Mobile Fingerprint Authentication System with Enhanced Security

Li, Yi

(出版者 / Publisher)
法政大学大学院情報科学研究科

(雑誌名 / Journal or Publication Title)
法政大学大学院紀要. 情報科学研究科編 / 法政大学大学院紀要. 情報科学研究科編

(巻 / Volume)
11

(開始ページ / Start Page)
1

(終了ページ / End Page)
6

(発行年 / Year)
2016-03-24

(URL)
https://doi.org/10.15002/00012875

Mobile Fingerprint Authentication System with Enhanced Security

Yi Li
Graduate School of Computer and Information Sciences
Hosei University
Tokyo 184-8454, Japan
yi.li.7r@stu.hosei.ac.jp

***Abstract*—Although use of fingerprint is highly effective in user authentication of networked services such as electronic payment, there are some problems in conventional systems, including high cost due to need for specialized fingerprint readers and limited usability. To resolve these problems, we propose a new system which incorporates a web-based fingerprint authentication using a smartphone as a fingerprint input device. Additionally, a watermark-based encryption solution is used to enhance system security. With this solution, the system can prevent information interception and replay attack. We demonstrate through prototype implementation and experiments that our solution enhances security of web-based system, and the cost and usability problems in conventional systems can also be resolved.**

Keywords*—*fingerprint authentication; watermark; encryption; web security

I. Introduction

A. Fingerprint-based E-payment User Authentication System

Fingerprint identification system is an important identity authentication system. It will be more and more important in modern society. Traditional fields such as security system, access control system must use this technology. With the rise of the mobile Internet, more and more emerging fields such as e-commerce, electronic payment also need this technology.

The traditional fingerprint authentication system adopts the following technical architecture:

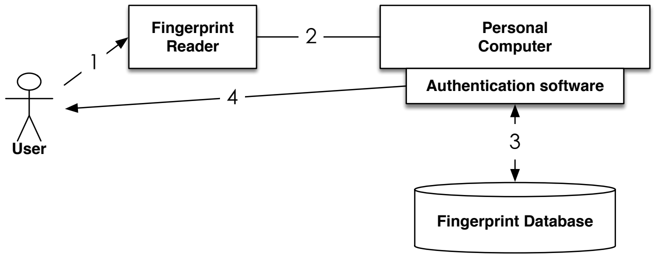

Fig. 1. Traditional fingerprint authentication solution

The transaction process is as follows:

1) User presses the fingerprints on the fingerprints sensor.
2) The sensor is connected to the computer system and sends the user's fingerprint directly to the computer system.
3) The similarity of fingerprint data between user and database will be compared.
4) The fingerprint authentication system judges whether the user is authenticated by the similarity and returns the result to the user.

However, with the rise of mobile Internet, the drawbacks of this traditional fingerprint authentication system were exposed. First is cost issues, once scenarios need to use this system, we must purchase the corresponding hardware and software, leading to high cost. Second is usability issues, the traditional fingerprint system requires sensor to input users fingerprints and the system deployment requires a separate terminal hardware and terminal software. Without the specialized hardware, user cannot use this system for payment. The usability is not good enough.

In this paper, the drawbacks of traditional fingerprint authentication system have been improved and reorganized. We proposed a web-based fingerprint authentication solution. Finally, the cost and usability issues in traditional fingerprint authentication system can be solved by using the proposed solution.

System architecture is as follows:

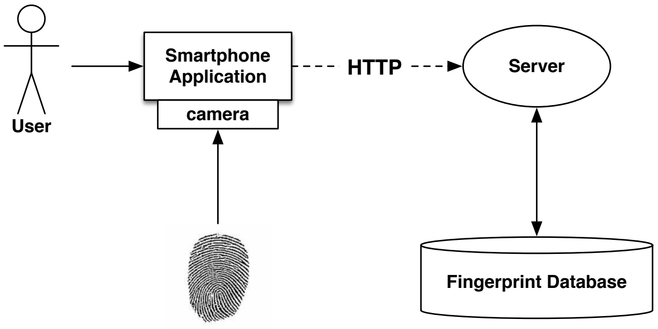

Fig. 2. New fingerprint authentication solution

The innovation of the system is:

- Using smartphone camera instead of the traditional fingerprint sensor, to solve the cost and usability issues.
- Using Web Service to provide cross-platform service to solve the cost issues.

