, {1.23}}, {{0}, {1.19}, {1.16}}, {{0}, {1.46}, {2.63}}
2	1, 2, 3, 3	4	0.17799999 999988358, 0.16799999 999966530, 0.15200000 000004366, 0.15200000 000004366	0.00000000 000000000 0.51200000 000062570, 0.50799999 999981080, 0.53600000 000005820	0, {{0}, {1.68}}, {{0}, {1.10}, {1.02}}, {{0}, {1.66}, {2.09}}

To understand, we can explain with the line graph. If the sequence of fingers' number/beat is

1 2 1 2 3 1 2 1 2 1

it means that the users uses only one finger to create first beat, two fingers to create the second beat and so on and so forth. Figure 3.4 shows a line graph of the number of fingers and the sequence of taps that correctly matches the enrolled template. Two graphs must be the same, otherwise the user will not be authenticated. Figure 3.5 shows an example when an attacker knows the user's rhythm but not the number of fingers per beat and try to login as the valid user. The system rejects the login since, the sequence of the numbers of fingers does not match the enrolled template.

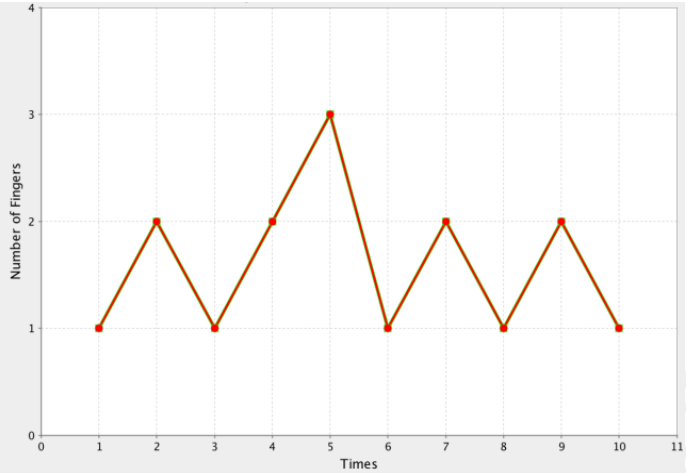


Figure 3.4 The sequence of fingers/beat in the rhythm matches the enrolled template

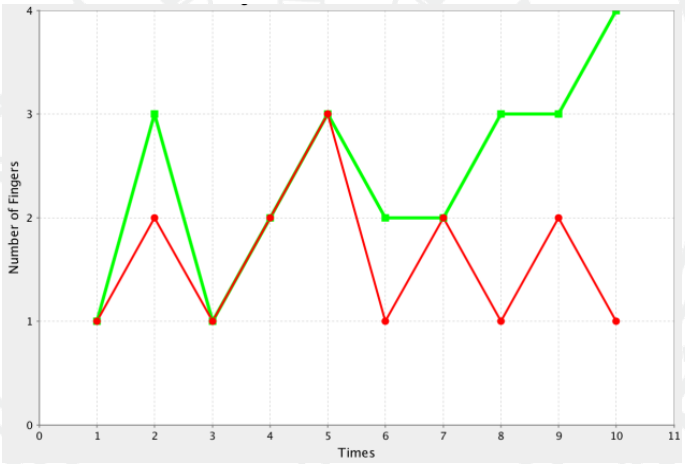


Figure 3.5 The sequence of fingers/beat in the rhythm does not match the enrolled template

After filtering by using the number of fingers per beat we got the minimum records that was shown in Table 3.4, the next phase is the classification process.

3.3 Classification

The classification process was designed into two methods namely:

- Real-time classification using our java programming

- Weka program classification

3.3.1 Real-time classification using our java programming

By using our java program, we can classify real-time (model creation and classification), when the user enters the Rhythmprint for authentication; The authentication process to find a matching record in Figure 3.3 can be separated into two steps. First, when the feature extraction is completed, we will obtain four attributes from the user's input including the number of fingers, distance between fingertips, holding time and latency time per beat. The number of fingers per beat can be used to filter the records in database directly. After filtering, we will obtain the minimal records. Second, the system will classify the login template by using the holding times, latency times and distance between fingertips against the template in the minimal records from the previous step using the K-NN algorithm. The principle of the K-NN algorithm is to compare the similarities between interest data with the set of stored data to find what the class of the interest data should be, if the interested data distance is the closest with a set of stored data, the class of the nearest data is the class of interest data. To find the distance matrix, the K-NN algorithm is using Euclidean Distance algorithm with following equation

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

If we can find the minimum distance calculated from the Euclidean Distance algorithm, the user is authenticated, and we can calculate the Euclidean Distance by using following equation:

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

After we got the distance by Euclidean Distance, we can find the minimum distance.

3.3.2 Weka program classification

We plan to use the Weka program to calculate an accuracy rate of the Rhythmprint in order to test whether the algorithm that we have designed can identify the users by using our four measurement including, the holding time, latency time, number of fingers per beat and distance between fingers. By using the Weka program for classification in the experiment section, we have to prepare data from the database into two datasets namely:

- Data for model creation (learning data)
- Data for classification (test data)

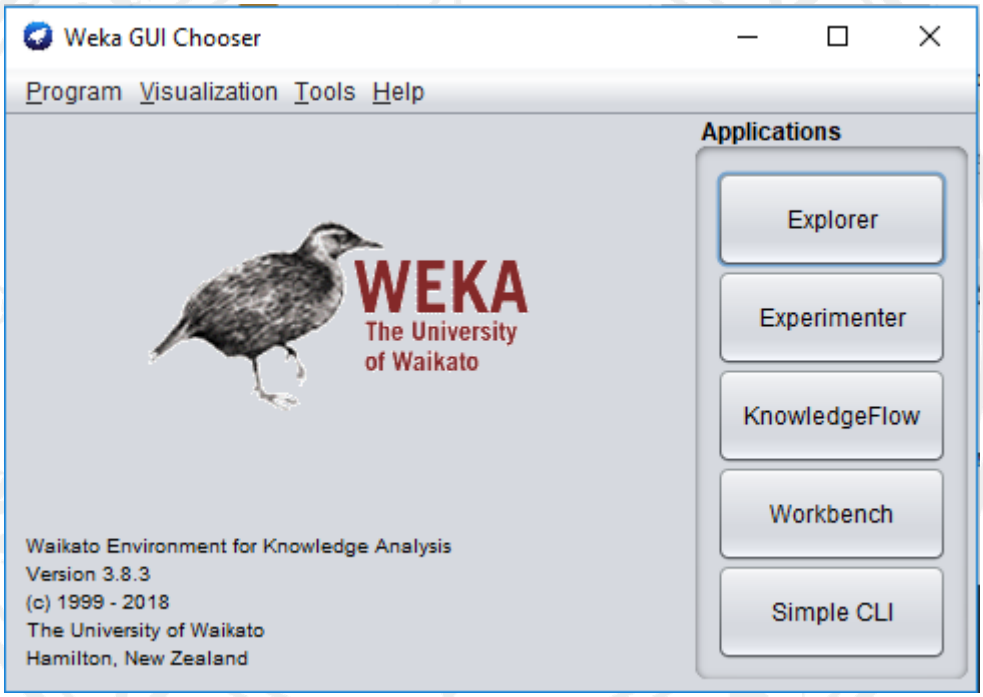


Figure 3.6 The Weka program GUI

The model which we used for model creation consist of six attributes namely:

