2018 Master's Thesis

PINWrite: A New Smartphone Authentication Scheme Using Handwriting Recognition

Supervisor Professor Tatsuya Mori

A Thesis Submitted to the Department of Computer Science and Communications Engineering, the Graduate School of Fundamental Science and Engineering of Waseda University in Partial Fulfillment of the Requirements for the Degree of Master of Engineering

Student ID :5116FG12

Jing Peiqing

Research Guidance: The Research on Networked Systems

Date： 2018/7/24

Abstract

Recently, with the rapid expansion of smartphone accompanied by the Internet, the smartphone has influenced a lot in humans' daily life. As the first barrier to smartphone security, authentication method works like an important security guard, who can protect the data in smartphone safely. The most common authentication method now is the PIN code, easy and quick to use while suffers a lot of attacks like smudge attack and shoulder surfing attack.

This paper introduces a new authentication method named PINWrite based on handwriting method. To use this system, a user should first select a 4-digit password and write each digit 3 times through setting part. After that, the user just needs to write this 4-digit password on the screen when he/she wants to unlock their smartphone. As the user makes use of her/his handwriting digits, rather than only tapping PIN code, the PINWrite system provides more defense to different kind of attacks. We also note that the behavioral biometrics such as finger movement can work as an extra authentication factor, let this system offer better security. As users are already familiar with the traditional PIN code authentication mechanism, they need less effort to learn our system.

To check the usability and security of our system, we have done two user studies and collect digit samples from 35 users. By calculating the successful rate, error rate, and other properties, this paper shows the performance of the PINWrite system and how it works to defend the different attacks such as the content-aware attack and observer attack.

Contents

List of Figures

List of Tables

Chapter 1 Introduction

Smartphone has become an indispensable means of communication used in the people ’ s daily lives. Ref. [3] reported that there are 77% of the United States adults who own at least one smartphone device. Not just using it for calling or texting, many smartphone users also use it for online shopping and social networking. In other words, smartphones are increasingly responsible for the sensitive information including personal photos, email, online banking accounts, etc. [4]. Thus, protecting privacy-sensitive information collected on the smartphone has become one of the critical issues for many users.

Smartphones need to have a strong authentication mechanism to protect the data from being stolen by an attacker. To this end, a majority of smartphone users are using authentication mechanisms to lock/unlock their smartphones, i.e., PIN code, fingerprint scanner, password, the pattern of dots, and some other types [5]. Unfortunately, even most users select one or two authentication methods to protect their smartphone; they are still prone to many risks. For example, the PIN code, known as the most simple and quick way to (un)lock smartphone could suffer from the shoulder surfing attack [6], especially under crowded conditions. Besides, after typing it, the oily residues, also known as smudges, will remain on the screen. The malicious users can use these smudges to perform an attack called “smudge attack” [7]. Not surprisingly, the same attacks can also happen to other knowledge-based authentication like the pattern of dots and password.

In addition to the knowledge-based authentication methods, many smartphone providers like iPhone started to use the biometric feature to do the authenticate work. For example, the fingerprint recognition (Touch ID) and the face recognition (Face ID). Although these biometric approaches are often supposed to be very secret since they are not vulnerable to observation and the “password” can ’t be forgotten by users [8], they still exist some problems like usability and security issues. As the fingerprint or face recognition on a smartphone requires extra hardware, these methodologies cannot be shipped by a vendor who does not own such technologies.

Besides, some users could feel difficult to use these methods, since they may have weak fingerprint so the sensor cannot recognize their fingerprint, while some users may feel awkward when they use Face ID as they found that it is uncomfortable to hold a smartphone in front of their face when there is someone closed to them [9]. For security issue, the Touch ID has already shown to be easy to break by using 2D Printed Fingerprints [10]. Furthermore, all the biometric approaches still need to use a knowledge-based method like PIN code to work as an alternative authentication mechanism in case the biometric authentication methods fail; i.e., an attacker could across these physical approaches by merely invoking the alternative method like the PIN. These observations highlight that we still need to pay efforts to enhance the knowledge-based authentication which works as a base authentication method on smartphones.

1.1 Research Purposes

In this paper, we present a framework named PINWrite. It aims to establish a usable and reasonably secure authentication system by combining behavioral biometrics feature with the knowledge-based authentication. The key point is to let the user write the password on the screen rather than just click the keyboard when unlocking their smartphone. By adding the handwriting feature to the password, a large number of personalized passwords can be created since we proposed that different users could have different handwriting. As a result, the PINWrite system can use both password and the writing behavior to give a higher security level to the standard knowledge-based authentication. Figure 1.1 provides an overview of the PINWrite system.

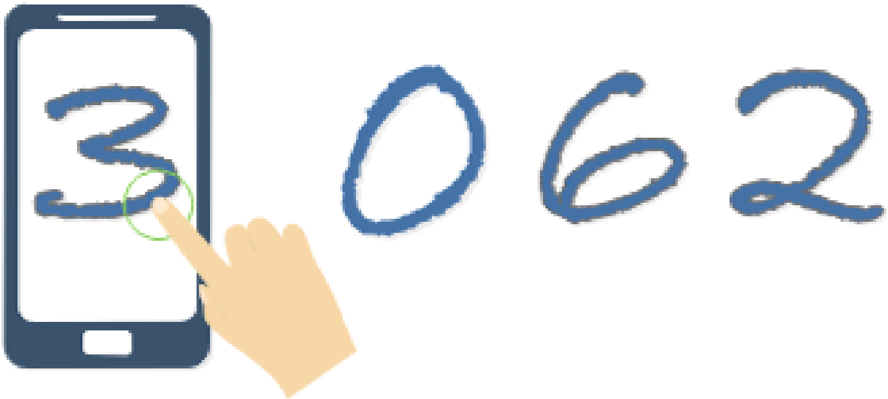


Fig. 1.1 How PINWrite system works on smartphone:users are asked to write their password on the screen one by one instead of typing the keyboard.

PINWrite provides several benefits that are not in the conventional knowledge-based authentication mechanisms. First, this system gives higher resistance to the smudge attack. Since PINWrite asks the user to write down password digit one by one when unlocking a smartphone, a new digit input by a user will overwrite the last digit they have written. As the smudge left on screen is not clear, an attacker may not be able to recover the input digits. Besides, PINWrite is tolerant to the shoulder surfing attack. Because we combine the handwriting and password, even though an attacker observes the password user input on a screen, it is not easy to mimic the handwriting as the user ’ s input. To prove it, we did a small human study. We ask participants to write down each password several times and let other participants be an attacker and try to mimic their handwriting. We found that even an attacker knows the right password and how the handwriting looks like, they still can ’ t unlock smartphone in most cases. Another advantage of this system is that the new system is easy to learn. People are familiar with writing digits in the daily life. They are also familiar with using the password authenticate method. These observations ensure that our system is easy to learn. Users also do not need many efforts and time to get used to this system.

1.2 Main Contribution

The contributions of this paper are as follows:

1. We propose an enhancement to a standard passcode-based authentication mechanism with the behavior biometric feature handwriting; our system provides higher resistant to smudge and shoulder surfing attack.
2. We implemented the PINWrite system by using two different technologies. To analyze a handwritten digit, we adopt the Convolutional Neural Network (CNN). For the handwrit-

ing, we used the ORB (Oriented FAST and Rotated BRIEF) to constitute the handwriting analyzer.

3. We evaluated the effectiveness of the PINWrite framework through human studies. We aim to test how it works to the attack that may happen to this system and the overall performance. The results demonstrated that PINWrite could reject most attacks successfully and works well through the studies.

1.3 Constitutes of Paper

The rest of the paper is organized as follows. In Section 2, we introduce the background of this research like the authentication method types and introduce behavior biometrics in detail. An overview and the specific methodologies we used in the PINWrite system are presented in Section 3. Section 4 shows the way we evaluate the PINWrite system like the possible attacks and the data collection, then shows the result we got from the experiment including the performance of different components, time-consuming and usability. Related works are summarized in Section 5. Finally, in section 6, we conclude this paper and discuss some future research topics.

Chapter 2 Background

In this section, we briefly introduce some technical preliminaries including the authentication mechanisms, the problems faced by existing mechanisms and the concept of behavioral biometrics.

