Implementation and Applications of a Fingerprint Encoding System

I-Fu Lin and Tzong-An Su
Dept. of Information Engineering and Computer Science
Feng Chia University
Taichung Taiwan
tasu@fcu.edu.tw

Abstract- For the fingerprint verification system, preventing from fake fingerprint attacks is a very important security issue. The fingerprint is one of the most common biometrics used for security today. Most research works focus on the living finger identification technology. But the cost of living fingerprint verification system is still higher than that of the traditional one. In our research, we utilized a sequential verification method called the sequential fingerprint verification to enhance the resistance of fake fingerprint attacks on the traditional fingerprint verification system. We can also extend this method to use fingerprints from a group of people to control the access to secret information. The experiment result showed that our method can delay the break-in from attacks of fake fingerprints. Some new applications based on our method are also discussed.

I. INTRODUCTION

The use of fingerprints in various applications has been seen for thousands of years. People use fingerprints for signatures on art works, to authenticate a document, for criminal identification and most recently, for access control. Most of these applications involve so called fingerprint verification in which input fingerprints from some person are used to match with fingerprints pre-stored in a database.

Among those applications, using fingerprints in access control to ensure the system or information security is the most popular one in recent years, for example, many PCs have been installed fingerprint reader to verify authorized users. Lots of researches have been focused on supporting this type of application. Previous research in fingerprint verification area can be divided into fingerprint recognition and quality estimation [1, 2, 3, 4, 5], security issues on fingerprint verification system [6, 7, 8, 9, 10], and implementations of commercial products and applications [12].

In this paper, we will discuss the security issues on fingerprint verification systems. Generally, a fingerprint verification system can be divided into two parts: front-end and back-end [6]. On the back-end side, the security issues could be seen as general information security issues, such as operation system security, database security and communication security. Researchers have proposed various security technologies to work with the fingerprint verification system on the back-end side to protect the fingerprint data, such as image encryption [11]. In [12], the authors distributed the fingerprint minutiae into fingerprint verification system and the smart card. This method could prevent the whole fingerprint features from stolen at the back-end side. Thus, our focus will be on the front-end side.

On the front-end side, the most important security issue is how to prevent the system from fake fingerprint replica attacks. [7] introduced different types of fingerprint readers, including optical, capacitive, ultrasonic and thermal sensors. [6] showed that different types of fingerprint readers had their own drawbacks. The authors indicated that there was 80% success rate by using fake fingerprint replicas to pass the fingerprint verification on the traditional optical fingerprint reader. Although the capacitive reader could filter off some isolative fingerprint replicas, [8] pointed out that attackers could use gelatin fingerprint replicas with moisture and resistance characteristics similar to a real human finger to fool the capacitive reader. [9, 10] showed the approach to make the fingerprint replicas and the conclusion is that it is not difficult. With all the discussion above, we should be able to realize the natural vulnerabilities of traditional fingerprint readers.

The design of traditional fingerprint reader mostly uses 1-1 or 1-N approach, i.e., one input fingerprint matches another stored fingerprint or matches among several stored fingerprints. This design feature actually adds even more vulnerability to the replica attacks. Some people suggested use multiple input fingerprints to match with stored ones in order or without order. Unfortunately, we haven’t seen any real implementation due to the reader’s simple architecture and weak capability including limited memory size.. Also, to change the “key” fingerprint normally involves a re-input of another fingerprint. It is both time-consuming and confusing.

Although the traditional fingerprint reader is subject to the issues stated above, it still enjoys high market acceptance rate. One of the reasons is due to its low manufacturing cost. The other reason is that even the high end and expensive biometrically enhanced reader can not completely eliminate the replica attack. Therefore, how to reduce or even eliminate the replica attack and add more power to the traditional fingerprint reader becomes an important work and deserves paying more attention.

II. METHODOLOGY
In our research, we design and implement a sequential (match with order) multiple fingerprint verification reader based on the following requirements:
- The number of input fingerprints used in the matching process is determined by the user.
- The set of fingerprints used as input is determined by the user.
- The order of fingerprints used in matching process is determined by the user.
- The reader should allow fingerprints from multiple persons as an input in a matching process.
- The reader should allow users to change their secret finger sequences dynamically and easily.