- Sequence number of fingers per beat
- Number of fingers
- Latency time
- Holding time
- Distance between finger per beat
- Username

Table 3.5 The example of attribute for Weka model creation

Number of fingers/beat	Number of finger	Holding time	Latency time	Distance between finger/beat	Username
{1, 2, 3, 3}	4	{0.0940000000000509 3, 0.1040000000002692 1, 0.0799999999999272 4, 0.0880000000006475 6}	{0.0940000000000509 3, 0.1040000000002692 1, 0.0799999999999272 4, 0.0880000000006475 6}	{0, {{0}, {1.23}}, {{0}, {1.19}, {1.16}}, {{0}, {1.46}, {2.63}}}	Nakinthon

Table 3.5 shows the examples of dataset for model creation (training data), fetched from our database. Table 3.6 show the unseen datasets which we are using to test our model.

Table 3.6 The example of unseen data which using for test our model

Number of fingers/beat	Number of finger	Holding time	Latency time	Distance between finger/beat	Username
{1, 2, 3, 3}	4	{0.0940000000000509 3, 0.1040000000002692 1, 0.0799999999999272 4, 0.0880000000006475 6}	{0.0940000000000509 3, 0.1040000000002692 1, 0.0799999999999272 4, 0.0880000000006475 6}	{0, {{0}, {1.23}}, {{0}, {1.19}, {1.16}}, {{0}, {1.46}, {2.63}}}	?

3.4 Multi-Factor Authentication

Multi-Factor authentication is an authentication method where computer users are only allowed access after presenting more than 1 type of authentication for verification (what you are?, what you have ?, what you know?). A good example of this is an OTP (which we explained in Chapter 2.) which is a multi-factor authentication method because an OTP uses two factors to verify the users including, with what you know (the user has to know the password) and what you have (the user must have a unique phone number to receive an OTP SMS).

Rhythmprint is multifactor authentication for the following reasons:

- the user must know the rhythm and the number of fingers per beat (what you know)
- the user must have the same hand which were used in the registration process (what you have)
- each person has a different hand geometry. Therefore, the distance between the fingertips of the user is different when his/her fingers touches an input device. (what you are)

For this reason we can conclude that Rhythmprint is a three-Factor authentication method.

CHAPTER 4 RESULTS OF THE EXPERIMENT

The step of our experiment was split into three phases. First phase is the proof of concept; to design and develop a piece of software according to our algorithm on a laptop (real-time classification), and import our database into the Weka program for model creation and classification. FAR and FRR was records in this phase. Second phase is the security challenge; we will try to attack our algorithm using humans and compare how it performs against traditional keystroke authentication in terms of shoulder suffering and eavesdropping attack. The last phase is minimisation; by minimising Rhythmprint's attributes, we try to reduce the complexity of Rhythmprint by eliminating the distance between fingers attribute from the template creation and authentication process (it can reduce the times of template creation making it faster than the full version of Rhythmprint in terms of the authentication process). From there, we and compare the result of the minimised version with full version of Rhythmprint.

4.1 Proof of Concept

4.1.1 Real-time classification

To verify our proposal, we designed and developed software using our algorithm on the laptop using Java programming language for the software development. A MacBook Pro made by Apple Inc. was used in our experiment. We recruited 100 participants to test our program consisting of 34 men and 66 women aged between 16-61 years old. The four measurements used were, the holding times, latency times, number of fingers/beat and, distances between finger tips. These measurements were used to create the user template and the K-NN for classification, while the FAR and FRR were also recorded in this phase. In the experiment, we asked the users to replicate our predetermined fixed rhythm. Our predetermined fix rhythm was created using 10 beats. We tapped this rhythm on the table 10 times for the users to listen too. After that the user was tasked with replicating our rhythm using one or more fingers. Each user must try to replicate our fix rhythm 10 times with one's own sequence and number of

fingers per beat. After all users already created the template, we would have 1,000 records of 100 user templates in our database. Figure 4.1 shows the workflow of template creation.

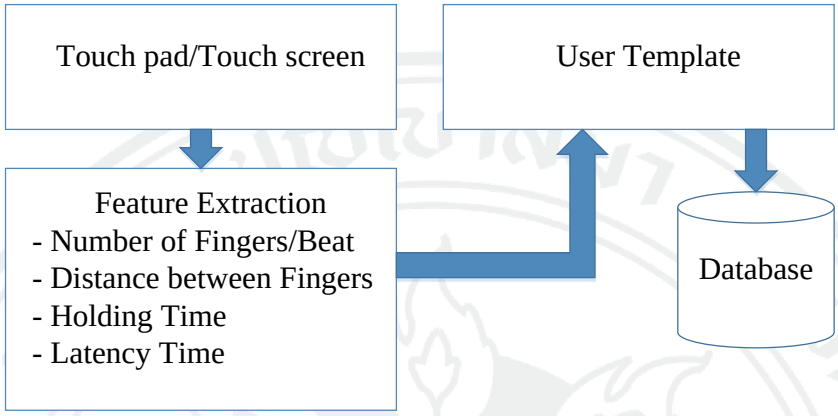


Figure 4.1 The work flow of template creation

After obtaining 1,000 templates of 100 users, we selected 50 of 100 users to authenticate oneself by trying to access our laptop by replicating our predetermined fix rhythm 10 times using their own sequence and numbers of fingers per beat.