The proposed solution has resolved cost and usability problems, but the security problem occurs in the same time. Because the user information is transferred by http protocol, the web attackers can listen to the internet and intercept user's information. Some kinds of security problems occur such as information interception, replay attack, etc.. If we design such a system that could incur multiple security problems, I will first analyze the security aspects.

B. Fingerprint Authentication System with Enhanced Security

The paper discusses a fingerprint authentication system with enhanced security. First, we will propose a new solution instead of the traditional solution, we will not use fingerprint sensor to get user fingerprint data. Instead, we use smartphone camera to get the user information. It will be more convenient and low cost. Second, we use Web Service to provide web-based authentication service with enhanced security. We use some methods to improve the system security and prevent network attack. Finally, we will develop a system prototype with convenience, security and low cost.

II. Related work

This part will give some current research. Because our research is comprehensive research. This parts will contain three or more kinds of research.

Chris Stein, Claudia Nickel and Christoph Busch [1] proposed a fingerphoto recognition solution with smartphone cameras. They developed a prototype using Android phone and realized touch-less fingerphoto recognition. But they didn't use web service and they didn't take security issues into account.

Waiton [2] proposed Least Significant Bit Algorithm to realize a reversible watermark. He used the least significant bit to store a key and took the other bits to generate the key. This algorithm is simple to generate a reversible watermark and easy to use.

A.Z.Tirkel, R.G.van Schyndel, C.F.Osborne [3] proposed a two-dimensional digital watermark using least significant bit algorithm and discussed compatibility of the technique with JPEG image transmission.

Zhang Ning, Zang Ya-Li, Tian Jie [4] gave a new solution for secure identity authentication by integration of biometrics and cryptography. They discussed fingerprint and security key and gave a idea to combine biometrics with cryptography.

The fingerprint group [5] at nist(National Institute of Standards and Technology) developed an open source software called NBIS. The software can be used for fingerprint feature extraction and matching. SourceAFIS is another open source software for fingerprint recognition.

Google corporation [6] developed an open source framework called Zxing. Using this framework, programmers can generate barcode easily in their system. Eui-Hyun Jung and Seong-Yun Cho [7] researched barcode technology and watermark technology and then proposed a watermark solution using 2D barcode technology.

In this paper, We will do a comprehensive research and incorporate their research findings into our prototype.

III. Proposed Solution

This chapter discuss the security strategy in the fingerprint authentication system. First, we will analyze security threat briefly; Second, we discuss the traditional approach to enhance security; Third, we propose a fingerprint watermark approach to solve the security problems. We will discuss reversible watermarking technology in this part.

A. Conventional Approach

1) Security Threat: There are some types of internet attack. One is to intercept information. The purpose of intercepting information is to steal data content itself. Another one is to replay attack [8]. Replay attack is a form of network attack in which a valid data transmission is maliciously or fraudulently repeated or delayed.

2) Common Solution: File encryption and digital signature is a common approach to enhance system security.

- File encryption : Prevent information leaks.
- Digital signature : Ensure the message sent by the original sender.

The conventional approach is good enough to enhance system security and prevent internet attack. However, there are still some disadvantages to apply it to our fingerprint system.

- Complex: Digital-signature need to be controlled in a complex architecture. Besides, we need to build two separate subsystems, one is for file encryption, the other one is for digital signature. It will be more complex.
- Generality: It is a common approach to enhance system security. It does not take into account the characteristics of the fingerprint image.

B. Proposed Solution

To overcome the disadvantages of conventional approach, a proposed solution will be presented in this part. First, we analyze what problems will emerge without conventional approach. Then we will improve the original solution step by step.