2.1 Authentication Mechanism

Not surprising that there are lots of work related to authentication mechanisms has been down in the few past years since the necessary of smartphone protection and the practical issues. Generally speaking, the proposed authentication mechanisms can be traditionally divided into three categories as shown below [11]:

- Knowledge-based Authentication: For example, the PIN code and password. This method is based on something that user knows, which means it will ask users some specific questions and gain access by verifying these answers are correct or not. Knowledge-based authentication has become prevalent since it is the simplest authentication method and also effective to all the users to manage their authorization. Although Knowledge-based Authentication is used by most people these days, this technique is quite easy to break since users often share their personal information on the Internet like the birthday or some memorial days that are always used as the password. By looking through the users' social network, attackers can get all the information they want with a little effort.
- Possession Based Authentication: For example, the token (software or specific device). This method is based on something that a user has, which is used to check users' validity. Token will provide users a login credential in a specific period, hence the permission can be granted to the users without input their password by using the token. This technique offers security in the way that token will expire after an amount of time, so the users need to log in again to get the access. On the other hand, some disadvantages should also be taken into consideration, the most obvious problem is that it will cost higher implement to finish the authentication procedure because users need to take extra device or software and need to keep it safe, which is very inconvenient.
- Biometric Based Authentication: This method is based on something that user is which is the most technologically advanced solution today [12]. There are many components of our body that can be used in this method, such as fingerprint, iris, and face. To recognize these parts, smartphone sometimes needs an extra scanner like the fingerprint scanner, but for face and iris, only use the camera smartphone already have is OK. These technique makes the authentication easier for users since they do not need to remember something or take something with them. However, a significant drawback is that, compared to the other two authentication method, the false rate can be relatively high. Besides, in case user's password has been leaked, they can reset their password easily while they can't change their body part

when fingerprint or iris has been compromised or duplicated.

2.2 Behavioral Biometrics

Behavioral biometrics is a cutting-edge technology that provides us with a new way to figure out the security problems by using the interaction between users and smartphones to identify individuals.

According to how user holds or move the smartphone or how the user's finger-movement on screen, the program using behavioral biometrics can create a user profile including all the behavioral data collected by smartphone like pressure, tremors, navigation, scrolling and other movement arguments [13]. Hence, the smartphone can tell the user's identity by comparing this profile with the user's following iteration.

One of the greatest strengths of using the behavioral biometrics is that although it cannot replace the password or other knowledge-based identity authentication, it does add some features to make it easier to protect the sensitive information. For example, if you want to have a strong password, the only way you can do is to make your password as long as possible or add many symbols to make your password complicate enough. However, by using the behavioral biometrics and providing an additional layer of identity assurance, users can get enough security while not to use an extreme complicate password at the same time.

behavioral biometrics contains many types, which means it can use a number of behavioral traits to identify the user through pattern recognition. There are mainly two subdivisions that used most prevalent currently, speaker recognition and signature recognition.

Speaker recognition, obviously, is using the human voice and the individual information contains in speech signals such as the movement of jaws, tongue, and larynx to automatically recognize who is speaking [14]. There are several good points, for instance, the voice is easy to be recorded by smartphone and does not need extra equipment. However, to some people who have the problem with their voice, this method seems not usable.

Signature recognition is the process of analyzing the way that user write on the screen including the physical activities like stroke order, the speed, and pressure. The handwriting we talked about in this paper can also be seen as a part of it. The merit of signature recognition is that it is difficult to mimic since everyone has a unique writing style and people are used to signing so the system is practical for everyone to use. But this system still has some weakness that people may not always write in a consistent way since they may write by different hand or in different conditions [15].

Chapter 3 Implementation of PINWrite System

Our idea is to use the handwriting which is known as a behavioral biometric feature on the basis of the PIN-based authentication to create a new authentication method. In the following, we first introduce the overview of the whole system and then discuss the detailed implementation method of it.

3.1 Overview of PINWrite System

PINWrite system contains two phases: the setting phase and the authentication phase as shown in Figure 3.1.

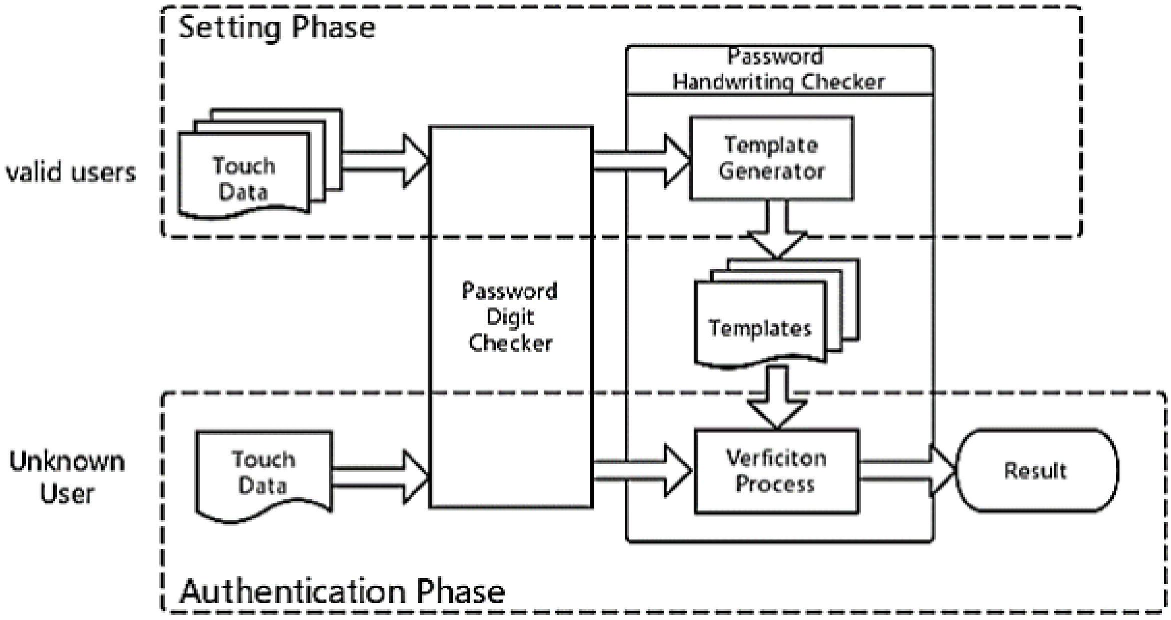


Fig. 3.1 Overview of PINWrite system. The component of PINWrite consist of a Password Digit Checker (PDC) and a Password Handwriting Checker (PHC)

In the setting phase, a valid user first need to choose a 4-digit PIN as the password, then the user is asked to write this 4-digit PIN for several times on the touch screen, here we set the number as three. After the user's input, the PINWrite system will extract information like the x-y-coordinates and pixel data from each written sample. By using these data, the system will analyze all the input sample and save it as the image in the smartphone as the alternative templates for authentication.

In the authentication phase, when an unknown user wants to unlock the smartphone, the user needs to write the 4-digit PIN in a correct way on screen. To check if the login sample is correct or not, the system will first recognize each digit from the unknown user's input and then check if it's the same PIN with valid user set in setting phase. If the PIN itself is incorrect, the system will reject the unknown user without checking their handwriting. If the PIN is correct, the PINWrite system will then verify the handwriting of the digit by calculating a similarity score between login sample with the template provided by setting phase. If the similarity score is higher than the threshold system set, the unknown user will get permission to unlock the smartphone.

In this way, PINWrite system could use two factors to authenticate an unknown user, the PIN and the user's behavioral characteristics handwriting, which gives users a double protection to their smartphone. To achieve these function, we create two subsystems named Password Digit Checker (PDC) and Password Handwriting Checker (PHC). In short, PDC is used to recognize the digit from user's input and verify the correctness of PIN, while PHC is used to analyze the handwriting to see if the PIN that has passed PDC is written by the same user. The detailed implementation methodology will be given in the next sections (Section 3.2).

3.2 Methodology

3.2.1 Password Digit Checker (PDC)

As we said above, the Password Digit Checker is responsible for checking if PIN written on the screen is the same as the user set before. It mainly contains two parts to realize this function, Digit Recognizer and PIN Checker.