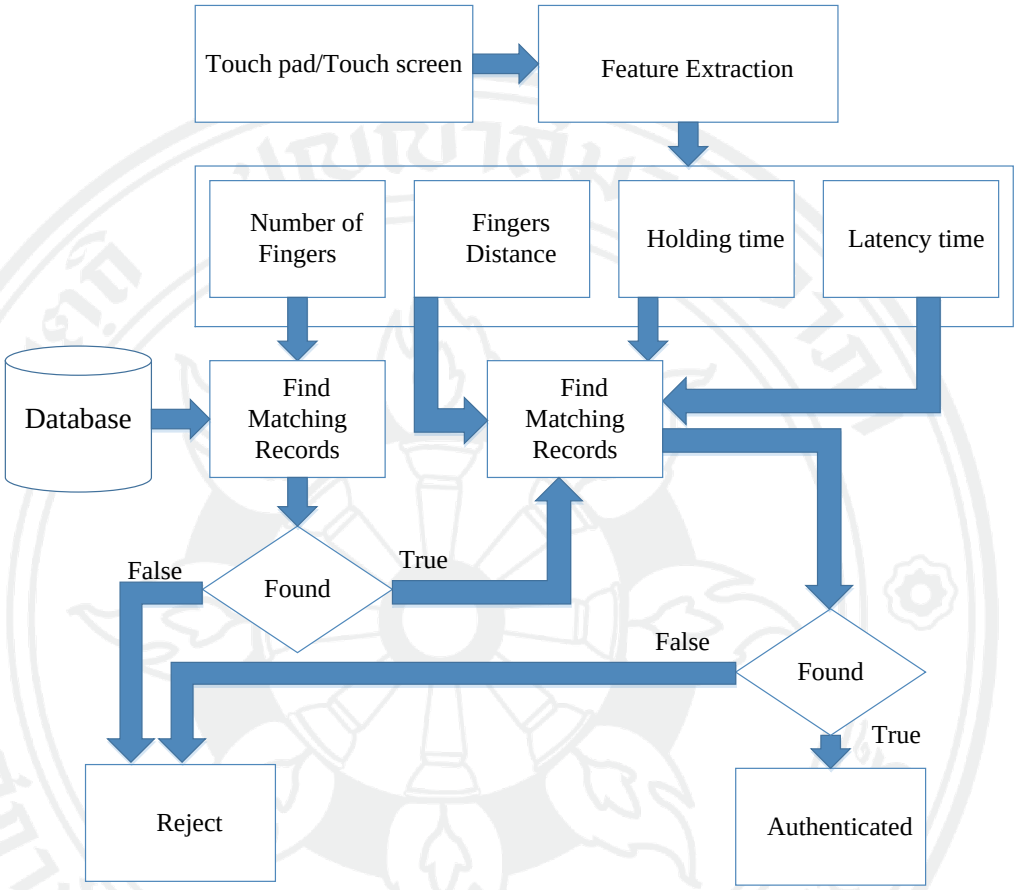


Figure 4.2 The work flow of autentication process

The result of 50 user authentication 500 times is shown in Table 4.1.

Table 4.1 The fix rhythm authentication result

FAR	FRR	Valid
0.4%	3.2%	96.4%

The result of FAR and FRR in our experiment using our java program was showed in the Table 4.1. The percentage of the correctness is 96.4% of valid authentications and only 3.2% of FAR and 0.4% of FRR.

4.1.2 Offline classification

We used data from the database which were collected from 100 participants (1,000 records) using the previous test program. We used four measurements including with the holding times, latency times, number of fingers/beat and distances between fingertips to create the user template and K-NN for classification. FAR and FRR will be recorded in this phase. In the experiment, we selected 5 of 10 of each user's data from database to be used for testing (500 records). We inputted 500 selected records into our program for automatic classification.

Table 4.2 The fix rhythm authentication offline classification results:

FAR	FRR	Valid
0%	0%	100%

The result of FAR and FRR in our Offline classification experiment was shown in the Table 4.2. The percentage of the correctness is 100% of valid authentication and 0% of FAR and 0% of FRR.

4.1.3 Weka classification

We used Weka program to calculate an accuracy rate of the Rhythmprint in order to test whether the algorithm that we have designed can identify each individual user by using our four measurements namely, holding time, latency time, number of fingers per beat and distance between fingers. By using the Weka program for classification in the experiment section, we have to prepare data from the database into two datasets namely:

- Data for model creation (learning data)

- Data for classification (test data)

The model which we used for model creation consist of six attributes namely:

- Sequence number of fingers per beat
- Number of fingers
- Latency time
- Holding time
- Distance between finger per beat
- Username

We select ten records of each username from the database which were collected from the previous experiment; five records for the Weka modal creation and the other five for test set. The data of users in the database were recorded in JSON format, which cannot be imported directly into Weka, for this reason we have to convert data from the database into the correct format of datasets for Weka first.

Example

The sequence number of fingers per beat was collected in the database using the following format

{1, 2, 3, 3}

It means that the first beat was created by one finger, the second beat was created by two fingers, the third beat was created by three finger and final beat was created by three fingers. We know the maximum amount of number of fingers per beat of each user by rule of the experiment is 10, so we transform the JSON data of number of finger per beat {1, 2, 3, 3} into 15 columns (plus 5 for some user who enter the wrong sequence). 0 is entered when there are no beats produced by a touch input. Table 4.3 shows the number of fingers per beat in the csv format.

Table 4.3 The number of fingers per beat in the csv format

Username	Beat_1	Beat_2	Beat_3	Beat_4	Beat_5	Beat_...	Beat_15
Nakinthorn	1	2	3	3	0	0	0

Firstly, enable to begin classification by using the Weka program a model has to be created, this is done by importing 250 records of prepared data (5 of 10 of each user data from the database) into Weka. Figure 4.3 shows how to import data into Weka and the list of attributes was shown in figure 4.4.

