Software-hardware Cooperative Embedded Verification System Fusing Fingerprint Verification and Shared-key Authentication

By
Weiwei Yan


Department of Microelectronics and Information Technology (IMIT), School of Information and Communication Technology (ICT), Royal Institute of Technology (KTH), Shanghai, China
M.Sc Diploma Work in System-on-Chip Design Area
Abstract

In order to protect the security of the commercial information, personnel information, military information, governmental information on the Internet, the claimed identity should be authenticated. Now there are three main security authentication methods: first: using user PIN, such as password; second: using physical key, such as USBKey; third: using biological authentication technology, such as fingerprint, iris, voice and palm prints, etc.

Because of the uniqueness, invariance, and ubiquity properties of biometric authentication, biometric authentication is becoming popular, especially fingerprint recognition. However, when the fingerprint recognition information is transported on the public channel, it may be attacked, such as the fingerprint information is stolen. So a cryptology mechanism is needed to protect the fingerprint recognition information.

In the field of embedded security authentication system, the traditional hardware implementation mechanism, such as ASIC, can satisfy requires of functions and performances, but it is not configurable, flexible, and easy to expand; the traditional software implementation mechanism, such as general purpose processor, is flexible, but the cost and the power consumption are higher than hardware implementation.

In order to take the advantages of biometrics, cryptology, hardware implementation, and software implementation, a hardware-software cooperating embedded authentication system based on shared-key authentication and fingerprint verification is proposed. First, this system authenticates the identities of client and server by shared-key authentication, creates the current encrypt key and hash key, and then authenticates the identity of them via fingerprint recognition. During fingerprint recognition, the information of fingerprint is not needed to transmit over the public channel, so the security of fingerprint is increased. Theoretic analysis and experiments show that, this system reach very high authentication rate and security. This system can resist replay attack, server template attack, device template attack, effectively.

Keywords: fingerprint, shared-key, embedded system, hardware-software cooperating
## Content

Chapter 1: Introduction .............................................................................................................................. 1

1.1: Introduction of biometric technology .......................................................................................... 1

1.1.1 Classification of biometric authentication systems ........................................................... 1

1.1.2: Work flow of biometric authentication system ........................................................ 2

1.2 Fingerprint authentication ............................................................................................................ 5

1.2.1 The history of fingerprint authentication ................................................................... 6

1.2.2 Automatic Fingerprint Identification System ............................................................ 6

1.3 Overview of fingerprint authentication algorithms ...................................................................... 7

1.3.1 Fingerprint factors ..................................................................................................... 7

1.3.2 Workflow of AFIS ..................................................................................................... 9

1.3.3 Evaluate an AFIS ..................................................................................................... 13

1.4 Security authentication based on cryptology protocol ............................................................... 14

1.4.1 Security of Wireless local area networks ................................................................ 14

1.5 Motivation of this paper ................................................................................................... .......... 17

1.6 Organization............................................................................................................... ................ 18

Chapter 2: Fingerprint recognition............................................................................................. .............. 19

2.1 Fingerprint enhancement.................................................................................................... ........ 19

2.1.1 Median filter ............................................................................................................ 21

2.1.2 Normalization .......................................................................................................... 23

2.1.3 Orientation field ...................................................................................................... 24

2.1.4 Frequency field ........................................................................................................ 26

2.1.5 Region mask ............................................................................................................ 29

2.1.6 Gabor filter ............................................................................................................. 29

2.2 Fingerprint thinning ................................................................................................................... 31

2.3 Minutiae detecting and fingerprint matching ............................................................................. 32

2.3.1 Minutiae detecting ................................................................................................... 32

2.3.2 Fingerprint matching ............................................................................................... 34

Chapter 3: Security authentication based on cryptology mechanism....................................................... 42

3.1 Shared –key authentication ........................................................................................................ 42
Chapter 1: Introduction

At the history, in Chunqiu dynasty, Ban Lu, a Chinese man, developed the earliest identity authentication system – key and lock. This mechanical system may be the most original and simplest authentication system. With the rapid development of computer and network technology, information security becomes more important than ever, and as the precondition mechanism of information security, identity authentication is regarded more and more important. Currently, the traditional “User name + Password” and physical medium methods are not enough for information security, because they are threatened by the high-tech key breaking and physical medium forging. So developing a bran-new, secure, and convenient identity authentication technology is becoming pressing. Now, the biometric technologies attract human’s attention, and fingerprint technology is one of the earliest and most developed biometric technologies.