To set an effective Digit Recognizer, the first step is seeking a dataset to do the training part. In this step, we use a handwriting database named MNIST Database hosted on Yann LeCun's website [1]. The MNIST Database is a large database that contains 60000 training data and 10000 test data, and each MNIST data point includes mainly two parts: a handwritten digit image with a corresponding label telling which digit it is. In this database, all the digits are written in black and white and have already been normalized and centered in a fixed size image with 28 × 28 pixels as shown in Figure 3.2, which is very suitable for the data training.

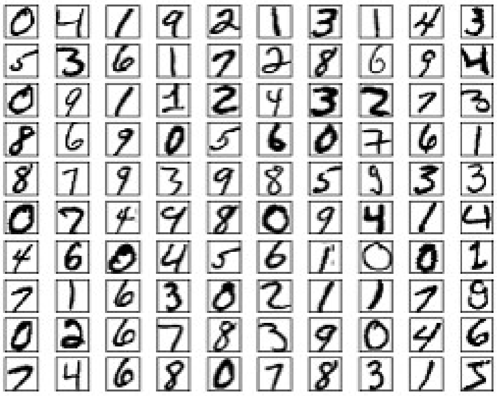


Fig. 3.2 The handwritten digit in MNIST Database [1]

After finding an appropriate dataset, the next step is to train this dataset. In PDC, we used a simple Convolutional Neural Network shown in Figure 3.3 that contains two convolutional layers works to calculate a single value to the output feature map from each sub-region of input image by performing a cluster of mathematical operations , two max-pooling layers that extracts sub-regions of the feature map and only reserving the maximum value,a fully-connected layer to do processing on the entire image and a readout layer to do the SoftMax regression that help us classify the digit from 0 to 9.

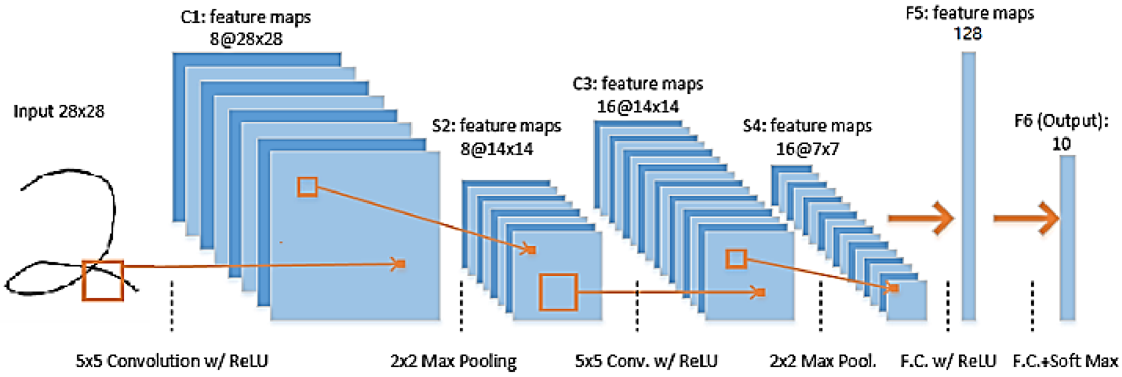


Fig. 3.3 The constitution of Convolutional Neural Network [2]

Typically, the digit recognition including three main processing steps: preprocessing, feature extraction and classification. [16] We basically follow these steps while some adjustments have been made in our system.

- Preprocessing - To a general image, the preprocessing is important since the key signal should be separated from the noise, which will make the following steps more effective and robust. The noise in handwriting can always be the differences in size, the slant, thickness of strokes and other variables. However, in our system, we avoid these noises from emerging. As stated above, in MNIST Database, the handwritten images are size-normalized and centered in a fixed-size image. On the other hand, in both setting phase and authentication phase of the PINWrite system, we ask the user to write the digit on a 28 × 28 size canvas which is in the same size with image in MNIST Database.
  To make the data suitable for the feature extraction and classification step, we need to convert the format of images to a big array including 28 × 28=784 numbers. Each element in this array represents the intensity of a particular pixel in the particular image, between 0 and 1.
- Feature extraction - The feature extractor is used to generate a set of features, making the classification algorithm are able to get a better result when distinguishing various characters. Here we use the convolution and pooling layers in the Convolutional Neural Network to accomplish this step, making the data much simpler and more representative for the classification algorithm to use.
- Classification - In PINWrite system, we use the SoftMax regressions belongs to read out layer in the convolutional neural network as the classification algorithm. Given a set of training samples whose respective class is already known, a classifier can estimate the distribution of various classes in pixels. So, when the classifier meets a new pattern, it will estimate all the probabilities for each class, which is the work of SoftMax who can provide us a list of value from zero to one that adds up to one. For example, the given pattern could be an image of six, while the classier gives 90% probability to six, it still can give a chance to be known as other digits, like 5% to eight since the classifier isn't 100% sure.
  A SoftMax regression contains two processes [17]. The first step is to add up the evidence

that the given pattern belongs to the certain classes. The evidence can be got from a weighted sum of the pixel intensities. If the pixel in a given pattern has a higher intensity, the weight is negative while the weight is positive if the pixel has a lower intensity. So, the result of evidence for class i and given pattern x can be:

$$evidence_i = \sum_j W_{i,j} x_j + b_i \tag{3.1}$$

W_i is the weights and b_i is the extra evidence named bias for class i, and j is an index for summing up all the pixels in the given pattern x.
The next step is to convert the evidence we got into probabilities by using the SoftMax function. The SoftMax will first exponentiate the inputs evidence which means the change of evidence will give multiple effect to the weight of hypothesis, then the SoftMax will normalize these weights, from 0 to 1 and add up to 1, producing a valid probability distribution. In the end, the highest probability class will be identified as the correct number of the given pattern.

After recognizing the digit from user input, PIN Checker, the second part of PDC, will receive a list of digits from Digit Recognizer. If both content and the sequence are same with the 4-digit PIN in setting phase, this user's input could be seen as valid and be transmitted to next step, the Password Handwriting Checker. Otherwise, the user will reject by the system and asked to log in again.

3.2.2 Password Handwriting Checker (PHC)

The Password Handwriting Checker plays an important role in the whole PINWrite system because we need to judge whether a user is valid mainly by distinguishing the handwriting of users' input. In this section, we describe how PHC works in details.

Once the user ' s input passed the verification of PDC, the images of the digit that user have written on the touchscreen will be sent to this component. As we said in the last section, we set the canvas size as 28×28 which is good for digit recognize but difficult to process, so we change the size to 180×180 when saving it as images. By using these images, the PHC will verify users' handwriting in two procedures: Similarity Calculation and Verification. Given a set of digit images, the way to calculate the similarity of handwriting between each image is to compare them in pairs. To do the comparison, the first step is extracting the features from each image and then matching these features. Here we first use a technique named ORB, Oriented FAST and Rotated BRIEF, which is the most efficient algorithm for image processing [18].

The features can be simply regarded as some outstanding points of a image, like the contour point, the light point in dark space or the dark point in light space. ORB use the FAST (features from accelerated segment test) algorithm to detect feature points by setting an intensity threshold between the centroid pixel and the circle of pixels around it. We use a formula to define whether a point is the feature point:

$$N = \sum_{x \forall (circle(p))} |I(x) - I(p)| > \varepsilon_d \tag{3.2}$$

I(x)is the intensity of random pixel in the circle, I(p)is the intensity of the centroid pixel, ε_d is the threshold, if n is higher than the threshold, always set as 3/4 of all the circle points, the point p will be regarded as a candidate feature point. Then ORB algorithm will apply a method called Harris [19] to calculate a Harris score of all candidate feature points have been detected, sort and then select top N points which are most suitable.

After getting the feature points, we need to use some method to describe the attribute of the feature points, which are known as feature descriptors. The ORB BRIEF algorithm is used to calculate the descriptor of a feature point. The main idea is to choose N counterpoints following a certain pattern around the feature point P and comparison results of N counterpoints will be combined as the descriptor of P. For example, we set N=2, so we will choose 2 groups of counterpoints like $P_1(A, B)$ and $P_2(A, B)$. Then, we will do the comparison between these counterpoints, the result can be

$$\text{T(P(A,B))=}\begin{cases} 0 & I_A > I_B \\ 1 & I_A \leq I_B \end{cases} \tag{3.3}$$

T is the result of the comparison, I_A is the intensity of point A and I_B is the intensity of point B. So the descriptor of P could be $\{T(P_1(A, B)), T(P_2(A, B))\}$.