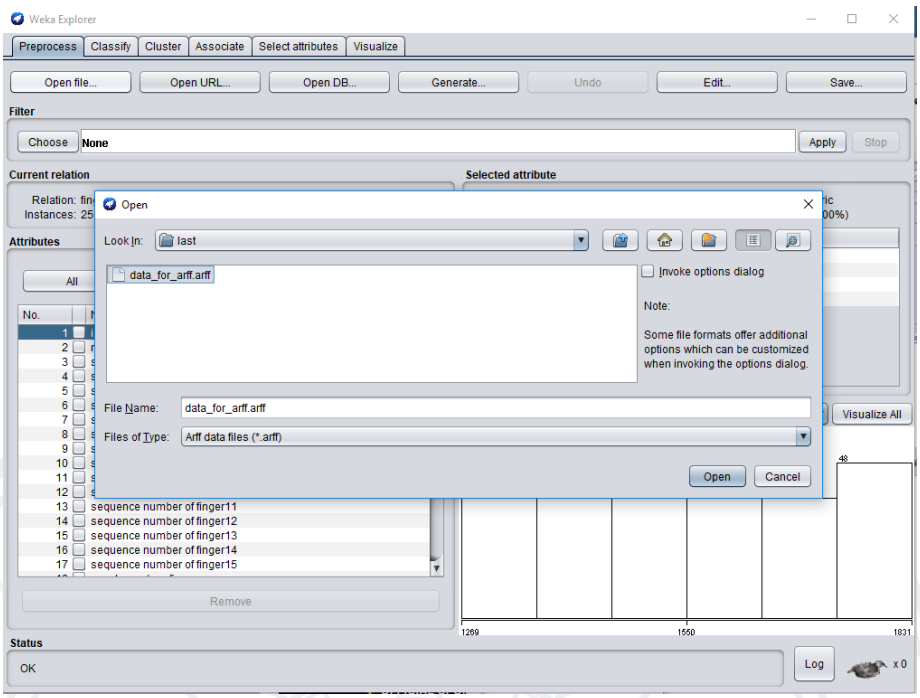


Figure 4.3 The selection of learning data into Weka program

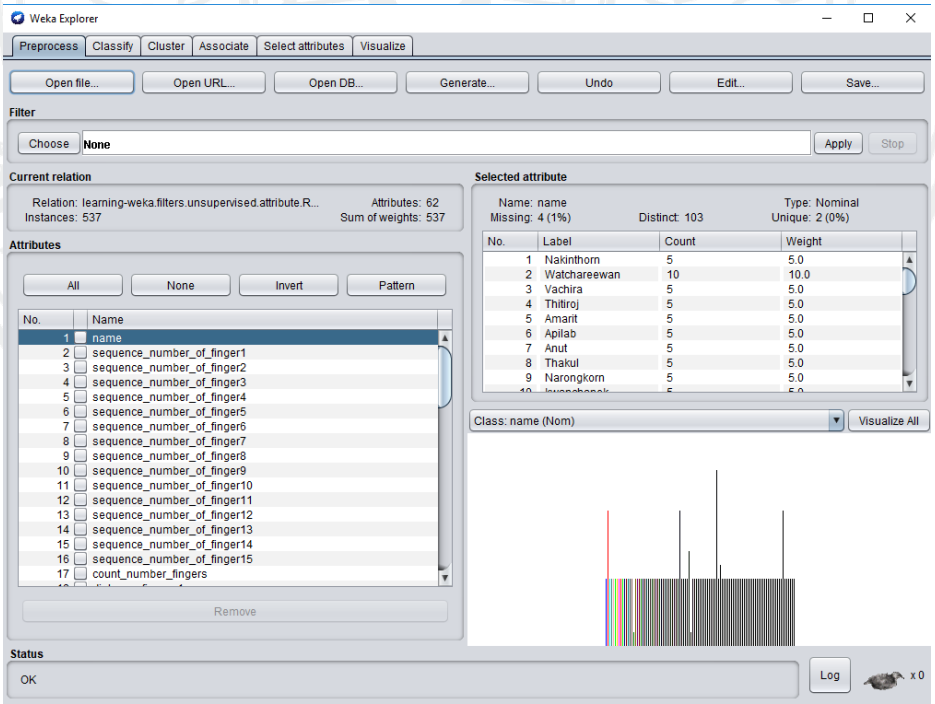


Figure 4.4 The list of attributes for model creation shown in the Weka program

In the Weka program, we have selected the algorithm called lazy-IBK for classification using the K-NN algorithm method. Figure 4.5 shows the classification algorithm selection and figure 4.6 show the result of our model.

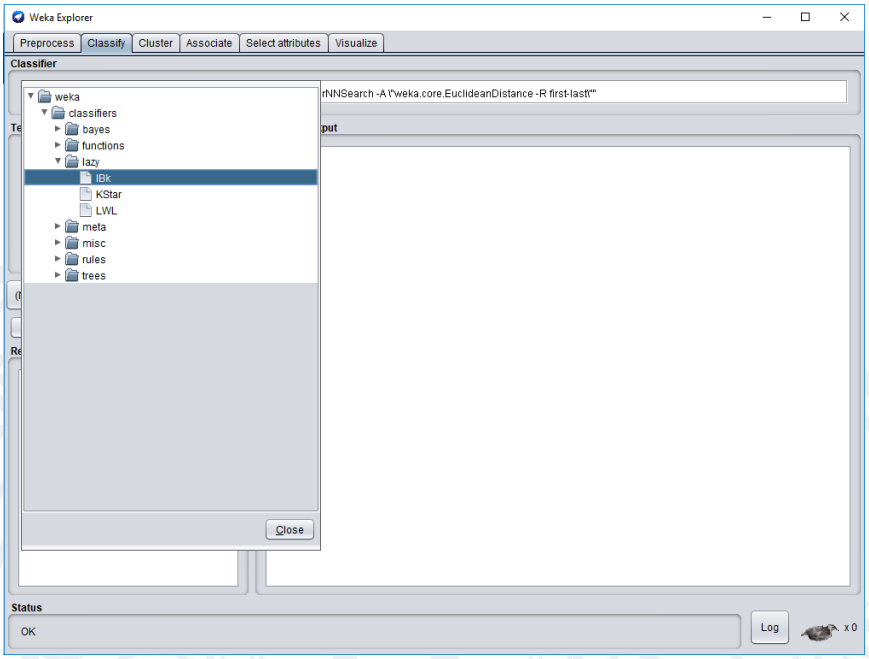


Figure 4.5 The K-NN algorithm selection in the Weka program

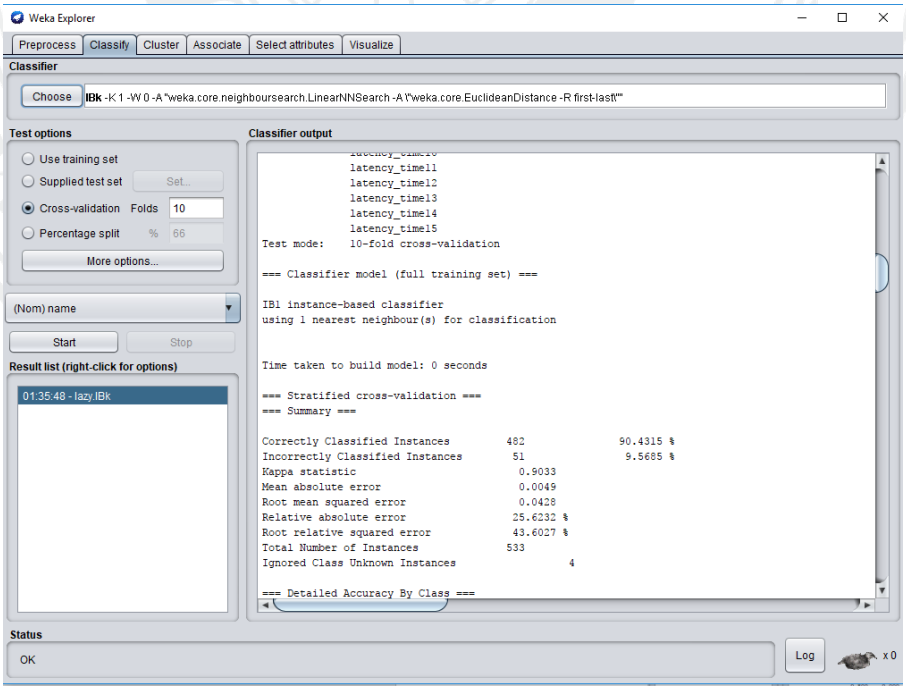


Figure 4.6 The result of model creation in the Weka program

Finally, we import test set into Weka for testing our model. Figure 4.7 shows the test datasets. We entered “?” (the question mask) symbol instead of the username in the test file for testing in the Weka program. Figure 4.8 shows the result of the classification.