![Figure 1-1: The forecast of biometric industry](image)

1.1: Introduction of biometric technology

There are three stages of chasing a secure and convenient way to protect the security of file, business and stuff, etc. The first stage is the mechanical keys, such as
lock-key; the second stage is the digital keys, such as passwords or bar codes, which are developed from mechanical keys; the third stage is using the physical or behavioral characteristics to verify the identity, biometric authentication, which is the most secure authentication system in the digital life, now.

Compared with traditional authentication methods, biometric authentication has the following advantages: not easy to be forgotten or stolen, not easy to be forged, convenient. In additional, based on the IBG (International Biometric Group) report of 2009, in the following years, the revenues of biometric industry market are growing very fast. They will increase to $9,368.9 to 2014, which is shown on figure 1-1.\[1\]

1.1.1 Classification of biometric authentication systems

As mentioned, biometric authentication is based on physical or behavioral characteristics. Currently, of biometric authentication based on physical characteristics such as face, fingerprint, hand geometry, hand vein, iris, retinal pattern, fingerprint recognition is most developed; of biometric authentication based on behavioral characteristics such as signature, voice and facial thermograms, signature and voice are the most developed.

Why physical or behavioral characteristics can be used in identity authentication systems? The reason is they satisfy the following requirements: (1) universality, which means that every person should have the characteristics; (2) uniqueness, which indicates that no two persons should be the same in terms of characteristics; (3) performance, which means that the characteristics should be invariant with time; (4) collectability, which indicates that the characteristics can be measured quantitatively. In practice, there are some other requirements: (1) performance, which refers to the achievable identification accuracy, the resource requirement to achieve an acceptable identification accuracy, and the working or environmental factor that affect the identification accuracy; (2) acceptability, which indicates to what extent people are willing to accept the biometric system; (3) circumvention, which refers to how easy it is to fool the system by fraudulent techniques.\[2\] Until now, there is no single biometric characteristic can satisfy all of the requirements, and they all have their
advantages and weakness. Table 1-1 will show the comparisons of the several biometric technologies.[3]

<table>
<thead>
<tr>
<th>Biometric</th>
<th>Universality</th>
<th>Unique</th>
<th>Permanence</th>
<th>Collectable</th>
<th>Performance</th>
<th>Acceptable</th>
<th>circumvention</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fingerprint</td>
<td>medium</td>
<td>high</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
</tr>
<tr>
<td>Iris</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>low</td>
<td>high</td>
</tr>
<tr>
<td>Face</td>
<td>high</td>
<td>low</td>
<td>medium</td>
<td>high</td>
<td>low</td>
<td>high</td>
<td>low</td>
</tr>
<tr>
<td>Retinal Scan</td>
<td>high</td>
<td>high</td>
<td>medium</td>
<td>low</td>
<td>high</td>
<td>low</td>
<td>high</td>
</tr>
<tr>
<td>Hand Geometry</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
</tr>
<tr>
<td>Hand Vein</td>
<td>medium</td>
<td>high</td>
<td>high</td>
<td>medium</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
</tr>
<tr>
<td>DNA</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>high</td>
<td>low</td>
<td>high</td>
</tr>
<tr>
<td>Signature</td>
<td>low</td>
<td>low</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>high</td>
<td>low</td>
</tr>
<tr>
<td>Voice Print</td>
<td>medium</td>
<td>low</td>
<td>low</td>
<td>medium</td>
<td>low</td>
<td>high</td>
<td>low</td>
</tr>
<tr>
<td>Gait</td>
<td>medium</td>
<td>low</td>
<td>low</td>
<td>high</td>
<td>low</td>
<td>high</td>
<td>medium</td>
</tr>
<tr>
<td>Keystroke</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>medium</td>
<td>low</td>
<td>medium</td>
<td>medium</td>
</tr>
</tbody>
</table>

From table 1-1, we can see that no one biometric technology is better than the others at every factor. Currently, fingerprint is the most developed and profitable biometrics technology, because of better uniqueness, permanence, and technique advantage. Figure 1-2 shows the revenues from biometric technologies in 2009. It is obvious that the revenue from the biometric technology about fingerprint (Fingerprint and AFIS / Live-Scan) takes the largest part.[1]

![Biometric Revenues by Technology, 2009](image)

Figure 1-2: Revenue of biometric technologies in 2009

Now, many biometrics technologies have been used in our daily life, such as fingerprint recognition, facial recognition, iris recognition, signature verification, etc. Due to advantages and weakness of every biometric technology, the integration of several biometrics into one system is a trend. But fingerprint is widest used compared
with other biometric authentication technologies, currently, because of the low cost of device, easy processing of the scanned fingertip image, etc.