1) Problems: If we do not use the technology of File encryption and Digital signature, we will encounter security holes in our system first. In this case, User A wants to use this system for authentication, he uses smartphone to take his fingerprint photo, then A sends it to the server for matching directly. However, the attacker B was listening to this communication, he is a malicious attacker and intercepts user A's fingerprint image. After few minutes, the attacker B was posing A to send the fingerprint image to the server. Thus the system will take B as A, the attacker B will obtain system authentication. Now user A is unsafe, because B gets the same system authority as A. Our system cannot prevent interception and replay attack, we should improve the system with some security arrangement.

2) Improvement 1: To enhance security of the system, we import the security module.

- To prevent information leaks, the client should not send the original fingerprint picture to the server.
- To prevent replay attack, the system can use a technology called one-time pad [9]. It means that the key is valid only once. To achieve one-time pad system, we can take timestamp as a key. When user wants to authenticate the fingerprint, server-side checks the timestamp first, if the

timestamp key is right, then the system do the matching. Otherwise, the system will not match the fingerprint with database.

In this case, user A take photos first, then he sends the request to server. The server takes current timestamp to generate a barcode picture and sends it back to client. After client receives the barcode picture, it sets the original fingerprint picture to the center of the barcode picture. Then the new picture will be sent to the server. The server receives the new picture and decodes the barcode to check whether timestamp is correct. If it is the same as server-side timestamp, the system will send the picture to matching module. Otherwise, the server will return fail information to client.

The security of our system has been improved. However, it's still not safe enough. The attackers have some methods to get the original fingerprint picture easily. They can cut down the center of the picture then they will get the original fingerprint picture. Then they can encode qrcode with original fingerprint image, it is a kind of replay attack.

3) Improvement 2: The basic idea will be not changed. The problem of the solution described above is that it just combines fingerprint picture with barcode picture directly. The fingerprint image is not hidden and is easy to recover. So if we can use an algorithm to hide the fingerprint image, it will be hard to reverse. Then the system will be safer.

A technology called reversible digital watermark will be applied in our system. Using this technology, we could combine fingerprint image with barcode image easily and the fingerprint image will be hidden at the same time. The fingerprint picture will be as watermark and it will not be sent to the server directly.

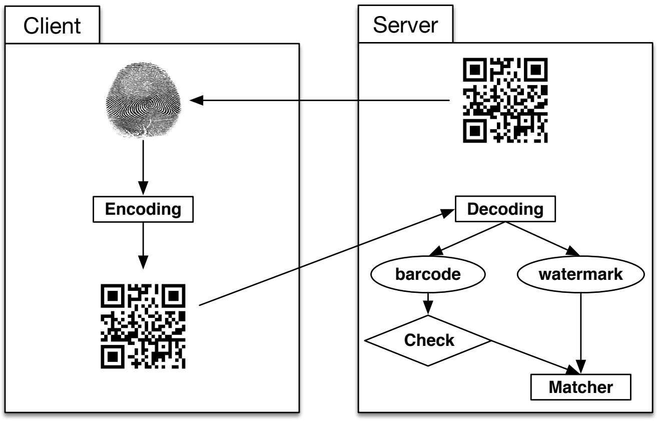

Fig. 3. A watermark-based solution

We just use watermark technology in this solution. The client takes fingerprint picture as watermark and encodes it into the barcode picture. Now the system is safer than before. The attackers will not get the original fingerprint picture easily. They can only get qrcode picture while that picture is not useful for authentication. If attackers attempt to recover the original fingerprint picture, they must know the algorithm first. However, if the attackers get the algorithm of encoding, the system is still unsafe. So we have to improve the system more.

4) Improvement 3: Finally, we add a key to the system. We take user's password as a key to encode the above picture, then client sends the encoded picture to the server. Server-side receives the picture and decodes it using the same key. After that, the server can do the same operation as the former part.

In this way, the attackers must get both key and algorithm to recover the original fingerprint picture. The cost is very high. So we think the system is safe enough.

IV. System Architecture

This chapter will introduce system flow and the choice of architecture. This part will be divided to two parts. Part One introduces the specification of the hole system; Part Two discusses system framework and how it works.