Instead of just giving simple descriptors to the feature points, we also use a method named intensity centroid [20] to add the orientation to them. This method assumes that a corner 's intensity is offset from its center, so the orientation is from this corner point to centroid. For every feature point, it first defines the moments of a patch (neighborhood pixels) as:

$$m_{pq} = \sum_{x,y} x_p y_q I(x, y) \tag{3.4}$$

(x, y) is in the position of the feature point, and I(x,y) is the intensity of (x,y). Then we can find the centroid by these moments:

$$C = (\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}) \tag{3.5}$$

After find the centroid, the orientation can be simply defended as the angle formed by feature point and the centroid point:

$$\Theta = arctan(m_{01}, m_{10}) \tag{3.6}$$

The biggest characteristic of the ORB algorithm is the computing speed, which is benefit from oFAST, since FAST can detect the feature points in a quick speed just like its name. Besides, the BRIEF use binary string to describe the feature descriptors not only saving the storage space, but also shorten the time of matching. For example, A: 10101010 and B: 10101011 are two descriptors of feature points A and B. We can match A and B points by verifying its similarity. In this example, the descriptor of A and B only have one difference, so the similarity could be 87.5%, if we set the threshold as 80%, then A and B can be seen as same feature points and match to each other.

Figure 3.4 shows how ORB works when comparing two digits image written by the same person. In this figure, we find that most feature points could match to each other correctly while still exist some wrong matching points. These wrong matching may somehow affect the verification of PHC, so we add an algorithm named RANSAC (random sampling consensus) [21] to eliminate these mismatches.

RANSAC is a method of robust estimation proposed by M. A. Fischler in 1981. It can remove outliers by computing model parameters based on random voting principle even the space of data containing more than half of the noise points. Meanwhile RANSAC could also effectively deal with multiple structure data. [22] Figure 3.5 shows a fitted line performed by RANSAC, where outliers have no influence on the result.

After employing the RANSAC method, most mismatching points were removed as shown in Figure 3.6, making the result of comparison more accurate and satisfactory.

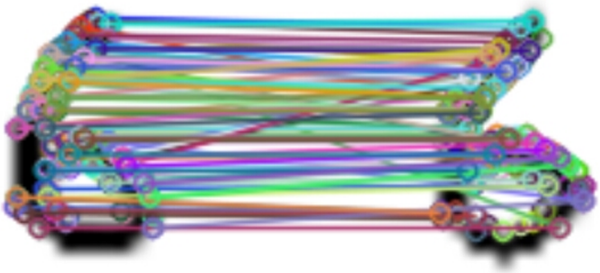


Fig. 3.4 The matching points between two digit by employing ORB methodology

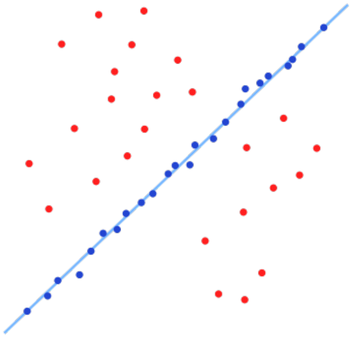


Fig. 3.5 Fitted line with RANSAC;Outliers have no influence on the result.

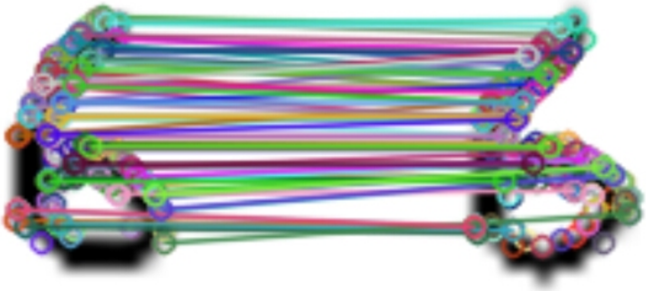


Fig. 3.6 The matching points between two digit by employing ORB combining with RANSAC methodology.

Through the number of matching points by comparing handwritten digit images, we can simply calculate the similarity of the handwriting by calculating the proportion of matching points in all feature points.

To verify handwriting between the templates in setting phase and the login sample in the authentication phase, PHC serves different functions in different phases.

- In setting phase, PHC is used to determine a reference template based on the templates set T:T=$\{T_i\}$ for each digit that we ask users to write in the setting phase. Because user is asked to write each digit for three times, the template set will contain three element $\{T_1, T_2, T_3\}$. The first step is to calculate the average pairwise matching rate between T_i and the rest templates for every template T_i in T.

$$Rate(T_i, T) = \frac{N(matchingpoints(T_i, T))}{N(featurepoints(T_i))} \tag{3.7}$$

Then, the template with the biggest average rate will be chosen as the reference template T_{ref} that will be used to represent the whole templates T to compare with the login sample in the authentication phase.

- In the authentication phase, to verify the valid user, we need drive a similarity score of the login sample that indicates whether the login sample has the same handwriting with the template set before. Thus, by comparing the matching rate between the login sample P and the selected templates T_{ref} with the biggest average rate we calculated in the last step, we can get the final similarity score of the unknown user.

$$Score(T_i, T) = \frac{Rate(T_{ref}, P)}{Avg(Rate(T_{ref}, T))} \tag{3.8}$$

Thus, when analyzing the similarity of handwriting, we can use this score to verify the valid user.

By combining these two parts, PDC and PHC, PINWrite system are able to verify a real user by first checking the password itself and then detect the digit's handwriting of the password, realizing a double defense to the smartphone. Highly improved the security of the traditional PIN code authentication, making users feel more secure about their smartphone.

Chapter 4 Main Study and Result

In this chapter, we have done two user studies and a questionnaire to evaluate the PINWrite system. It first introduces the procedure of user studies and the details about data collection. Then evaluate the PINWrite in different aspects and presents in the following sections.

4.1 Study Procedure

Experiments were designed to simulate the attacks and the real use of the written PINs. Mainly two experiments are conducted in this study, the first aims to know how this system works to different attacks, we asked 5 participants writing the password assigned to them and let attackers imitate their password. Another evaluation was based on the dataset that collected in the user study that we asked another 30 participants to write the password in a more realistic situation to get the overall performance of the PINWrite system.

Before introducing the approaches of data collection of two experiments, we will first present some attacks may occur to this PINWrrite system that will be tested in the first experiment.

4.1.1 Attack Model

Since PINWrite requires users to write digit one by one on the screen when unlocking the smartphone. The smudge of the newly written digit will wipe the trace of the previous one, which means attackers cannot find a distinct trace of digit by just observing the touchscreen. Thus, we can draw a conclusion that our system performs a higher resistant to the smudge attack.

Apart from the smudge attack, we assume two types of attack that may have some threats to our system.

- Content-Aware Attack: In content-aware attack, the attacker may be users' family members, friends who are familiar with users' personal information that used as the password or strangers who can obtain these information form social network and guess the password based on birthday, luck numbers, etc. In this attack, the adversary only knows the content of password, but hasn't observed handwriting that how user writes the digit on screen.
- Observer Attack: In an observer attack, the adversary may just stand behind user when the authentication takes place, so he/she is able to observe both the number and the writing behavior. Then, the adversary can imitate the users' handwriting.

To obtain an intuition on how well the PINWrite system can resist these two attacks, we have done an experiment and details are present in 4.1.2.

4.1.2 Data Collection

In the first experiment that aims to know how PINWrite system works the content-aware attack and observer attack, we found 5 participants in the university. All participants use smartphone frequently in their daily life and had used the password or PIN method before.

The experiment steps proceeded as follows:

1. First, we will give each participant a unique 4-digit password so that all the digit 0 to 9 could be collected as the sample and be used for analyzing.
2. Then, they were asked to write this 4-digit password for 3 times on the smartphone touchscreen. After 5 min to 10 min, all the participants are asked to input this 4-digit password for 3 times again, considering the influence of the change of the handwriting from the same person. Therefore, in this section, we got 6 samples for each password every participant, totally 30 samples. All the data collected were used as the valid samples.
3. To collect the attack sample, we asked each participant to be the attacker for the rest 4 participant and collecting their input samples for two types of attacks.
    - For Content-Aware Attack, attackers were given a list of passwords that we just assigned to the rest participants and were asked to write down these password on the touchscreen by their own style, since the attackers didn't know the original handwriting of other participants.
    - For Imitation Attack, attackers were given the images of the 4-digit password that we collected as valid input of other participants and were asked to observe each image for 1 seconds for just one time. Then they need to imitate the valid input for each password that shown to them.