1.1.2 Work flow of biometric authentication system

Figure 1-3 shows how the biometrics authentication system works: (1) biometric abstraction: the raw biometric image is abstracted by the biometric scanning device; (2) biometric enrollment: the raw image is processed to get the measured characteristics, and the template is generated; (3) template storage: after the template is generated, the template is stored in memory to be verified with the input measured characteristics; (4) biometrics verification: the live-scanned image template is verified with the stored template; (5) verification result: after verification, whether the visitor is legal to access the system is decided.

With fingerprint recognition system, (1) biometric abstraction: which indicates that fingertip is scanned by fingerprint sensor, and fingertip image is generated; (2) biometric enrollment: which means that fingerprint template is generated, after the preprocessing, thinning, and minutiae abstract of fingertip image; (3) template storage: which indicates that fingerprint template is storied in memory; (4) biometric verification: which means that the live-scanned fingertip image is processed, and it is verified with the storied template via matching algorithm; (5) verification result: which indicates that after the verification, a matching score is generated, and compare it with a pre-specified threshold, if it’s lager than the threshold, the user is legal, otherwise, illegal.
1.2 Fingerprint recognition

A Fingerprint is the pattern on the fingertip, as shown on figure 1-4.[²] There are several authentication approaches of fingerprint: matching the minutiae abstracted from fingertips, directly matching the pattern of the fingerprint, etc. In this paper, the first one is chosen because of high recognition accuracy. This approach is based on that the bifurcation, and ending on fingertips are different, which are called minutiae during processing. These minutiae are different between every person, and every fingertip, and they also keep invariant day after day, year after year. These are the uniqueness and invariance of fingerprint. Based on the uniqueness, different persons can be recognized; based on the invariance, there is no need to worry that the same person can not be recognized at different times.

Figure 1-4: Fingerprint patterns and a fingerprint classification schema of six categories: (a) arch, (b) tented arch, (c) right loop, (d) left loop, (e) whorl, and (f) twin loop
1.2.1 The history of fingerprint authentication

Humans have used fingerprints for a very long history. Based on the proof of archaeology, before B.C. 7000 to 6000 year, fingerprint had been used as an approach of identity authentication in the ancient city of Jericho. In the book Archaeology of the Holy Land, Dame Kathleen Kenyon reported that the bricks of which the walls were constructed were made by hand (not in moulds, as is usual later), in shape rather like a flattened cigar, with the surface impressed with a herringbone pattern by pairs of prints of the brick-layer’s thumb, thus giving a keying such as is provided by the hollow in modern bricks. In 1684, English plant morphologist Nehemiah Grew, who is believed to be the first fingerprint pioneer, published a paper which reported his study on ridges, furrows, and pores on the hand and foot surfaces, in the most beautiful phraseology, description. Since then more researchers were attracted to fingerprint researching. Thomas Bewick, who was alive between 1753 and 1828, used an engraving of his fingerprints as a signature in his few books, which is believed to be an important contribution to the fingerprint study. On December 22, 1823, Joannes Evangelista Purkinje published his thesis, in which he classified the fingerprint to several types based on ridges, furrows, and pores. The classification method enabled fingerprint forms bearing differing patterns to placed in a certain order, thus enabling the search area to ne minimized. Henry Faulds’ letter to Nature, in 1880, announced that fingerprints were sufficiently personal in pattern to supply a long-wanted method of scientific identification. Sir Francis Galton was a great fingerprint pioneer, who made great contribution for fingerprint research at Nineteenth Century. Later in 20th century, with the emergency of computer, it searches of crime scenes imprints provide excellent result, but because of the limitation of algorithms and not perfect integrity of fingerprint template storage, the compute-aid fingerprint authentication is not 100% efficient.

1.2.2 Automatic fingerprint identification system
AFIS (Automatic Fingerprint Identification System) uses computer to verify whether two fingerprints are from the same fingertip. Although computer-aid fingerprint authentication is not 100% efficient as said last section, AFIS still attract many attentions, with the fast developing of computer, image processing, and pattern authentication. AFIS was developed by FBI (Federal Bureau of Investigation), Home Office in UK, and Paris Police Department from early 1960’s.[2] It was developed so successful that a large number of AFIS’s are currently installed and in operation at law enforcement agencies and civilian applications.[2]

1.3 Overview of fingerprint authentication algorithms

Currently, widely used fingerprint authentication mode is based on fingerprint minutiae, which is proposed by FBI.