A. Basic Process

The basic flow of the system is shown below:

1) Fingerprint Image Acquisition: User login android client, then the client calls camera and image acquisition module to get user fingerprint image.
2) Image Preprocessing and Feature Extraction: The client calls feature extraction module to preprocess the fingerprint image and extract the feature of users fingerprint.
3) Image Encryption: First the client receives the QR Code information from server, then it calls encryption module to encode fingerprint image with QR Code information.
4) Image Upload: The client calls Web API to upload the encrypted fingerprint images to the server.
5) Image Decryption: The server calls decryption module to decode the encrypted fingerprint image, then it will get original fingerprint image and QR Code information.
6) User Authentication: The server calls authentication module to check the QR Code information and ensure the authenticity of the user.
7) Fingerprint Matching: The server calls Fingerprint Matcher SDK to compare user-uploaded fingerprint image with the fingerprint database, then it will get the matching score.
8) Result Display: The server sends the matching score back to the client. Then the client determines whether the user is authenticated by matching score.

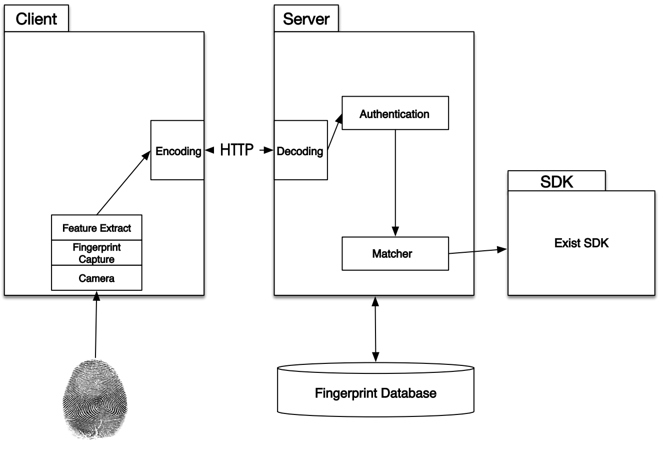

Fig. 4. Basic process flow

B. System Framework

We use Nancy Framework to achieve a lightweight REST-style Web Service. Nancy Framework is a lightweight web framework based on .Net and mono platforms, it follows the MVC model and is designed to provide REST-style Web Service. On its official website [10], it introduces the following features of Nancy:

- Nancy is a lightweight, low-ceremony, framework for building HTTP based services on .Net and Mono. The goal of the framework is to stay out of the way as much as possible and provide a super-duper-happy-path to all interactions.
- Nancy is designed to handle DELETE, GET, HEAD, OPTIONS, POST, PUT and PATCH requests and provides a simple, elegant, Domain Specific Language (DSL) for returning a response with just a couple of keystrokes.
- Nancy is built to run anywhere.

The design ideas of Nancy Framework comes from Ruby's Sinatra Framework, whose basic idea is MVC model. MVC model is Model-View-Controller mode.

- Model is a part of application system to deal with data logic, typically used to encapsulate data objects to store data.
- View is a part of application system to display data to user. View data usually need to be obtained from model.
- Controller is responsible for reading data from the view, controlling user input, and sending data to the model.

The architecture diagram of the fingerprint identification system is as follows:

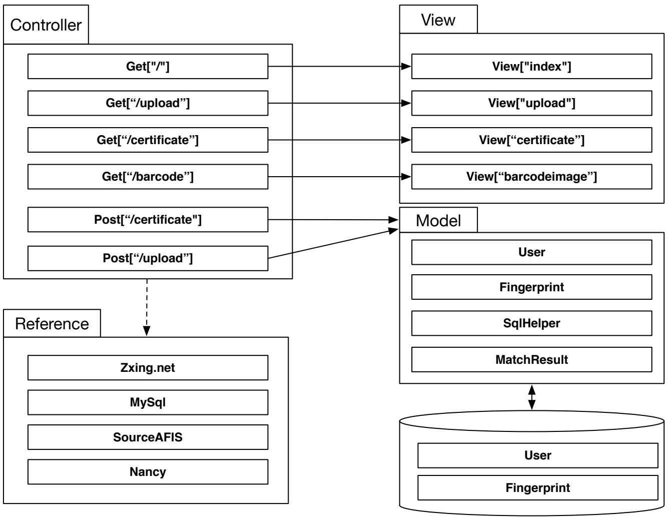

Fig. 5. System framework

V. Watermark Algorithm

This chapter will discuss the main algorithm of our system. As we have talked in section III B. (2), we used reversible watermark technology to incorporate barcode picture with fingerprint picture. The watermark algorithm will be discussed in this part. The algorithm is based on Least Significant Bit algorithm.

A. Display Matrix

Suppose the QR Code image matrix is

$$Q = \begin{bmatrix} 255 & 254 & 255 & 1 & 2 & \cdots \\ 255 & 255 & 255 & 1 & 1 & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix}$$

Its binary form is

$$Q = \begin{bmatrix} 11111111 & 11111110 & 11111111 & 00000001 & \cdots \\ 11111111 & 11111111 & 11111111 & 00000001 & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix}$$

The watermark image matrix is:

$$M = \begin{bmatrix} 167 & 63 & 15 & \cdots \\ 255 & 127 & 128 & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix}$$

Its binary form is:

$$M = \begin{bmatrix} 10100111 & 00111111 & 00001111 & \cdots \\ 11111111 & 01111111 & 10000000 & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix}$$

Now we will separate the watermark matrix to encode it to the QRcode matrix.

We will focus the watermark matrix first.

B. Watermark Matrix

For the watermark matrix above, we called it M, we can see M(1,1)=167, in binary form, M(1,1)=10100111. We separate it to 4 parts.

- Part I: the 1-2 bit
- Part II: the 3-4 bit
- Part III: the 5-6 bit
- Part IV: the 7-8 bit

Then we can get 4 matrix from 4 parts:

1.The Part I Matrix is:

$$M1 = \begin{bmatrix} 10 & 00 & 00 & \cdots \\ 11 & 01 & 10 & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 & \cdots \\ 3 & 1 & 2 & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix}$$

2.The Part II Matrix is:

$$M2 = \begin{bmatrix} 10 & 11 & 00 & \cdots \\ 11 & 11 & 00 & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix} = \begin{bmatrix} 2 & 3 & 0 & \cdots \\ 3 & 3 & 0 & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix}$$

3.The Part III Matrix is:

$$M3 = \begin{bmatrix} 01 & 11 & 11 & \cdots \\ 11 & 01 & 00 & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix} = \begin{bmatrix} 1 & 3 & 3 & \cdots \\ 3 & 1 & 0 & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix}$$

4.The Part IV Matrix is:

$$M4 = \begin{bmatrix} 11 & 11 & 11 & \cdots \\ 11 & 11 & 00 & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix} = \begin{bmatrix} 3 & 3 & 3 & \cdots \\ 3 & 3 & 0 & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix}$$

C. Encoding Method

The QRCode Image is bigger than watermark image. The rows of QRCode image is 3 times bigger than watermark image; the columns is 5 times higher than watermark image. So we can embed the watermark matrix into qrcode matrix. It is very important. In our method, the watermark matrix M is a 392*357 matrix; the qrcode matrix Q is a 900*5000 matrix.

In addition, the QRCode image and watermark image are both grayscale images.

We use Least Significant Bit Algorithm. First, we separate matrix Q to 5 parts.