To fulfill these requirements, we use a mapping scheme between fingerprints and a set of codes which are symbols. We associate each registered fingerprint with a unique code, i.e., encoding all the registered fingerprints. And then users can use this encoding scheme to create his secret finger sequence. The following describes details of these two steps:

A. Fingerprint Encoding

The simplest method to encode fingerprints is to map different fingerprints to different symbols like numerals or alphabets. In Figure 1, we see that one user has enrolled his five different fingerprints in the reader system. There are two kinds of associated codes for this example in Figure 1. In the simplest case, we can associate a single digit with a fingerprint and each digit can only be associated with one fingerprint. The other case is that we can associate any character in a set of characters to a fingerprint. The restriction is that the intersection of any two such mapping sets must be empty. That is, any character in the mapping set of a fingerprint should be able to uniquely identify the specific fingerprint.

<table>
<thead>
<tr>
<th>User fingers</th>
<th>Associated codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Right thumb</td>
<td>0 a b</td>
</tr>
<tr>
<td>Right index</td>
<td>1 c d</td>
</tr>
<tr>
<td>Right middle</td>
<td>2 e f g</td>
</tr>
<tr>
<td>Right ring</td>
<td>3 h i</td>
</tr>
<tr>
<td>Right little</td>
<td>4 j k</td>
</tr>
<tr>
<td>Left thumb</td>
<td>5 l m n</td>
</tr>
<tr>
<td>Left index</td>
<td>6 o p q</td>
</tr>
<tr>
<td>Left middle</td>
<td>7 r s t</td>
</tr>
<tr>
<td>Left ring</td>
<td>8 u v w</td>
</tr>
<tr>
<td>Left little</td>
<td>9 x y z</td>
</tr>
</tbody>
</table>

Figure 1. Fingerprint Encoding of an Individual User

As an example, consider Figure 1. We can associate right thumb with digit ‘0’. On the other hand, we can associate it with the set of character ‘a’ and ‘b’. In the first case, the right thumb fingerprint could be represented as the numeral ‘0’. In the second case, it can be represented as the alphabet ‘a’ or ‘b’.

As the system requirements state, our reader should allow fingerprints from multiple persons to be registered. To handle this requirement, after multiple persons registering their fingerprints, each person should follow the method described above to establish his own mapping between his fingerprints and symbol sets. Note that users do not need to register all their ten fingerprints. Also, the mappings of two persons are not related, e.g., they even can have the same mapping although it is not recommended.

B. Creating the secret fingerprint sequence

After encoding fingerprints appropriately, users could begin to create their secret fingerprint sequence. We would divide this step into two cases based on verifications to be done on single person or on a group of persons.

1) Verifications involved only one single person: To create a secret fingerprint sequence, a user can choose some from those registered fingerprints and input to the reader the corresponding mapping codes. For the example in Figure 1, if the user inputs the code sequence as ‘1751’, the system would sequentially verify the user’s right ring, left thumb, right index and right ring fingerprints. For another example, if the user sets sequential codes as ‘lab17’ the system would sequentially verify the user’s right index, right thumb, right thumb, right ring and left thumb fingerprints. In our design, we allow users to create multiple secret fingerprint sequences to avoid situations such as finger injured.

2) Verifications Based on a Group of Persons: Group sequential fingerprint verification allows using several persons’ fingerprints together for verifications to protect sensitive information. Since the verification involves all members in the group, for each code in the sequence, we need to have a mechanism to tell which person it belongs. The way we do it is to give each member an ID and associates each code in the sequence with the ID of some member in the group.

For example, if there are three members in the group, we can assign their user IDs as ‘A’, ‘B’ and ‘C’ respectively. Based on the mapping in Figure 1, if the group sequential code is ‘5A1B6B0C5A7A6B7B1C’ the system would sequentially verify A’s right index fingerprint, B’s right ring fingerprint, B’s left index fingerprint, C’s right thumb fingerprint, A’s right index fingerprint, A’s left thumb fingerprint, B’s left index fingerprint, B’s left thumb fingerprint, C’s right ring fingerprint. By the way, if the members wanted to change their fingerprints in the group verification, they could simply change their associated codes.

IV. IMPLEMENTATION

In this section, we introduce the design and implementation of the system. We introduce the devices used first.

A. Equipment

To prove that our method can apply to most of fingerprint verification systems, we chose a traditional fingerprint
verification module, which contained some basic functions on fingerprint verification system.