Now, fingerprint authentication system saves fingerprint minutiae abstracted from fingerprint image instead of directly saving fingerprint image, because of human privacy and memory storage. So, fingerprint authentication algorithms are used to abstract factors from fingerprint and match them.

1.3.1 Fingerprint factors

There are two types of fingerprint factors: whole factor and local factor.

1. Global factor

The global factor is the factors that can be seen with human eyes. It includes fingerprint pattern, core, delta, and density of ridge lines. Fingerprint patterns are shown on figure 1-4, arch, tented arch, right loop, left loop, whorl, and twin loop. Core, delta and density of ridge lines are shown on figure 1-5. These factors are level 1 factors of fingerprint which are obvious to human eyes.
2. Local factor

Local factors of fingerprint are minutiae on fingertip. Two fingerprints may have similar whole factors, but the local factors are totally different. Currently, with wildly used fingerprint classification, there are six types: ending, bifurcation, line unit, line fragment, eye, and hook, which are shown on figure 1-6.\textsuperscript{14} These minutiae are unique information to do fingerprint authentication. With investigation, there are more than 150 local minutiae, and the existing probabilities of these factors are different, and many minutiae very infrequently exist. The six types of minutiae shown on figure 1-6 are familiar.

![Figure 1-6: Local factors of fingerprint](image)

Of the six types of minutiae, ending and bifurcation are relatively stable, and not sensitive to the pressure from scanning. So, FBI advanced to use them in fingerprint matching mode based-on minutiae. Further, the minutiae of fingerprint include type, location and orientation, and we can use these minutiae to configure a fingerprint.
1.3.2 Workflow of AFIS

The workflow of AFIS is shown on figure 1-7. “Sensor” is the first stage of AFIS, which means acquisition of fingerprint image. Inked (off-line) and ink-less (live scan) are the primary two method of capturing a fingerprint image. The steps to capture the inked fingerprint image: obtaining an impression of an inked fingertip on a paper, and then scanning the image using a flat bed document scanner. Acquisition of inked fingerprint is not convenient, and not suitable to be used in an identity authentication system. But special kind of off-line images, extremely important in forensic application, are so-called latent fingerprints found at crime scenes. Due to the oily nature of the skin, the impression of a fingerprint can be deposited on a surface that is touched by a finger. What’s more critical is that the impression can be lifted from the touched surface by employing certain chemical technique. [5]

On the other hand, Ink-less fingerprint capture is a method, in which a fingerprint image can be gotten directly from the fingerprint sensor without the intermediate step of getting an impression on a paper. Currently, a number of live-scan sensing mechanisms can be used to capture the ridges and valleys present in fingertip such as optical FTIR, capacitive, thermal, pressure-based, ultrasound, etc. Figure 1-8 shows fingerprint images scanned with off-line and on-line sensor. [8] Although optical scanner has been used for the longest time, the new solid-state state sensors are more and more popular, for the small size and ease of being embedded to electronical devices.

“Fingerprint Preprocessing” is the second stage of AFIS, which indicates that the scanned fingerprint image should be processed to make getting the minutiae easier. In
the authentication system of this paper, this step includes fingerprint enhancement, binary, and thinning. The purpose of fingerprint enhancement is to minimize the noise of the image, and improve the clarity of ridge structure. Because the goal of enhancement algorithm is to improve the clarity of ridge structures

![Fingerprint Images](image.png)

(a) (b) (c)

Figure 1-8: Fingerprint image from: a) a live-scan FTIP-base optical scanner; b) a live-scan capacitive scanner; c) a live-scan thermal scanner; d) an off-line inked impression; e) a latent fingerprint

of input fingerprint images to facilitate the extraction of minutiae, spurious ridge structure should be not generated after enhancement processing. The objective of thinning is to thin the width of fingerprint to one pixel, in order to facilitate the extraction of minutiae. Figure 1-9 shows the fingerprint images before and after enhancement and thinning processing. As shown on figure 1-9, after enhancement, the clarity of ridge structure is improved, and after thinning, the minutiae can be extracted from the image via some 3x3 templates easily.
“Minutiae Abstraction” is the third stage of AFIS, the goal of which is to abstract the minutiae of pre-processed image. The variety of fingerprint minutiae has been talked in last section, of them, ending and bifurcation are used to represent fingerprint.

“Template Storage” is the last step of enrollment in AFIS, which is to store the representation of fingerprint as template to be matched. Storing raw fingerprint images may be problematic for large AFIS. In 1995, the size of the FBI fingerprint card archive contained over 200 million items, and archive size was increasing at the rate of 30,000 to 50,000 new cards every day. In embedded biometric application, storage is also at a premium. For example, in smartcard, typically 2Kbytes of storage are available. In such situations, the fingerprint representation should be compact. On the other hand, the fingerprint representation should contain distinctive information about the fingerprint, in order to keep the uniqueness of fingerprint.