$$Q = \begin{bmatrix} A & B & \\ C & D & \\ & & E \end{bmatrix} \tag{1}$$

For matrix A,B,C,D, they are all 392*357 matrix. E is the rest part of Q. Then we use A,B,C,D and M1,M2,M3,M4 to encode. We set the least 2 bits of A,B,C,D to 0. For example:

$$A = \begin{bmatrix} 11111111 & 11111110 & 11111111 & 00000001 & \cdots \\ 11111111 & 11111111 & 11111111 & 00000001 & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix}$$

set the least 2 bits to 0:

$$A1 = \begin{bmatrix} 11111100 & 11111100 & 11111100 & 00000000 & \cdots \\ 11111100 & 11111100 & 11111100 & 00000000 & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix}$$

then we use matrix M1 instead of the least 2 bits of A:

$$A2 = A1 + M1 \tag{2}$$

For B,C,D and M2,M3,M4 ,we do the some operation:

$$B2 = B1 + M2 \tag{3}$$

$$C2 = C1 + M3 \tag{4}$$

$$D2 = D1 + M4 \tag{5}$$

After that, we will get a new matrix called Q2:

$$Q2 = \begin{bmatrix} A2 & B2 & \\ C2 & D2 & \\ & & E \end{bmatrix} \tag{6}$$

The matrix Q2 contains all the information of watermark images. Now we have realized Encoding Module.

D. Decoding Method

It is easy to get the watermark matrix. From Q2 we can extract A2,B2,C2,D2, then we will get M1,M2,M3,M4 by mod([A2,B2,C2,D2],4).

From M1, M2, M3, M4, we will recover watermark matrix easily.

$$M(i,j) = M4(i,j)+4\times M3(i,j)+16\times M2(i,j)+64\times M1(i,j)$$

In this way, now we have already recover the watermark matrix.

E. Image Encryption Algorithm

As we discussed in section III B. (4), we still need to encode user's password to the above picture. Using Least Significant Bit Algorithm, we can separate the least bits of every pixel from the watermark picture. Then we can set them zero, then use them to encode other information as a key. If we use this method, the least bits of watermark picture will be lost. So we should control the bit number and choose a suitable number to ensure that the encrypted picture is not changed too much. It need be tested by cutting different number of bits. In chapter VII, we will discuss the experiment result.

VI. Matching

A. Basic Concept

After the server receives the picture send from client, it will decode it and get the real fingerprint picture. Now the fingerprint picture is send to matching module. It will be matched with fingerprints stored in the database. Basically, the matching module will extract fingerprint features from fingerprint picture. Traditionally, two fingerprints have been compared using discrete features called minutiae. These features include points in a finger's friction skin where ridges end (called a ridge ending) or split (called a ridge bifurcation).

The matching engine will compare two fingerprints by minutiae. After that, it will give a matching score. If the fingerprint is matching, the match score will be high. Otherwise, the score will be zero.

B. SourceAFIS

There are some open-source fingerprint matching softwares on the internet. We tried NBIS software [11] and SourceAFIS software [12]. We also used a commercial software called UrU SDK [13] to do the experiment. Finally, we choose SourceAFIS as our matching engine. The matching algorithm is like this:

Step1: Define a match engine AFIS.

Step2: Define 2 Fingerprint Objects from fingerprint pictures.

Step3: Define 2 Person Objects and add the Fingerprint Objects to Person.

Step4: Use the method Extract of match engine to extract features.

Step5: Use the method Verify of match engine to get the match score.

Step6: The client determines whether the authentication is passed by the score.

We use two fingerprints to test the matching module.The first fingerprint is the same one as fingerprints in database, the other one is different from database. For correct case, we will get good scores for each sample, while the score will be zero if we use other fingerprint to authenticate.

VII. Results

Now we will analysis the result after applying proposed solution.

A. Cost and Usability

The new system makes the cost of fingerprint authentication lower than before while the usability is better than traditional solution.

TABLE I
COMPARISON

Items	Traditional Solution	Proposed Solution
Client Type	Fingerprint Sender	Smartphone APP
Server Type	PC Software	Web Service
Hardware Cost	Hardware Expensive	Hardware Cheap
Software Cost	High	Middle
Deploy Cost	High	Low
Portability	No	Good

From the table, the proposed solution uses smartphone instead of sender as client and takes web service as server. The cost of software and hardware is lower than traditional solution. Portability is greatly enhanced. So we think the proposed solution resolved the cost and usability problems in the traditional solution.