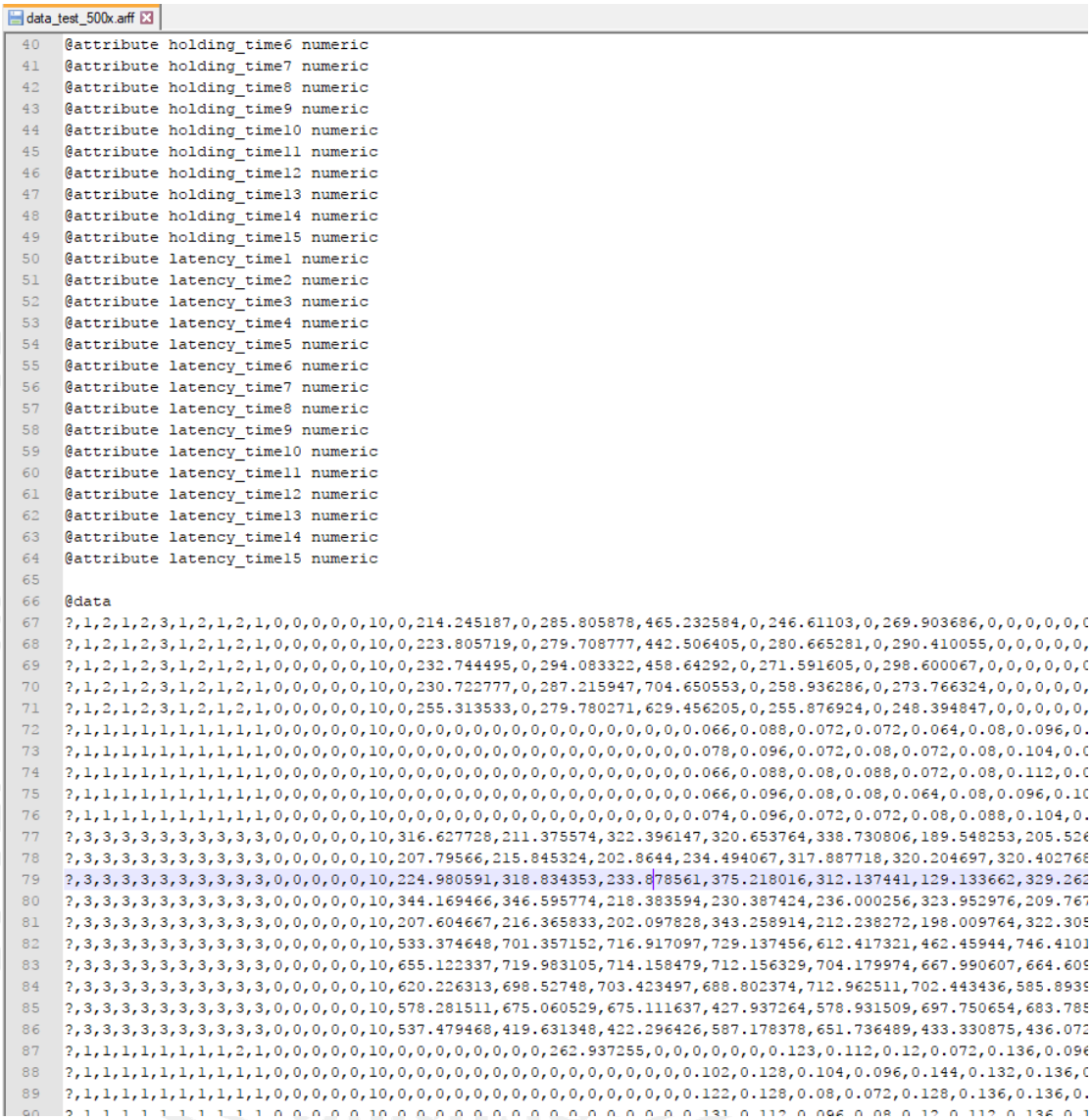


Figure 4.7 The test datasets for testing our model in the Weka program

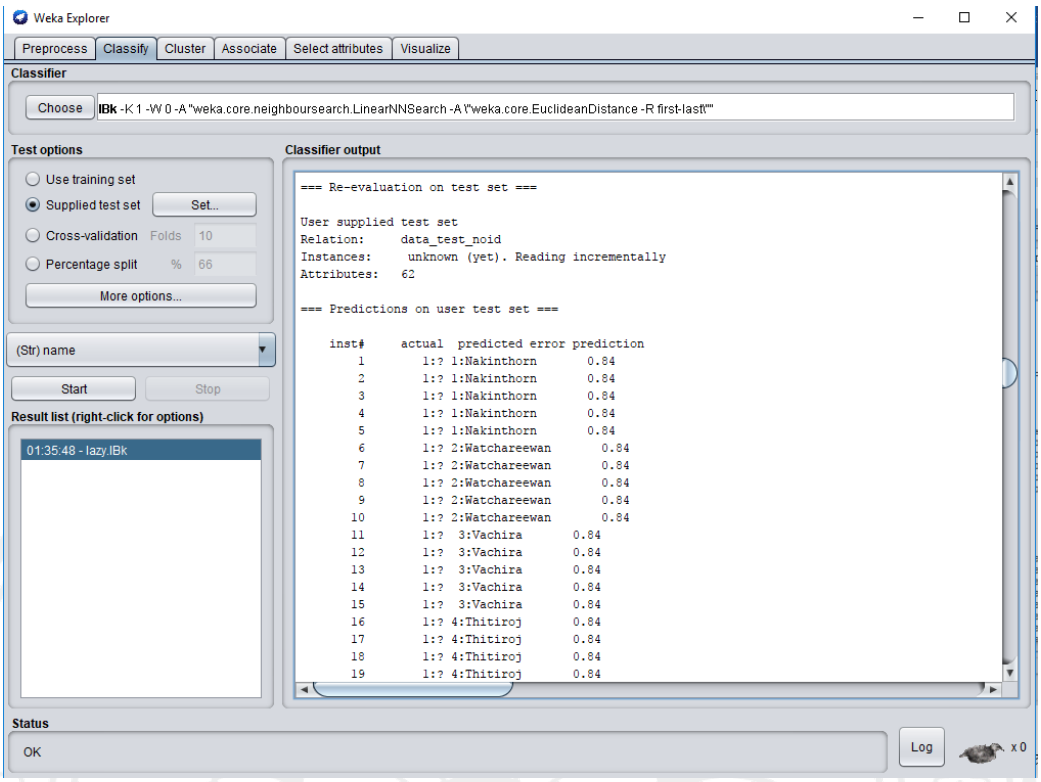


Figure 4.8 The result of classification in the Weka program

The result of the classification for predicting the username using five attributes by the Weka program, shows that from 247 records belonging to 50 user's (some user has only 4 record) test files, we found that in 220 instances the username prediction was correct while 27 was incorrect. This results in a calculated correctness of classification rate of 89.07%

4.2 Security Challenge

The Rhythmprint authentication combines the advantage of the traditional keystroke authentication with the multi-touch technology based on a touchpad on a laptop. With the Rhythmprint authentication, the user is less likely to suffer from shoulder surfing and eavesdropping attacks. This experiment provides empirical evidence to verify the security performances of the Rhythmprint authentication compared with traditional keystroke authentication for shoulder surfing and

eavesdropping attacks. This is evident when the user tries to login to a program on a laptop 10 times in a public place while the attacker stands behind them.

We implemented all the authentication programs for the Rhythmprint authentication and the traditional keystroke methods on a laptop. For the experimental design, we simulated a situation where the user must authenticate themselves to an application on a laptop, sitting in a public place while an attacker stands behind the user. The attacker stands behind the user at all times, while the user is trying to authenticate himself/herself onto an application on a laptop using the Rhythmprint authentication, and the traditional keystroke authentication.