“Minutiae Matching” is the last step of AFIS, the objective of which is to determine whether two fingerprint representations are from the same finger. The standard and threshold of determine whether two fingerprint representations are from the same finger are defined at this stage.

In case of minutiae-based matching, it is reduced to match minutiae pattern of fingerprint. In ideal situation, the corresponding minutiae from two fingerprints are known, there is no deformation and rotation, and each minutiae present is exactly localized. In this situation, matching is just counting the matching pairs of minutiae. But in real situation because of displacement, rotation, partial overlap, non-linear distortion, variable pressure, changing skin condition, noise, and feature extraction error, fingerprints from the same finger may look quit different, and fingerprints from
different fingers look quite similar, as shown in figure 1-10.\[5\]

![Figure 1-10: Difficulty in fingerprint matching: (a) and (b) are from the same finger, (c) and (d) are from different fingers.](image)

In light of the real situation, human fingerprint examiners evaluate the following several factors to estimate whether two fingerprints are from same finger: 1) global pattern configuration agreement, which means two fingerprints must be of the same type; 2) local minutiae agreement, which requires that corresponding minutiae details must be same and the number of same minutiae must be larger than a settled threshold; 3) the relationship between the minutiae details, which specifies that the connection between the minutiae details must be identical.\[5\] In AFIS, the same mechanism as human fingerprint examiners using does not be followed. Actually, after developing for several decades, a large number of matching approaches have been designed, including minutiae-based matching which is inspired by the manual procedure. A categorization of fingerprint matching approaches is: correlation-based matching, ridge feature-based matching, and minutiae-based matching. In minutiae-based
matching, minutiae are extracted from the two fingerprints and stored as sets of points in the two-dimensional plane. Minutiae matching essentially consists of finding the alignment between the template and the input minutiae sets that results in the maximum number of minutiae paring.\[5\]

1.3.3 Evaluate an AFIS

The result of a matcher in a fingerprint recognition system is typically a matching score, which is in the range of [0, 1], and indicates the similarity between the input fingerprint and the database template. The closer the score is to 1, the more certain that the two fingerprints come from the same fingerprint; the closer the score is to 0, the certain that the two fingerprints come from different fingerprints. The system decision is regulated by a threshold: if matching score of two fingerprints higher or equal to the specified threshold, they are regarded as matching pairs; if lower than the specified threshold, they are inferred as non-matching pairs. In the system of this paper, an empirical value 0.3 is selected as a matching score in this paper.

In a typical fingerprint verification system, two performance is cared: mistaking fingerprint measurement from two different fingers to be from the same finger (called false match) and the percentage of doing the right estimation that taking the fingerprint measurement from the same fingers to be from the same finger (called right match). Note that these two performances are also denoted as right acceptance and false rejection. In practical use, the higher of right acceptance and the lower of false rejection are expected.

Except the two performances, operating time of an AFIS is another key factor to be considered. The time of an AFIS costs the time of scanning a fingerprint, the time of fingerprint image pre-processing, the time of matching. The time of fingerprint image pre-processing is mainly the time of fingerprint enhancement. The time of fingerprint enhancement includes normalization estimation, orientation field computation, frequency field estimation, region mask, and Gabor filter processing. Table 1-2 shows the time of fingerprint enhancement processing in software, we can see that the time of orientation filed estimation, frequency filed computation, and
Gabor filter processing takes a large part of the enhancement time, so these three parts are implemented in ASIC in this paper, in order to accelerate the speed of fingerprint recognition system.

Table 1-2: The time of fingerprint enhancement

| The Run Time of Enhancement algorithm on Intel Pentium 2 E2140 @1.60GHZ Median Filter (seconds) Normalization (seconds) Orientation (seconds) Frequency (seconds) Region Mask (seconds) Gabor Filter (seconds) Total (seconds) |
|---|---|---|---|---|---|---|
| 0.109 | 0.016 | 1.546 | 2.641 | 0.031 | 3.875 | 8.218 |

1.4 Security authentication based on cryptology protocol

In the history, many elegant authentication scheme and powerful access control mechanisms have been developed. With these authentication scheme and access control mechanism, the system can decide whether a person has the right that he/she claimed. Currently, most security systems still employ the traditional security authentication mechanism, such as user-password, another widely used mechanism is to use physical medium that you have – a card key, smart card and token, and more secure mechanism is to fuse user-password and physical medium, for example when using bank card at the ATM machine, password and bank card should be used at the same time.