B. Security

The system takes reversible watermark technology to enhance the system security. As we talked in chapter IV, we choose different number of bits to control the watermark picture. First we choose to lose the least 2 bits of fingerprint picture to encode watermark. In this way, the watermark will only lose very little informationbut the length of key is limited. Then we will test 4 bits and 6 bits. In these cases, the original image will lose more information but the length of key will be longer than 2-bits.

The figure shows the barcode images generated from different conditions.

Fig. 6. Qrcode picture

However, we cannot find much difference from barcode pictures. Now we will check the recovered fingerprint pictures.

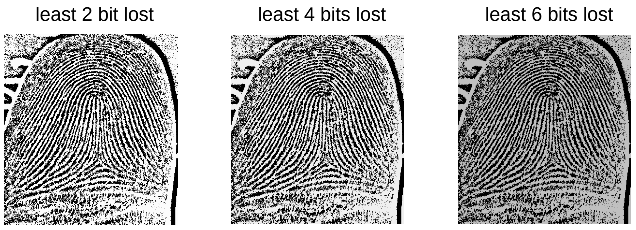

Fig. 7. Recovered fingerprint picture

We can find the the fingerprint images are a little different. The picture at right lost some information because it using least 6 bits to encode other data. However, the picture is still clear enough and not changed too much.

If we use these pictures for matching, can we pass the authentication successfully? We use the three pictures to do a basic authentication test. The result is as below:

TABLE II
RESULT

Score	2 bits lost	4 bits lost	6 bits lost
LIndex 1	33.2933044	47.68031	49.24296
Lindex 2	33.44867	37.3911247	45.64931
LIndex 3	56.163063	53.7554665	66.34745

From this table, when we use above pictures for authentication, there are little difference in matching score. It means even we use a 6 bits-lost/pixel picture for authentication, we will still pass the authentication successfully. Therefor, system can use 6 bits to encode other information to the watermark. The length of key can be much longer than 2 bits lost case.

So the system security has indeed been enhanced.

VIII. CONCLUSION

The cost and usability of traditional fingerprint authentication system is not very well. To resolve these problems, we proposed a web-based system solution. However, the security of web-based system need to be enhanced. Then we give a watermark-based solution to resolve the system security problem. Finally, the proposed solution has solved the cost and usability problems in traditional systems while the security problem in web-based system is also improved by using the reversible watermark technology.

REFERENCES

[1] Chris Stein, Claudia Nickel, Christoph Busch, “Fingerphoto Recognition with Smartphone Cameras,” International Conference of the Biometrics Special Interest Group, Gesellschaft fr Informatik e.V, 2012
[2] Waiton S., “Image authentication for a siippery new age,” Dr. Dobb s Journai, 20(4), pp.18-26, 1995
[3] Andrew Z Tirkel, Ron G van Schyndel, CF Osborne, “A two-dimensional digital watermark,” Dicta, volume 95, pp.5-8, 1995
[4] Zhang N, Zang Y L, Tian J, “The integration of biometrics and cryptographya new solution for secure identity authentication,” Journal of Cryptologic Research, 2(2), pp.159176, 2015
[5] Craig I. Watson.,Michael D. Garris.,Elham Tabassi.,Charles L. Wilson.,R. Michael McCabe.,Stanley Janet.,Kenneth Ko, “User’s Guide to Export Controlled Distribution of NIST Biometric Image Software,” National Institute of Standards and Technology, USA
[6] ZXing project: https://github.com/zxing/zxing
[7] Eui-Hyun Jung and Seong- Yun Cho, “A Robust Digital Watermarking System Adopting 2D Barcode against Digital Piracy on P2P Network,” IJCSNS International Journal of Computer Science and Network Security, vol.6, no.10, pp.263-268, 2006
[8] Replay attack: https://en.wikipedia.org/wiki/Replay_attack
[9] One-time pad: https://en.wikipedia.org/wiki/One-time_pad
[10] Nancy framework: http://nancyfx.org/
[11] NBIS Software: http://www.nist.gov/itl/iad/ig/nbis.cfm
[12] SourceAFIS Software: http://www.sourceafis.org/blog/
[13] UrU SDK: http://www.crossmatch.com/uareu-sdk-for-windows/