For each method, the user must try to authenticate themselves 10 times, the attacker has to perform a shoulder surfing and an eavesdropping attack every time. The user enters the password on the keyboard using the traditional keystroke authentication method and makes the rhythm on the touchpad for the Rythymprint authentication. Every time the user can authenticate successfully, we tested whether or not the attacker is able to authenticate on the victim's laptop then the results are recorded. For this experiment, we do not allow the user to use the other hand to cover the hand used to authenticate when tapping on the touchpad or typing on the keyboard. Figure 4.9 shows the simulated situation of shoulder surfing attacks.

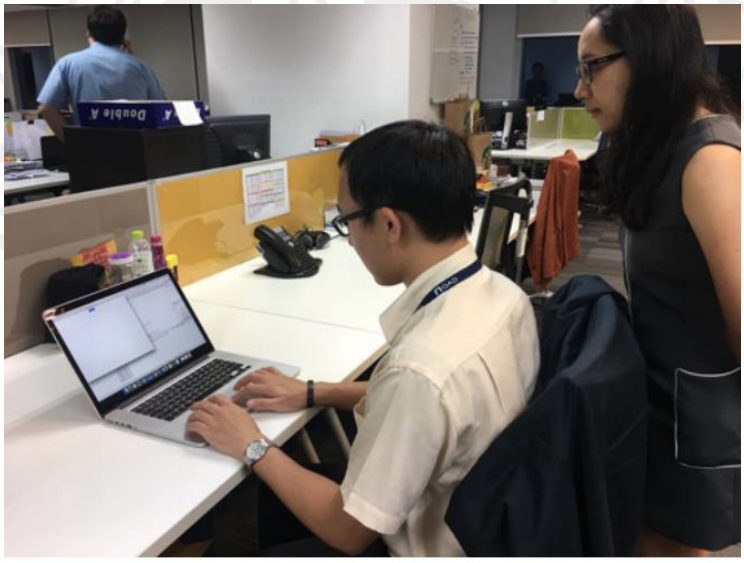


Figure 4.9 The shoulder suffering and eavesdropping attack situation

We have ten volunteers consisting of five women and five men. The age range of the volunteers is between 19-45 years old. The experiment was conducted at a coffee shop, a public place where everybody can see when the user tries to authenticate on the laptop. The attacker will be standing behind the user while simultaneously trying to perform a shoulder surfing and an eavesdropping attack at all the times of the experiment. The user must successful verifying their identity using each authentication method 10 times (Rhythmprint and keystroke) while they are not allowed to use anything to cover themselves while touching on the touchpad or typing on the keyboard. Every time a user successfully authenticates, the attacker will try to replicate it immediately 10 times on the same device. Table 4.4 shows the experiment's results. *The Attack Success Time* from Table 4.4 means the minimum amount time that the attacker used to crack the Rhythmprint or the keystroke of each user. From Table 4.4, it is shown that attacks on Rhythmprint are only successful in only one of ten users, and the attacker had to replicate the process 8 times to successfully attack, while keystroke authentication was always attacked successful (every user was attacked successfully by the attacker). As a result, we can conclude that the Rhythmprint authentication is more secure than keystroke authentication.

Table 4.4 The experimental result of security comparison between Rhythmprint and Keystroke authentication

User	Attack Success Time	
	Rhythmprint	Keystroke
1	Unable to attack	5
2	Unable to attack	9
3	Unable to attack	8
4	Unable to attack	8
5	Unable to attack	10
6	Unable to attack	Unable to attack
7	Unable to attack	5
8	Unable to attack	7

9	8	7
10	Unable to attack	1

From Table 4.4, the minimum attack success time of Rhythmprint were 8 and the number of the user who were attacked successfully was only 1/10. In contrast, the minimum attack success time of keystroke authentication were only 1 and number of the user who were attacked successfully was 9/10.

The result undoubtedly proved that Rhythmprint authentication is more secure than keystroke authentication in terms of both shoulder surfing and eavesdropping attack.

4.3 Minimize Version of Rhythmprint

The minimize version of Rhythmprint authentication uses multi-touch technology for collecting the rhythm when the user touches a touchable device. Three measurements which consist of holding time, latency time and number of fingers per beat, are collected and used to create the user template. What differentiates it from normal Rythymprint authentication is that distance between fingers are no longer collected. When the user needs to log in to a device, the user only needs to touch their fingers on the touchable device with the registered rhythm. K-NN algorithm was used for classification. The attacker must perform shoulder surfing and eavesdropping attacks in order to successfully attack the user; this is because the attacker must know two things: the rhythm and the number of fingers per beat. An eavesdropping attack hardly occurs because touching the finger on a device does not make an audible sound. The algorithm of the minimized version of Rhythmprint authentication can be split into two modules, namely, the registration module and the authentication module. Figure 4.10 shows the registrations flowchart of the minimized version of Rhythmprint authentication and Figure 4.11 shows how the authentication process of the minimized version of Rhythmprint authentication works.

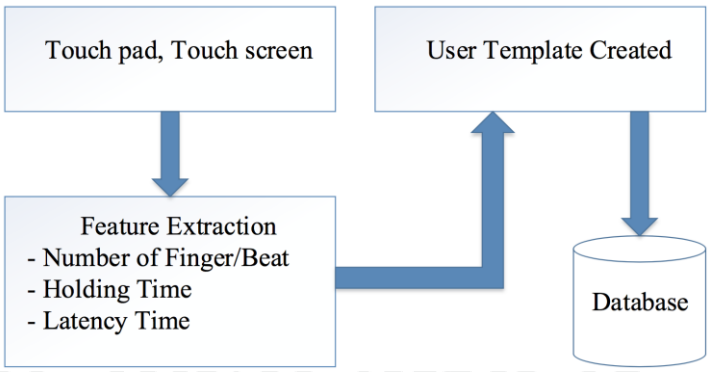


Figure 4.10 The registrations flowchart of the minimized version of Rhythmprint authentication