1.4.1 Security of Wireless local area networks

Figure 1-11: Wireless concept: WLAN, Bluetooth
In recent years, ubiquitous access to IP networks is becoming increasingly important and popular. Current trends indicate that wide-area wireless IP networks such as those based on third-generation (3G) CDMA-200 and Universal Mobile Telecommunications System (UMTS), and local area wireless IP networks such as those based on IEEE 802.11 will compete and coexist to provide such access. In fact, 802.11 has become one of the most popular and easiest ways to provide wireless access to enterprises, homes, and public hotspots, and has seen explosive growth due to low cost of deployment.\cite{6} Figure 1-11 shows wireless concept, the range of WLAN is from 0 to 100 meters. IEEE 802.11 specifies frequency distribution, configuring mode, and security mechanism of WLAN.

However, security is a serious issue because the wireless medium is open to public channel with a certain range, as shown in Figure 1-12. There are two aspects common to all kinds of wireless networks: authentication and encryption. Authentication: before the terminal or end user can access the authentication, authorization, and accounting server, the identity of the terminal or end user will be authenticated by the server. If the terminal or end user is legal visitor to the server, it will get the services from the server; if not legal, its accessing request will be denied. Encryption: the data transmitted on the air interface between the user and the server. Often symmetric encryption technology is used, and usually, the key used in the symmetric encryption is derived during the authentication phase. Different wireless technology use different encryption technology. For example, 802.11 networks use simple shared-key authentication; 3G code-division multiple access (CDMA) networks use symmetric encryption.

Actually, there are three identity authentication methods in 802.11, including
public SSID (Service Set ID) in open system, non-public SSID in closed system, and shared-key authentication; the confidentiality of data is realized by WEP (Wired Equivalent Privacy); the integrity of data is promised by CRC-32, as shown in table 1-3.

<table>
<thead>
<tr>
<th>Authentication</th>
<th>SSID</th>
<th>WEP</th>
<th>Confidentiality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open System</td>
<td>No Need</td>
<td>No Need</td>
<td>Not Support</td>
</tr>
<tr>
<td>Closed System</td>
<td>Need</td>
<td>No Need</td>
<td>Not Support</td>
</tr>
<tr>
<td>Shared Key</td>
<td>Need</td>
<td>Need</td>
<td>Support</td>
</tr>
</tbody>
</table>

![Table 1-3: The authentication methods of 802.11](image)

WEP is the first Security mechanisms for IEEE 802.11. WEP means that an optional cryptographic confidentiality algorithm specified by IEEE 802.11 that may be used to provide data confidentiality that is subjectively equivalent to the confidentiality of a wired local area network (LAN) medium that does not employ cryptographic techniques to enhance confidentiality. In WEP, shared-key authentication was used, which is shown in figure 1-13. In figure 1-13, STA stands for station, and AP stands for access point.

As shown on the shared-key authentication flowchart:

1: The authentication-initiating wireless client sends a frame consisting of an identity assertion and a request for authentication.

2: The authenticating wireless node responds to the authentication-initiating wireless node with challenge text.

3: The authentication-initiating wireless node replies to the authenticating wireless node with the challenge text that is encrypted using WEP and an encryption key that is derived from the shared key authentication secret.

4: The authentication result is positive if the authenticating wireless node determines that the decrypted challenge text matches the challenge text originally sent
in the second frame. The authenticating wireless node sends the authentication result.

1.5 Motivation of this paper

As mentioned before, there are three stages of chasing the security authentication. Currently, in the digital life, there are three methods of security authentication: 1) password, user remember the authentication content; 2) physical password, user keep some physical medium to proof legal identity, such as USBKey; 3) biometric authentication, using the intrinsic factor of human, such as fingerprint.\[^{[8]}\]

Now, the first method is widely used. In order to be convenient, users use simple number or letter as password, and do not change the password frequently, called “Static Password”. But this authentication mechanism is not secure, the password is easy to forgotten, stolen or guessed, and maybe damage is caused. The second method, if the physical security authentication medium is lost or stolen, then the illegal user can complete the authentication via the medium, so this method is also very weak on security authentication. In some systems which need very high security level, the cooperation of the first and second methods is used, such as that the ATM machine require both the hank card and the password. But with this method, maybe, the physical medium and the password are still stolen. The third method, the advantage is the identity authentication is connected with the irreplaceable biometric characteristics, to implement the more secure identity authentication.\[^{[5]}\]