Fig.2 the A04-WM100 optical fingerprint developing module

1) Hardware and middleware units: We chose the A04-WM100 fingerprint developing module with the optical reader, which was developed by the East Wind Technologies in Taiwan, as shown in Fig.2. This module contains 32 bit DSP (ADSP BF531) 396 MHz processor, the Dual RS232 port communication interface, and a 24-pin digital signal input from CMOS sensor for the fingerprint interface. The supporting are 9600, 19200, 38400, 57600 and 115200 bps for RS232/RS485, and the power consumption was 5V D.C., 220mA, when it is operating with the optical sensor. The matching Speed for 1-to-1 verification is 0.5 second, and the memory size can fit 500 users’ data. This module can be upgraded by applying middleware updates to extend the environment of software development. To achieve our objective of low cost implementation, we did not change the structure of the hardware and the middleware.

2) Software and the packet structure: The A04-WM100 fingerprint developing module uses Visual Basic and Visual C languages in its development; therefore, we chose Visual Basic as our implementation language. There is one exemplification program associated with the module, which included some basic functions such as the enrollment, verification, user data management, image display, image saving and other parameter setting. Some of the codes were written in the dynamic-link library (DLL) files as to protect the kernel technology.

Like other embedded systems, this module also needs the fixed packets to communicate with the main board. The packet structure shown in Fig.3, contains the header, data body and a checksum. The header was used to identify the packet, the data body would take the data, and the checksum is used to check if the packet is legal before the transmission.

Fig.3 the packet structure for A04-WM100 module

B. Data structure:

For the single user case, the user ID, password and role were used for individual identification and verification. The role category includes administrator and regular user. The fingerprint encoding data is used to record the fingerprint encoding information mentioned in section III, the sequential codes are used to do the sequential fingerprint verification, and the group associated data contains the groups in which the user participate.

For the group case, the group ID is used for identification of the group. The group password is used to verify the members in the group. To join the group, qualified users have to get the group password from the administrator. The group data also record the total number of members in the group and the total number of fingerprints. The last piece of data is the group sequential code. Based on our design, the group sequential code contains members’ IDs and their associated fingerprint codes. The associated codes help members easily change their fingerprints.

Fig.4 the structure of single user data and group data

C. System design:

The followings are the functionalities we implemented on the A04-WM100 hardware to create a low cost fingerprint verification system.

1) Enrollment for the single user case: This functionality supports users to enroll their information to the system. Information can be useful fingerprints, their user IDs, user passwords, fingerprint names and their associated codes. We
use the reserve fingerprint space in the hardware to store such information. Like most fingerprint verification systems, this module only has space for three fingerprints. In other words, each user could only register 3 fingerprints.

2) Data viewing for the single user case: This function allows users to view their own individual data, when they forgot them. The user data is protected by the user password, so no one can see these personal data except the user. The data included user ID, fingerprint encoding data and the sequential code.

3) Verification for the single user case: When the user data is established, the user can prepare to do the sequential verification. To prevent attackers to collect and analyze the error information, such as the length of the sequential code, we designed the system in such a way that it will ask users whether they want to continue to match the next fingerprint or not after matching each fingerprint in the sequential verification process. When users think that they have matched all fingerprints, they need to tell the system that the verification process is done. The threshold of the sequential verification i.e., the number of fingerprints used in the verification can vary depended on the number of fingerprint storage space that the hardware provides.

4) Enrollment for the single group case: This function provides two services, the enrollment of group data and the participation of group members. First, the system will request the user to fill in the group ID and the group password, then the system will check whether the group exists or not. If the group exists, the system will begin the process of the member participation and check the group password. After confirming that the user has the permission to join this group, the system will ask users to register their fingerprints to the security system by fingerprint enrollment number of total fingerprints they need to register. The group data included group ID, fingerprint encoding data and the sequential code.

5) Data viewing for the group members: This function is similar to user data viewing. If a user wants to view the group data, he has to fill in the group password. Only legal group members and the administrator can see the group data. The data included group ID, the number of total members, the number of total participated fingerprints, and the group sequential code.

6) Verification for the group and the authorization: The principle of the group sequential verification was similar to the single user verification. Because the length of group sequential code is longer than that of the single user, we can use the success number of fingerprint matching to grant authorization.

III. ADVANTAGES

We discuss the advantages of our design methodology below.

1) To reduce the consumption of storages: Many fingerprint verification system actually store fingerprints for each secret fingerprint sequence. In some situations, such as finger injury, we have to have the backup fingerprint sequence. Thus, physically storing each fingerprint sequence will consume lots of memory space especially in the group verification case. In our design, we only store one actual copy of each fingerprint and any fingerprint sequence can be represented as a sequence of mapping codes. This would greatly reduce the consumption of memory.