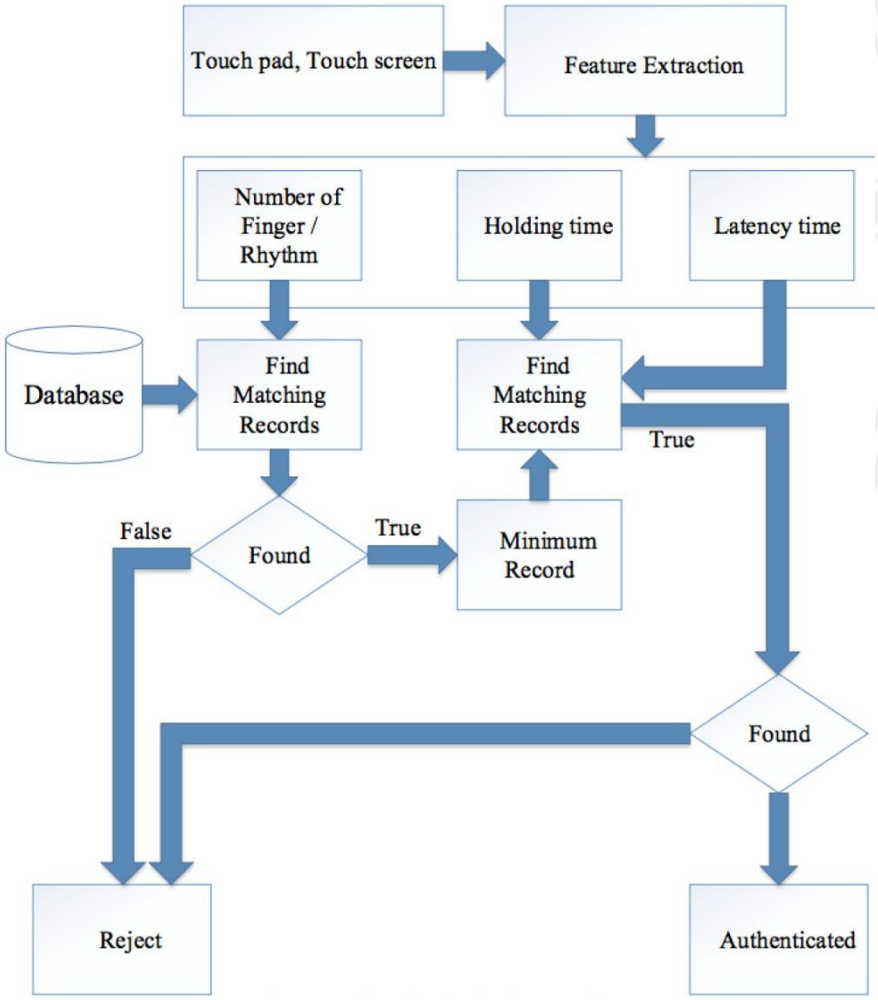


Figure 4.11 The authentication process of the minimized version of Rhythmprint authentication

This experiment attempts to measure the minimized version of Rhythmprint authentication's security in terms of defending against shoulder surfing and eavesdropping attacks by comparing it to a traditional keystroke authentication. We implemented all authentication programs for the minimized version of Rhythmprint authentication and the traditional keystroke authentication on laptops. For the experimental design, we again simulate the situation where the user must authenticate himself/herself to an application on a laptop while sitting in a public place and an attacker stands behind the user. The attacker again stands behind the user at all times while the user is trying to authenticate himself/herself using the minimized version of the Rhythmprint authentication and the traditional keystroke authentication. For each method, the user must try to authenticate 10 times, while the attacker has to perform a shoulder surfing and an eavesdropping attack immediately after every successful attempt. The user enters the password on the keyboard for the traditional keystroke authentication and makes the rhythm on the touchpad for the minimized version of the Rythmprint authentication. Every time the user can authenticate successfully, we tested whether or not the attacker is able to authenticate on the victim's laptop. The results are recorded. Figure 4.12 shows the simulated situation of shoulder surfing attacks.

Figure 4.12 The simulated situation of shoulder surfing attacks

We do not allow the user to use the other hand to cover the touching hand when tapping on the touchpad or typing on the keyboard. Figure 4.13 shows when the user tries to

authenticate to an application on a laptop with the traditional keystroke method without hand covering while Figure 4.14 shows when the user tries to authenticate to the application on a laptop with the minimized version of Rhythmprint method without hand covering.

Figure 4.13 The user tries to authenticate to the application on a laptop with the traditional keystroke method without covering it with the other hand.

Figure 4.14 The user tries to authenticate to the application on a laptop with the minimized version of Rhythmprint authentication without covering it with the other hand.

For verifying our experimental method, we designed and developed a piece of software on a laptop using Java programming language. For a laptop in our experiment, we used a Macbook Pro produced by Apple Inc. We recruited 10 participants which consists of five males and five females, aged between 30-40 and one male as an attacker. All users must authenticate themselves to the application on our laptop 10 times per method, while the attacker is standing behind them. Each time the authentication is complete and successful, the attacker immediately tries to authenticate himself on the designated laptop.

Table 4.5 The experimental result of the security challenge between the minimized version of Rhythmprint and the traditional Keystroke authentication

User	Attack Success Time	
	Rhythmprint minimize version	Keystroke
1	Unable to attack	3
2	8	5
3	Unable to attack	4
4	Unable to attack	2
5	7	4
6	Unable to attack	3
7	7	5
8	Unable to attack	1
9	9	7
10	Unable to attack	8

From Table 4.5, The experimental results show the security performance of the minimized version of Rhythmprint authentication compared to the traditional keystroke authentication. only 4/10 volunteers using the minimized version of Rhythmprint authentication were attacked successfully and the minimum time to successfully attack the victim's rhythm was 7, while all participants using the traditional keystroke

authentication were attacked and the minimum time to successfully attack the victim stroke was only 1.

From the above experimental results we can conclude that the minimized version of Rhythmprint authentication also provides higher security than the traditional keystroke authentication method. However, it is still inferior when compared with the full version of Rhythmprint authentication. Thus, we can conclude that the full version of Rhythmprint is more secure than both the minimized version of the Rhythmprint authentication and the traditional keystroke authentication in terms of defending against shoulder surfing and eavesdropping attacks.

4.4 User satisfaction survey report

To ensure that Rhythmprint authentication can be used in real life, we created a questionnaire to assess the satisfaction of the users towards the use of Rhythmprint. There topics were used in our questionnaire's likert scale:

- System response time
- User friendly
- Security performance
- The Rhythmprint can used in real life
- Shoulder surfing attacks prevent
- Eavesdropping attacks prevent

All of the topics above were scored between 1-5, 1 is defined as needing improvement, 2 is fair, 3 is moderate, 4 is good and 5 is excellent. Aside from the likert scale in the questionnaire, we have a simple yes or no question which is: "*If Rhythmprint must be installed on the user's computer Are the user willing to use Rhythmprint instead of the existing authentication method or not*."

Meaning	Excellent
4.51-5.00	Good
3.51-4.50	Moderate

2.51-3.50	Fair
1.51-2.50	Need improvement
<= 1.50	

We recruited 30 participants which consists of fifteen males and fifteen females, aged between 20-25 with all of the participants currently studying for a Bachelor's degree in computer engineering or related IT departments. The results of the user questionnaire is shown in Table 4.6.

Table 4.6 The questionnaire of the user satisfaction survey result

Topic	Satisfaction level				
	5	4	3	2	1
System response time	19	6			
User friendly	14	9	2		
Security performance	17	8			
The Rhythmprint can used in real life	12	11	2		
Shoulder surfing attacks prevent	18	7			
Eavesdropping attacks prevent	16	9			
Accumulated frequency (Number of respondents/Number of question)	16	8	1		
Score (Accumulated frequency x Satisfaction level)	80	32	3		
Total score			115		
Average (Total score/Number of participants)			4.6		

The average score is 4.6, when compared with assessment criteria the result of user satisfaction is *Excellent*, and the result of a question (the answer must only yes or no): *If Rhythmprint must be installed on the user's computer, are the user willing to use Rhythmprint instead of the existing authentication method or not* is 25 of 25 answer : yes.