In order to resolve the upper issues, an embedded authentication system mechanism and structure is proposed in this paper: it has higher secure level via fingerprint recognition; the fingerprint recognition information is protected via shared-key authentication. The fingerprint recognition systems based on DSP, ARM are widely used in daily life, but consumer electronical products become smaller and smaller, and at the same time people require the products more and more secure. The current fingerprint recognition systems are not small enough, and not convenient to be embedded in mobile product, so they do not satisfy requirement. In this condition, developing fingerprint recognition systems which are implemented with ASIC is necessary and has a broad market foreground. At the same time, software is convenient to modified, and costs less than hardware, so the parts which costs little
time with software are still implemented in software. So the system proposed in this paper was implemented with software-hardware cooperation.

### 1.6 Organization

This paper will discuss the design and implementation of this embedded authentication system with the following organization.

In chapter one, the background of security authentication, fingerprint recognition and shared-key authentication is discussed.

In chapter two, the algorithms used in fingerprint recognition are analyzed, including fingerprint enhancement, thinning, minutiae extraction and matching.

In chapter three, the shared-key authentication is discussed, and the security algorithm, SMS4, used in this paper is also analyzed.

In chapter four, the implementation of this authentication system is discussed detailed, including the software implementation and software-hardware cooperating implementation.

In chapter five, the metric and testing result of this system is mentioned.

In chapter six, the conclusion and future work is given.
Chapter 2: Fingerprint recognition

Fingerprint authentication can be separated into two categories: verification and identification. Verification is comparison of a claimant fingerprint against an enrollee fingerprint, where the intention is that the claimant fingerprint matches the enrollee fingerprint. Identification is that a fingerprint of unknown ownership is matched against a database of known fingerprints to find out the owner. In this paper, verification system is designed. As shown in figure 1-7, verification flowchart has a step, fingerprint pre-processing which includes fingerprint enhancement, thinning.

2.1 Fingerprint enhancement

Automatically and reliably extracting minutiae is a critical step in fingerprint matching. The performance of minutiae extraction heavily relies on the quality of input fingerprint image. In ideal situation, the alternation and flow of ridge and valley should be clear enough to make the minutiae of the ridge are clear, like ending and bifurcation as shown in figure 2-1 (a).\textsuperscript{9} In this situation, the minutiae can be extracted and located easily and correctly from the thinned image, as shown in figure 2-1 (b).\textsuperscript{9} But in actually, because of the variation of skin condition, acquisition device, impression condition, a significant percentage (about 10\%) of fingerprint is of poor quality.\textsuperscript{9} In the fingerprint images of poor quality, the ridge and valley flow are not clear, and the minutiae are not easily detected, as shown in figure 2-2. The poor
quality also leads to the following problems: 1) spurious minutiae will be created; 2) a large percent of genuine minutiae will be ignored; 3) many error in the location (position and orientation) of minutiae will be introduced.\textsuperscript{[9]} In order to make the extraction of minutiae easily and precisely in the images of poor quality, enhancement algorithm is need to improve the quality of images and clarity of ridge.

There are variety of research activities along the stream of reducing noises and improve the contraction between ridges and valleys in gray-level fingerprint images.\textsuperscript{[9]10} Because ridges and valleys on the fingerprint run parallel with each other, and parallel ridges and valleys form sinusoidal-shaped plane waves, in local neighborhood. So in many papers, a Gabor filter is tuned to local orientation and frequency to implement fingerprint enhancement.

![Fingerprint of poor quality](image1.png)  ![Fingerprint of poor quality](image2.png)

Figure 2-2: Fingerprint of poor quality

Gabor filters are band-pass filter, conforming well to mammalian visual receptive field profiles. They have tunable orientation and radial frequency bandwidths, tunable center frequencies, and optimally achieve joint resolution in space and special frequency. By comparing the channel amplitude responses, boundaries between textures can be detected. By locating large variations in the channel phase responses, discontinuities in texture phase can be detected.\textsuperscript{[10]} Fingerprint image is a special kind of texture image, whose texture is ridges and valleys of fingertips. Its ridges are parallel in local neighborhood, and the frequencies in local neighborhood are constant in the form of sinusoidal-shaped plane waves. So it’s very useful to employ 2-D Gabor filter to improve the clarity of fingerprint image.
As we said, Gabor filters are orientation and frequency tunable, so orientation and frequency of fingerprint image must be computed before employing Gabor filter. In order to computer orientation and frequency precisely, a relatively clear fingerprint image is needed, so the first two steps of enhancement in this paper are median filter and normalization. The following segmentation will analyze the several steps of fingerprint enhancement, as shown in figure 2-3.