   Each attacker was asked to input 3 forgery inputs for each password in each attack, collecting a set of $4 \times 3 \times 5 = 60$ forgery samples for each attack.

For the second experiment, we want to evaluate a overall performance of the PINWrite system. In this study, we found 30 participants (17 males and 13 females), most of them are students from the department of computer science and some of them are from media design. These participants were different from the earlier participants in the first experiment. Since we want to simulate the real use of this application, the steps of this experiment we designed would be the same as the sequence after downloading a new application.

At first, we introduce this study to all participants and give some instructions on the basic use of this application. Different from the first study that we assigned the password to participants, this time we asked users to choose a password by themselves. The distribution of digit choice is shown in Figure 4.1

Since we didn't ask participants to input their real password, most participants selected '1' and '2' in their password. The reason may be these digits are easy to write and cause less time than digits like '8' or '5' that didn't be chosen as much as '1' and '2'.

When users first using this PINWrite system, they were asking to finish the enrollment part by writing their password 3 times for each digit. Then, to get the test data, they were also required to try to unlock this application 3 times after setting part, following the steps of using this system. At the end of these study, 20 of them filled out a questionnaire about their thought with the usability and the security of the PINWrite system. In total, 180 4-digit password samples, including 90 training samples and 90 testing samples, were collected in this experiment.

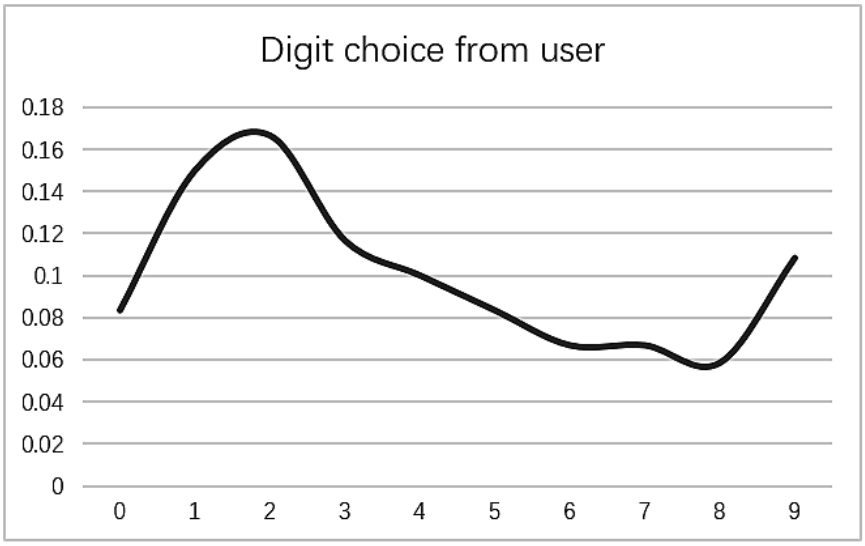
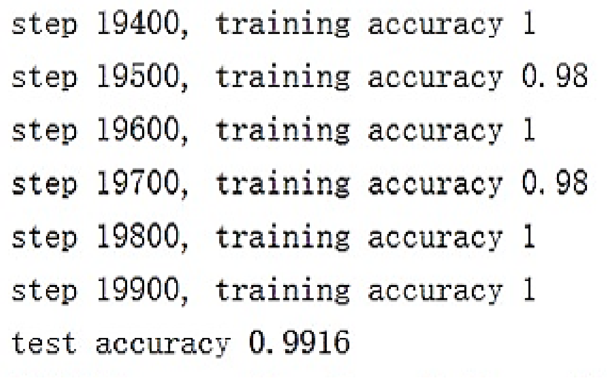


Fig. 4.1 The distribution of digit choice by user

4.2 The Performance of Password Digit Checker

As described in section 3.2.1 the digit recognizer was trained by the training set in the MNIST Dataset. To know the performance of training effect, we add a logging to every 100th iteration in the training process to report the training accuracy, after that we also use the testing set in MNIST Database that includes 10000 points of data to do a final test at the end of training. The result is shown in Figure 4.2.



```
step 19400, training accuracy 1
step 19500, training accuracy 0.98
step 19600, training accuracy 1
step 19700, training accuracy 0.98
step 19800, training accuracy 1
step 19900, training accuracy 1
test accuracy 0.9916
```

Fig. 4.2 The accuracy of trained digit recognizer

We can see that all the accuracies reported in the training process are higher than 0.98 and most of them are 1. And the final accuracy is 0.9916, also shows that the trained digit recognizer can recognize different digit in a very high accuracy.

By using the data we collected in the study and from the MNIST Dataset, we also calculated an average recognition rate for each digit from 0 to 9. The results are shown in Table 4.1.

As we can see in the table, most digit can be recognized successful in an accuracy of over 0.9 while ‘6’ and ‘9’ have a relatively low accuracy but still higher than 0.8. The average of all the digit, as well as the practical accuracy, is 0.933 which means for every 50 inputs, a user may be

Table 4.1 The recognition rate of individual digits from 0 to 9 by PDC

Digit	0	1	2	3	4	5	6	7	8	9	average
Accuracy	0.98	0.9	0.99	0.97	0.91	0.95	0.87	0.99	0.96	0.81	0.933

rejected 4 times incorrectly caused by the misrecognition of digits.

4.3 The Performance of Password Handwriting Checker

In section 4.1.1, we talked about two possible attacker models, the content-aware attack and observer attack. The performance of PHC plays a critical role to resist these two attacks since it relates to how writing behavior affect the authentication result. So, in this section, we will evaluate the success rate and error rates of PHC show how PHC works to two attacks talked above by using the dataset collected in the first experiment.

Since the samples number we collected in the first experiment is relatively small, we have done a cross-validation test. For each participant, we will randomly choose 3 valid samples and use it for selecting a reference template in the setting phase and the rest of samples includes the valid samples and imitation samples will be used for testing in each test, which highly enlarge the number of tests and provide more accurate results. Through this method, we can done $3 \times 5 \times 6 = 90$ valid authentication tests to calculate the successful rate that successfully verify the valid user, and $3 \times 6 \times 4 \times 5 = 360$ times for each attack test to get the error rate that wrongly recognize the attack sample as the valid sample.

The results are presented in Table 4.2. For all the password we provided to the participant, the average success rate of the valid test is 0.92. This success rate means that although the PINWrite system still has some misjudgment may cause by the change of users handwriting or some other influences, it can verify a valid user in most cases.

In terms of success rate of attacks, which is 0.09 to content-aware attack and 0.15 to observer attack. We can say that even the attackers know your password and how you write it, our system can still reject 91% and 85% of the malicious input, giving much privacy to the basic password authentication that cannot assist this attack at all.

Table 4.2 The evaluation results of the Password Handwriting Checker to recognize the valid user and attacks

	valid test	content-aware attack	observer attack
successful rate	0.92	0.09	0.15

This result proved that handwriting can be effectual for identifying a valid user and PHC is able to provide a basis to PINWrite system to have a reasonable performance.

4.4 Overall Performance of PINWrite System

In the previous section, we have analyzed the performance of components of the PINWrite system, the PDC (Password Digit Checker) and PHC(Password Handwriting Checker), separately. However, when a real user wants to unlock his/her smartphone by using this application, he/she should pass both PDC and PHC. Therefore, in this section, we analyze the overall performance of PINWrite system with the dataset collected from second experiment in section 4.1.2.