CHAPTER 5 Conclusion

The Rhythmprint authentication method can both defend against shoulder surfing attacks while being more convenient to use. This is because users don't need to use the other hand to cover the tapping hand to prevent shoulder surfing attacks. Even when an attacker can see which fingers and the number of fingers that were used to generate each beat, the attacker cannot impersonate the user. In the experiment, we asked all volunteers to use the same rhythm to create their templates but the FAR is still extremely low. The shape and size of the fingers are unique to an individual, therefore, the distance between finger tips when making taps are naturally unique.

Eavesdropping attacks are also unlikely because tapping on the touchpad does not make an audible sound. Even if the attacker knows the rhythm, the number of fingers per beat and which fingers were used for each beat, the distance between fingertips cannot be easily copied due to the hand geometry. However, if we allow the user to use his/her own rhythm the percentage of accuracy score will be lower than results found in chapter 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 up 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 rate will be decreased.

Rhythmprint is a multifactor authentication method because the user must know the rhythm and number of fingers per beat (what you know) and the user must have the same hand which were used in the previous registration (what you have) because each person has different hand geometry. Therefore, the distance between the fingertips of the user is different when his/her fingers touches on an input device (what you are)

From the experimental results, we can conclude the Rhythmprint authentication and the minimized version of Rhythmprint provides higher security than the traditional keystroke authentication and other related works that was explained and compared in chapter 2 in terms of defending against shoulder surfing and eavesdropping attacks. For the Rhythmprint authentication, when the user touches or taps on the touchpad, it does

not make an audible sound. Therefore, eavesdropping attacks are also extremely unlikely.

BIBLIOGRAPHY

Aloul, F.; Zahidi, S. and El-hajj, W. 2009. Two factor authentication using mobile phones. **In: 2009 IEEE/ACS International Conference on Computer Systems and Applications**, 10-13 May 2009, Rabat, Morocco. Pp. 641-644.

Antal, M.; Szabó, L. Z. and László, I. 2015. Keystroke dynamics on android platform. **Procedia Technology**. 19: 820-826.

Azenkot, S.; Rector, K.; Ladner, R. and Wobbrock, J. 2012. PassChords: secure multi-touch authentication for blind people. Paper presented at the **14th International ACM SIGACCESS Conference on Computers and Accessibility**, 22-24 October 2012, Boulder, CO, USA. Pp. 159-166.

Babich, A. 2012**. Biometric Authentication. Types of biometric identifiers.**

Bhattacharyya, D.; Ranjan, R.; Alisherov, F. and Choi, M. 2009. Biometric authentication: A review. **International Journal of u-and e-Service, Science and Technology**. 2(3): 13-28.

De Luca, A.; Von Zezschwitz, E.; Nguyen, N. D. H.; Maurer, M. E.; Rubegni, E.; Scipioni, M. P. and Langheinrich, M. 2013. Back-of-device authentication on smartphones. Paper presented at the **SIGCHI Conference on Human Factors in Computing Systems**, 27 April – 2 May 2013, Paris, France. Pp. 2389-2398.

Espinoza, M.; Champod, C. and Margot, P. 2011. Vulnerabilities of fingerprint reader to fake fingerprints attacks. **Forensic science international**. 204(1-3): 41-49.

Galbally, J.; Cappelli, R.; Lumini, A.; Maltoni, D. and Fierrez, J. 2008. Fake fingertip generation from a minutiae template. **In 2008 19th International Conference on Pattern Recognition**, 8 December 2008, Florida, USA. Pp. 1-4.

Galbally, J.; Cappelli, R.; Lumini, A.; Gonzalez-de-Rivera, G.; Maltoni, D.; Fierrez, J. and Maio, D. 2010. An evaluation of direct attacks using fake fingers generated from ISO templates. **Pattern Recognition Letters**. 31(8): 725-732.

Huang, X.; Lund, G. and Sapeluk, A. 2012. Development of a typing behaviour recognition mechanism on android. Paper presented at the **11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications**, 25-27 June 2012, Liverpool, United Kingdom. Pp. 1342-1347.

Kant, C. and Nath, R. 2009. Reducing process-time for fingerprint identification system. **International Journals of Biometric and Bioinformatics**. 3(1): 1-9.

Kim, J. J. and Hong, S. P. 2011. A method of risk assessment for multi-factor authentication. **Journal of Information Processing Systems**. 7(1): 187-198.

Leiva, L. A. and Català, A. 2014. BoD taps: an improved back-of-device authentication technique on smartphones. Paper presented at the **16th International Conference on Human-Computer Interaction with Mobile Devices & Services**, 23-26 September 2014, Toronto, ON, Canada. Pp. 63-66.

Mainka, C.; Mladenov, V.; Guenther, T. and Schwenk, J. 2015. Automatic recognition, processing and attacking of single sign-on protocols with burp suite. **Lecture Notes in Informatics (LNI)**. 251(November): 117-131.

Maiorana, E.; Campisi, P.; González-Carballo, N. and Neri, A. 2011. Keystroke dynamics authentication for mobile phones. Paper presented at the **2011 ACM Symposium on Applied Computing**, 21-24 March 2011, TaiChung, Taiwan. Pp. 21-26.

Marques, D.; Guerreiro, T.; Duarte, L. and Carriço, L. 2013. Under the table: tap authentication for smartphones. Paper presented at the **27th International BCS Human Computer Interaction Conference Article**, 9-13 September 2013, London, United Kingdom. Pp. 33-39.

Roth, V.; Schmidt, P. and Güldenring, B. 2010. The IR ring: authenticating users' touches on a multi-touch display. Paper presented at the **the 23nd annual ACM symposium on User interface software and technology**, 3-6 October 2010, New York, New York, USA. Pp. 259-262.

Saevanee, H. and Bhatarakosol, P. 2008. User authentication using combination of behavioral biometrics over the touchpad acting like touch screen of mobile

device. Paper presented at the **International Conference on Computer and Electrical Engineering Communications (ICCEE)**, 20-22 December 2008, Phuket, Thailand. Pp. 82-86.

Shanmugapriya, D. and Padmavathi, G. 2009. A survey of biometric keystroke dynamics: Approaches, security and challenges. **International Journal of Computer Science & Information Security**. 5(1): 115-119.

Subpratatsavee, P. and Kuacharoen, P. 2015. Transaction authentication using HMAC-based one-time password and QR code. **Lecture Notes in Electrical Engineering**. 330(January): 93-98.

Takada, T. and Kokubun, Y. 2013. Extended pin authentication scheme allowing multi-touch key input. Paper presented at the **International Conference on Advances in Mobile Computing & Multimedia**, 2-4 December 2013, Vienna, Austria. Pp. 307-310.

Zhang, D. and Kanhangad, V. 2011. Hand geometry recognition. **Encyclopedia of Cryptography and Security**. 2011: 529-531.

BIOGRAPHY

NAME	Nakinthorn Wongnarukane
ACADEMIC BACKGROUND	Bachelor's Degree with a major in Computer Science from Kasetsart University Siracha Campus, Chonburi, Thailand in 2010 with class honors and a Master's Degree in Computer Science at National Institute Development Administration (NIDA), Bangkok, Thailand in 2013 with class honors
EXPERIENCES	

ปญฺญาสมา ธนา
สำนักบัณฑิตอาสาสมัคร