2) To make the modification or the creation of the secret fingerprint sequence easier: Without using a coding scheme to represent fingerprints internally such as those traditional verification readers, to create or modify the secret fingerprint sequence has to re-input those fingerprints. This is a very time-consuming and cumbersome work. In our design, after all fingerprints are registered, to create or modify fingerprint sequence is just a setup or change of some internal codes. This works even better in the group verification case e.g., when a member of the group can not use some of his fingers and the fingerprint sequence has to be changed.

3) To reduce the success rate of fingerprint replicas attacks: Obviously, attackers have to know both the mapping scheme and sequential codes before they can use prepared fingerprint replicas to pass the verification. It is much more difficult than the traditional verification readers. We suggest storing these two pieces of data separately to reduce the risks of been stolen. Even in the circumstance that one of the two pieces of data has been stolen, we still can easily change the other piece of information to stop the attacking process and gain more time to fix the system vulnerability.

4) To create more applications: Based on our method, we can create more applications. One example is the so called “challenge-response” application. Say there is an administrator of a lab in a university. To control the access to the lab, only those authorized personnel are allowed to enter the lab. The administrator could setup a table containing symbols to be mapped by fingerprints. He then can ask all authorized persons to register their fingerprints to the security system by using the table he created. When every person’s fingerprint mapping is created completely, the administrator could set a sequential code on the system that everyone could see. The authorized persons could pass the verification because they know the fingerprint mapping. For unauthorized persons, they still couldn’t pass the verification even they could see the sequential code. Also, the administrator could change the sequential code as often as he needs to without notifying anyone.

IV. PERFORMANCE COMPARISON

In this section, we compare the traditional fingerprint verification method with our approach under the fingerprint replica attacks. We assumed that attackers already got the entire fingerprint replicas. For the simplest case where most of the PCs allow using just one registered fingerprint to login, there is no protection at all because the attacker can just try ten times in the worse case to gain access to the PC. Some PCs using combination of two fingerprints are also not able to provide enough protection because the attacker can try only 45 combinations of two fingerprints from ten fingerprints to crack
the login protection. Assume the verification reader matches the pattern without order.

Next, we discuss the expression (1), which was used to calculate the worst case attacking time for the sequential fingerprint verification. Let \( S \) be the number of fingerprints which are used to construct the sequential codes. \( V_t \) is the time to perform one verification and \( m \) is the number of persons in the group involved in the verification process. Expression (1) shows the total attacking time.

\[
(10m)^2 S V_t,
\]

In the following, we estimate the attacking time by varying the \( S \) value, i.e., the number of fingerprints involved in the verification process. We would separate the discussion into the single user case and the group case.

1) Estimation of the single user verification: We set \( S \) value from 1 to 10, \( m \) is 1, and the \( V_t \) is 0.5 second which we measure from our implementation. The estimation results are shown in Figure 2.

![Figure 2. Estimation results for single user](image)

Figure 2 shows the total attacking time of the sequential method would extend when the length of sequential code increases. By using 4 fingerprints, the attacking time is about 5.5 hours, and by using 8 fingerprints in line with sequential code the total attacking time would extend to about 12.6 years.

2) Estimation for the group verification: In this case, we would use the same values for the parameters as in case 1. The number of participated members \( m \) is set to 3. The results are shown in Figure 3. If we choose the number of fingerprints as 9, the attacking time would be extended to about 2808647.2 years. By the way, if we increased the number of participated members, the attacking time would be higher.

![Figure 3. Estimation results for the group](image)

With these estimation results, we could see that the sequential fingerprint verification method could extend the attacking time. In other word, we could gain more time to detect the attacks and fix the system’s vulnerabilities.

V. CONCLUSIONS

In this paper, we present a design to improve the vulnerability of traditional fingerprint verification systems against fingerprint replica attacks. This design can raise the protection capability of a cheap traditional fingerprint verification system to the comparable level of a much more expensive biometric system. It also provides the feature of easy creation or modification of the secret fingerprint sequences for users to quickly change their secret sequence to prevent or stop attacks. This design can be totally developed in the software, so we can reduce the research cost on the hardware and middleware. Furthermore, this method will not be limited by the hardware, such as the types or fingerprint sensors. Therefore it can be applied to any verification hardware and any place where entity authentications are required.

REFERENCES


[14] Atmel Corporation, “Thermal Fingerprint Sensor with 0.4 mm x 14 mm (0.02” x 0.55”) Sensing Area and Digital Output (On-chip ADC)”, 2007.