4.4.1 Evaluation Metrics

Here we adopted standard ROC curves and precision-recall curves to represent the performance of PINWrite system. Receiver operating characteristic (ROC) curve [23] is a fundamental tool to illustrate the diagnostic ability of a binary classifier system, which is consisted of the plots of true positive rate (TPR) over the false positive rate (FPR) at various discrimination threshold. An ideal system would get 100% TPR while the FPR is 0%, means that all the valid user could through the verification meanwhile zero attackers could pass the system successfully.

$$TPR = \frac{TP}{TP + FN} \tag{4.1}$$

$$FPR = \frac{TN}{TN + FP} \tag{4.2}$$

Table 4.3 The different fractions (TP, FP, TN, FN)

	p ’ (predicted)	n ’ (predicted)
p(actual)	True Positive	False Negative
n(actual)	False Positive	True Negative

To every possible threshold, there are four cases that might appear from a binary classifier. If the outcome is predicted as p and the actual value is also p, correctly classified as positive, will be called as true positive (TP), but if the actual value is n, a negative sample is classified as positive incorrect, will be said as false positive (FP). Adversely, a true negative (TN) will appear when both predicted and actual value is n. However, if the actual value is p while the predicted value is n, will be classified as false negative (FN).

Based on the formula of TPR and FPR, we can assume that in ROC curve when selecting a higher threshold value, the FPR will decrease while the TPR will also decrease, if the threshold value achieves max value, TPR=FPR=0. Meanwhile, if we select a lower criterion value, then the TPR will increase and the FPR will also increase, when the threshold is minimum, TPR=FPR=1.

As for the precision-recall(PR) curves [24], precision-recall curve is also a useful measure of the success of prediction, shows the relationship between precision and recall under the different threshold, which could work as a supplement to the ROC curves to get the full picture when evaluating and comparing tests.

- Precision is a measure of result relevancy, calculated by the percentage of valid samples out of all the samples that passed the verification, shows how cautious the system is to the trust a user. An ideal system needs to have 100% precision and only allow valid samples to pass the system.

$$Precision = \frac{TP}{TP + FP} \tag{4.3}$$

- Recall represents the degree of true positive in the returned result, shows the ratio of valid samples that have been accepted by the system out of all the valid samples. The recall will influence the user experience when recall achieves to 100%, a valid user can always through the verification of the system in his/her first attempt. A recall is 50% represents a valid user would have a half probability to pass the verification, which means a valid user need Two attempts to be accepted by system on average.

$$Recall = \frac{TP}{TP + FN} \tag{4.4}$$

By setting different thresholds, we can achieve varied precision, recall, TPR, FPR values, and draw ROC curves and PR curves.

4.4.2 Evaluation Result

To get the ROC and precision-recall curves, we need two kinds of samples, the positive samples and negative samples mentioned above. In the database collecting from the second experiment, for every participant, we got 3 samples for training and 3 for testing. Here we set the testing samples as the positive samples, and randomly choose samples from other participants that contain the same digit with the password set by this participant as the negative. Thus, the positive and negative group both contain 90 samples.

After testing all the samples by the PINWrite system, we have got a list of verification scores of both positive and negative scores. Table 4.4 shows a part of this list.

Table 4.4 The verification score of both postive and negative samples.

label	score
0	0.7368
1	0.7361
0	0.7337
1	0.7332
0	0.7331
1	0.7322
0	0.7253
1	0.7242
0	0
0	0.7158

The label ‘0’and ‘1’ represent the negative and positive sample, and the score is the similarity score calculated by PHC, so ‘0’ means that this sample didn’t pass the verification of PDC.

By using these data and set the score to different thresholds, we can get different TPR and FPR as Table 4.5, and get the ROC curve and PR curve of the PINWrite system shown in Fig 4.3.

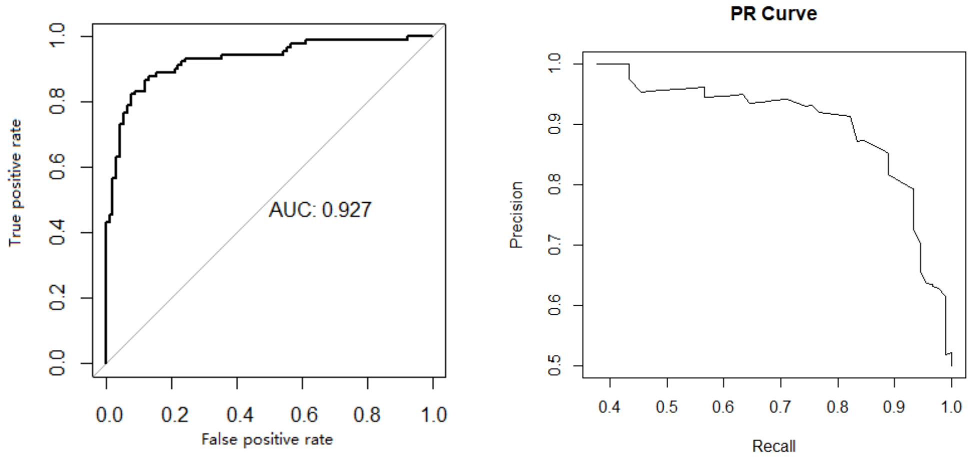
In ROC curve, each point represents a true/false positive rate corresponding to a particular threshold of the score. If a system can perfectly discriminate between the positive and negative groups, it ’ s ROC curve will pass through the upper left corner since it can get the result of 100%TPR and 0% FPR. Thus, the closer ROC curve reaches to the upper left corner, the system has a higher overall accuracy to verify the positive sample. Apart from the curve, we also calculate the area under the ROC curve(AUC), which could be regarded as the possibility that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one [25]. Therefore, when a sample from two groups can ’ t be classified by the classifier, where there is no difference between two groups, the AUC will be 0.5. On the other hand, if this system could separate the scores of two groups perfectly, which means there is no overlapping of the distributions, the AUC will 1. Thus, the closer AUC to 1, shows system has better ability to classify the positive and negative samples. Here, we calculate the AUC in the PINWrite system is 0.927. Based on these two

Table 4.5 Criterion values and coordinates of the ROC curve

Criterion	TPR	FPR
> 0	100	100
>0.4728	100	91.11
>0.6219	96.67	55.56
>0.7165	93.33	23.33
>0.7651	87.78	14.44
>0.7863	83.33	11.01
>0.7997	81.11	8.89
>0.813	75.56	5.56
>0.8714	62.22	3.33
>0.9197	55.56	2.22
>0.9877	38.89	1.11
>1.2768	0	0

features shown in ROC curves, we can assume that PINWrite system can verify valid users in most cases. Meanwhile, after drawing the ROC curve, we also find the best threshold of similarity score, which is 0.8354, resulting 72.22% TPR and 4.44%FPR. This threshold can allow most of the valid sample pass through the authentication while blocking attacks at the greatest extent.

The PR curve stands for the balance between precision and recall under different thresholds. When a system has both high precision and high recall, the AUC would be quite big. These high scores show that the system is able to return accurate results while most of them are from positive samples, since high precision relates to low FPR, that allow less negative samples pass the verification, and high recall relates to low false negative rate, that will misrecognize less negative sample as positive. When a system has high precision with a low recall, the system will return very few



(a) ROC curve

(b) PR curve

Fig. 4.3 The ROC curve and PR curve of PINWrite system.

results predicted as positive, but most of its predicted label is the same as actual. In contrast, if the system has a high recall but low precision, it will return many results predicted as positive, while most predicted labels are incorrect. Hence, an ideal system needs to have both high precision and high recall that will return mainly positive results and most of the results are predicted correctly. By setting the threshold as 0.8354, the system's recall is 0.72 while the precision is 0.94, illustrating that our PINWrite system could let a majority of positive sample pass the verification in a high accuracy.

4.5 Time Consumption

Apart from the precision rate, another important part when testing a system usability is time-consuming. The verification time of PINWrite system includes two parts, digits writing time and the system running time of PDC and PHC components during the unlock section. Since PDC and PHC components will not take much time to recognize digit and verify handwriting, most of the time will be used by the writing behavior of users in whole verification time. Thus, in the user study, we collected the time parameters to show the hold-up time of users when writing a digit and the whole 4-digit PIN when unlocking the system.

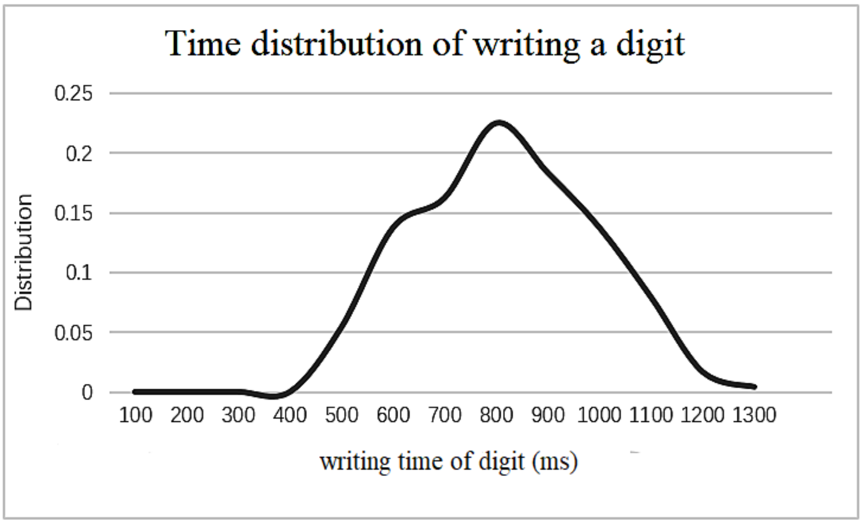


Fig. 4.4 The time distribution of writing a digit.

Fig 4.4 shows the distribution time of 30 users when writing digits on the screen in the second experiment. From the distribution curve, we found that most of participants write one digit from 600 milliseconds to 1 second, only little of them write a digit longer than 1s or less than 600ms. As for the whole 4-digit PIN, the time distribution is shown in Fig 4.5 , the average time consuming of writing a 4-digit PIN is 7.86 seconds. Comparing to the time of writing a single digit, it seems 4-digit PIN will cause user much more time. By observing the experiment process, the rise in writing time may cause by some intervals between each digit that user will take some break after writing one digit. Besides, the participants may also be influenced by other factors like rewrite a digit or talk to others while writing, thereby causing a long writing time when unlocking the system.

From the previous study, we found that for the simple numeric PIN method, it will take average 4.7 seconds to unlock the system. [26] It is apparent that our PINWrite system would take much more time than the simple PIN. In order to understand it impacts on usability, we also ask some of the participants after experiment about their thoughts with time-consuming. The feedback contains mainly two views, a small percentage of participants thought this method takes up too much time

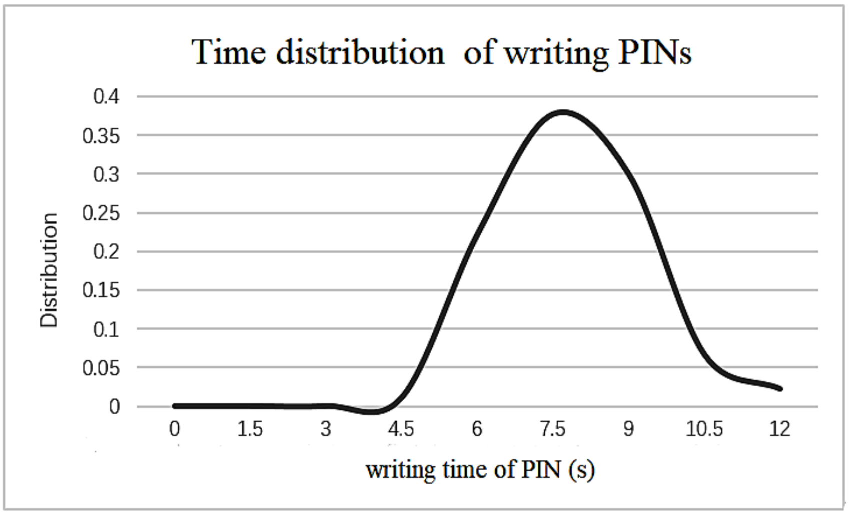


Fig. 4.5 The time distribution of writing the 4-digit PIN.

on the writing behavior while most participants say they didn't notice that it will take up much time when writing the digit and don't mind a little growth of time.

4.6 Usability Investigation

After the second experiment, we sent a questionnaire to 20 participants who had a real try of this PINWrite system to have an investigation of the usability. The demographics are summarized in Table 4.6. Due to limitation of experiment site, all the participants are students and most are IT students.

Table 4.6 The participant demographics of investigation

N	20	
Age	20-30 years	
Gender	50%	Female
	50%	Male
Occupation	100%	student
IT Experience	65%	have worked in or studied IT

In the questionnaire, we designed the questions based on the System Usability Scale (SUS) tool to collect their thoughts with the usability of the PINWrite system. The SUS was originally created by John Brooke in 1986, which help to evaluate a wide variety of products and services. [27] It consists of 10 questions with five response options: from Strongly agree to Strongly disagree. Here we adjust these questions as follows to fit this research :

1. I thought the system was easy to use
2. I think that I would like to use this system frequently
3. I found the system very cumbersome to use
4. I think that I would need the support of a technical person to be able to use this system
5. I am satisfied with the application functionalities.
6. I would imagine that most people would learn to use this system very quickly

7. I felt more secure using this system to lock my devices than traditional PIN system
8. I think this system is more novel than the traditional PIN system
9. I think this system will reveal less personal information than touch-ID and Face ID.
10. I will use it instead of my current locking system using in my daily routine.

Based on users' reply, we could get the scores for each question from 0 to 4, add them together and then multiply by 2.5 to change the original score from 0-40 to 0-100. At the end, we calculate the score as 78.25 that is higher than the average SUS score which is 68 [28]. This result shows that based on the questionnaire, the PINWrite system has a good usability to some extent while this system still needs the effort to improve its practicability.

Chapter 5 Related Work

In this section, we briefly review some approaches that come up as alternatives to password-based authentication which is related to our work.

To make authentication of smartphone resilient to shoulder suffering attack, Krombholz et al. [29] proposed a force-PIN method that adds the use of the pressure-sensitive touchscreen to the digit-only PIN. By using the pressure-sensitive component, users can select a higher entropy PIN which is not easy for shoulder surfer to observe. However, user sometime may not find the exact pressure when hey press the screen in a hurry which makes them easy to be rejected. To use this method, it also has some requirement to the smartphone, since this method will use pressure as a factor to authenticate. It is important that the smartphone needs to carry a pressure-sensitive touchscreen that many smartphones do not meet these conditions.

There are some methods also used the behavioral metrics as factors to do the authentication. Tian, et al. [30] proposed a behavior-based authentication that asks users to sign their password in 3D space and captures the motion using the Kinect which is a low-cost motion input sensing device. Unfortunately, this method does not work on the smartphone device and difficult for the smartphone to use since it needs the specific device. And Emanuel, et al. [31] present an authentication system named SwiPIN that allows users to input the traditional PIN by using some simple gesture like up and down which can against the human observers. But for the user, they need to learn a new mechanism that is harder to remember than a digit-only PIN, leading a decline in the usability.

Alexander, et al. [32]present a gesture-based authentication that use both the front and back of the smartphone to let users enter a stroke-based password. Users can make a use of the back screen to minimize the risk of shoulder surfing attack. However, at present, most smartphones does not have a back screen, so in order to use this method, they need to add a specific device to the smartphone, making the smartphone heavier to use.

Chapter 6 Summary

6.1 Conclusion

In this paper, we proposed a new authentication mechanism that integrated the writing behavior into the knowledge-based authentication. This PINWrite system strengthens the digit-only system with users' unique handwriting. The additional handwriting segment allows users to have an individualized password, which is harder for attackers to observe.

In terms of security and usability, we conducted two user studies to collect the data and use these data to analyze the performance of each component and the overall PINWrite system. To analyze the security side, we first collect the data of both valid users and attackers. The mainly work is finding how this system defends different attack models. The result shows that the password handwriting checker(PHC) in PINWrite system achieved a success rate of 92% to verify the valid user correctly. This indicates that the system works for most users in most instances.

According to the data of the attack sample, we found that PHC could reject 91% of the attacks that the attacker already knew the password of the user. It showed that our system can solve the attacks that users' password is known by others to some extent. In the scenario of the observer attack that attacker may have a chance to see how the digit is written by the user and train themselves to imitate the handwriting of user, PHC could still resist 85% of the attacks. This result indicates that the PHC in PINWrite system increased resistances of shoulder surfing attack for the knowledge-based authentication.

As for the overall performance of the whole system, we invite another 30 participants and collect their setting and unlocking data while using this system. Based on the collecting data, we created a list of similarity scores to draw both ROC and PR curves that can describe the system's performance. From the ROC and PR curves, we first calculate the AUC as 0.927 which is close to 1, indicating the system has good ability to classify the positive and negative samples. Then we found the best threshold of the system as 0.8354. Under this threshold the system could distinguish the positivist and negative in the most efficient way, resulting the system recall is 0.72 and precision is 0.94.

In order to analyze the usability side, we also collect the time data and hand out a questionnaire to participants who had a real use of the system. In time consumption, we found that by using the PINWrite system, users would cause 7.86 seconds per one unlocking period, which is much longer than the traditional PIN method. Thus, based on the problem, we also asked the participants' opinion. Although some of them thought the system take to much time, most of the reply tells the users don't mind the time increase. Besides, we also hand out a questionnaire, containing 10 questions that are modified from the SUS tool to 20 participants. In this questionnaire, we asked the participants' opinion towards to the practicability and safety. At least we calculate a usability score as 78.25 which is higher than the average SUS score, indicating that the system has good usability to some extent.

6.2 limitations and Solutions

However, this PINWrite system and our experiment still have some limitations. As we talked in section 4.2, user tends to choose digit that is easy to write like ‘0’,‘1’,‘2’ when using this system, which will lead a result that their password can be guess and imitation easier than other complex digits. Besides, in our experiment, we also meet some limitations. One thing is that we only used one type of the equipment when collecting the data form user, ignoring the influence of screen size and resolution of different smartphones to the final result. Another insufficient is that when collecting the attacks’ data, we only ask the attackers to look at the images of valid inputs instead of observing the real movement of the valid user directly, which is not the same as what would be happened in the real world. So, our experiment still has many points to be improved.

To eliminate these limitations, we may employ some countermeasures. For example, in the setting phase, when system asks users to select a 4-digit password, we can add a prompt on the screen to tell them whether the password is safe enough and suggest them to select a relatively complicated password to enhance the security. As for the experiment, to consider the influence of different types of equipment, we may ask users to use their own equipment if they use the android smartphone instead of just using the equipment from lab to collect the user’s data in the later experiment to take account of the variety of equipment. For another insufficient, since it will cost a lot of time and effort to ask the attacker to stand behind a real valid user, we may take videos of the valid user’s input and thus we can show the movement of writing to attackers to simulate the real observe attack.

6.3 Future Works

In the future work, we raised some potential update to the existing PINWrite system:

Now, in the PINWrite system, when user writing a digit, they can get the feedback, the trail on screen, to remind them which digit they are writing and confirm whether the digit is correct or not. But with a visible trail, the written digit is more vulnerable since the attacker could also see the trail if they are close enough. Thus, to improve the capacity of defending the observer attack, we could use the invisible trail while writing a digit in this system, which is much harder for attackers to observe the digit drawn by users.

On the other hand, instead of making adjustments to the writing behavior, we could also do some upgrade to enlarge the password space. For example, the system could add some other characters like English letters or even the letters from users’ country. By using these characters, users would have more choice in their password instead of just using digit. They could also combine different kinds of characters together, making their password complex enough to defend all kinds of attacks and thus improving the safety.

Acknowledgement

I would first express my sincere heartfelt thanks to my supervisor, Professor Tatsuya Mori, for his invaluable advice and constant encouragement.Then I would also like to express my gratitude to the lab members because they provide a lot of help to my experiment and also give me many valuable comments during my study.

Bibliography

[1] Y. LeCun, C. Cortes, and C.J Burges. Mnist handwritten digit database. (online). http://yann. lecun. com/exdb/mnist, 2010.

[2] Deep learning and convolutional neural networks: Rsip vision blogs. (online). http://www.rsipvision.com/exploring-deep-learning/, 2015.

[3] R. Lee and P. Andrew. 10 facts about smartphones as the iphone turns 10(online). http://www.pewresearch.org/fact-tank/2017/06/28/10-facts-about-smartphones, 2017.

[4] I.T. Fischer, C. Kuo, L. Huang, and M. Frank. Short paper: Smartphones: Not smart enough? In *In Proceedings of the second ACM workshop on Security and privacy in smartphones and mobile devices*, pp. 27–32. ACM, 2012.

[5] M. Anderson and Olmstead K. Many smartphone owners don ’ t take steps to secure their devices.(online). http://www.pewresearch.org/fact-tank/2017/03/15/many-smartphone-owners-dont-take-steps-to-secure-their-devices/, 2017.

[6] F. Schaub, R. Deyhle, and M. Weber. Password entry usability and shoulder surfing susceptibility on different smartphone platforms. In *Proc. MUM*. ACM, 2012.

[7] Adam J. et al Aviv. "smudge attacks on smartphone touch screens.". In *Woot 10*, 2010.

[8] L. Coventry, A. De Angeli, and G. Johnson. Usability and biometric verification at the atm interface. Proc. CHI ’ 03. ACM, 2003.

[9] C. Bhagavatula, K. Iacovino, S. M. Kywe, L. F. Cranor, and B. Ur. Usability analysis of biometric authentication systems on mobile phones. In *Proc.* SOUPS, 2014.

[10] Cao K and Jain AK. Hacking mobile phones using 2d printed fingerprints. 2016.

[11] Shafique U, Khan H, Sher A, Zeb A, Shafi U, Ullah R, Bashir F, and Shah MA. Modern authentication techniques in smart phones: Security and usability perspective. In *INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS*, 2017.

[12] Antoniou D and Socha K. "authentication methods". In *Security Whitepaper 16-003,CERT-EU.*, 2016.

[13] Messerman A, Mustafic T, Camtepe S A, and et al. A generic framework and runtime environment for development and evaluation of behavioral biometrics solutions[c]. Intelligent Systems Design and Applications (ISDA), 2010.

[14] Reynolds and Douglas A. "an overview of automatic speaker recognition technology."acoustics, speech, and signal processing (icassp). IEEE, 2002.

[15] Sun, Zhenan, Bangyu L, and Tieniu T. "statistical features for text-independent writer identification." behavioral biometrics for human identification: Intelligent applications. 2009.

[16] Matan O, Baird HS, Bromley J, Burges CJ, Denker JS, Jackel LD, Le Cun Y, Pednault EP, Satterfield WD, Stenard CE, and Thompson TJ. Reading handwritten digits: A zip code recognition system. Computer, 1992.

[17] Michael A. Nielsen. Neural networks and deep learning", determination press. 2015.

[18] Rublee, Ethan, and et al. "orb: An efficient alternative to sift or surf.". In *Computer Vision (ICCV)*. IEEE, 2011.

[19] Harris, Chris, and Mike Stephens. "a combined corner and edge detector.". In *Alvey vision*

conference. Vol. 15. No. 50., 1988.

[20] P. L. Rosin. Measuring corner properties. computer vision and image understanding. 1999.

[21] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model ïňĄtting with applications to image analysis and automated cartography. In *Communications of the ACM*, pp. 381–395, 1981.

[22] Guangjun Shi, Xiangyang Xu, and Yaping Dai. "sift feature point matching based on improved ransac algorithm.". In *Intelligent human-machine systems and cybernetics (IHMSC)*. IEEE, 2013.

[23] "detector performance analysis using roc curves". www.mathworks.com, 2016. MATLAB.

[24] Saito T and Rehmsmeier M. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. PLoS One, 2015.

[25] Fawcett and Tom. An introduction to roc analysis, pattern recognition letters. pp. 861–874, 2006.

[26] Harbach and Marian et al. "it ' s a hard lock life: A field study of smartphone (un) locking behavior and risk perception.". In *Symposium on usable privacy and security*. 2014, SOUPS.

[27] John Brooke. "sus-a quick and dirty usability scale.". In *Usability evaluation in industry*, pp. 189–194, 1996.

[28] Jeff. Sauro. "measuring usability with the system usability scale (sus). http://www. measuringusability. com/sus.php, 2015.

[29] Krombholz, Katharina, Thomas Hupperich, and Thorsten Holz. "use the force: Evaluating force-sensitive authentication for mobile devices.". In *Twelfth Symposium on Usable Privacy and Security*. USENIX Association, 2016.

[30] Jing Tian and et al. "kinwrite: Handwriting-based authentication using kinect.". NDSS, 2013.

[31] Von Zezschwitz, Emanuel, and et al. "swipin: Fast and secure pin-entry on smartphones.". In *33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2015.

[32] De Luca, Alexander, and et al. "now you see me, now you don't: protecting smartphone authentication from shoulder surfers.". In *SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2014.