### 2.1.1 Median filter

Impulse noise is one most common kind of noise in the electronic signals, which usually caused by unstable voltage. It changes one bit from “1” to “0” or from “0” to “1”. If this happens in the image, it changes one pixel from white to black or from black to white, and further if it happens in gray-level image, the value of pixel is changed from one value to another value between 0 and 255. In order to remove the impulse noise, many methods have been proposed, and median filter is the most widely used method.
How median filter works? Like the mean filter, median filter considers each pixel in the image in turn and looks at the neighborhood of the pixel to determine whether or not it is representative of it surrounding. Instead of simply replacing it with the mean value of its neighborhood, the mid-value of neighborhood is used. The steps are: 1) sorting the pixel values of its neighborhood; 2) select the middle pixel value, and replace the pixel with it, as shown in figure 2-4. There are two advantages of median filter compared with mean filter: 1) more robust average value, because the median is not the average of the neighborhood’s values, a very unrepresentative value in the neighborhood will not affect the result significantly; 2) Since the result is actually one of the neighborhood values, not one value derived from them, the sharp edges are preserved. One of the goals of fingerprint enhancement is to increase contraction of ridges and valleys, but at the same time, the impulse noise must be removed, so a median filter is a suitable choice at keeping the contraction and removing the impulse noise, due to the two advantages. Figure 2-5 shows an example of fingerprint image after median filter processing.

![Figure 2-5: Example of fingerprint image after median filter processing](image)

In the referenced enhancement algorithm [9], no median filter is used, so the enhancement results of fingerprint images with impulse noise are not ideal enough. The result of the extraction is also affected, and the performances of matching are not good. The original enhancement flow chart is shown in figure 2-6. Comparing the
enhancement results of referenced algorithm and improved enhancement algorithm with median filter, the contraction of ridges and valleys of improved algorithm is clearer, and the precise frequency is more easily to extract.

2.1.2 Normalization

The formula of normalization is shown as following, in which $M_0$ and $VAR_0$ mean pre-specified mean and variance. $M$ and $VAR$ are mean and variance of input fingerprint image. $I(i,j)$ represents the intensity of the pixel at $ith$ row and $jth$ column. $G(i,j)$ denotes normalized intensity on $Pixel(i,j)$.

$$G(i,j) = \begin{cases} 
M_0 + \frac{VAR_0(I(i,j) - M)^2}{VAR} & \text{if } I(i,j) > M \\
M_0 - \frac{VAR_0(I(i,j) - M)^2}{VAR} & \text{otherwise}
\end{cases} \quad (2-1)$$

From (2-1), we can see that normalization is a pixel-wise operation, so it can not change the clarity of ridges and valleys of fingerprint. The main goal of normalization is to reduce the variance along ridges and valleys to facility the following operation. Figure 2-7 is an example of fingerprint image after median filter and normalization processing.
2.1.3 Orientation field

Orientation field describes one of the basic structures of fingerprint, and is an intrinsic property of fingerprint image. It defines invariant coordinates for ridges and valleys in a local neighborhood.\cite{9} By viewing the fingerprint as texture image, a number of algorithms have been proposed to estimate the orientation field.\cite{9}\cite{11}\cite{12} To obtain reliable ridge orientation, of the proposed algorithms, the most popular approach is to go through the gradients of gray intensity. At the same time considering the computation cost, the approach proposed in reference [9] is selected in this paper.

The estimation steps in reference [9] include:

1) Divide the normalized fingerprint images into blocks of size $w \times w$ (15 x 15).

2) Compute the gradients $\partial_x(i, j)$ and $\partial_y(i, j)$ at each Pixel$(i, j)$ with Sobel operator. Sobel operator is shown in figure 2-8. The $\partial_x(i, j)$ and $\partial_y(i, j)$ are computed as following:

$$
\partial_x(i, j) = 2I(i+1, j-1) + 5I(i+1, j) + 2I(i+1, j+1) \\
-2I(i-1, j-1) - 5I(i-1, j) - 2I(i-1, j+1)
$$

(2-2)
\[
\partial_y(i, j) = 2I(i-1, j+1) + 5I(i, j+1) + 2I(i+1, j+1)
- 2I(i-1, j-1) - 5I(i, j-1) - 2I(i+1, j-1)
\] (2-3)

Figure 2-8: Sobel operator

3) The local orientation of each block centered at \text{Pixel}(i,j) using the following equations:

\[
\nu_x(i, j) = \sum_{u=\frac{-W}{2}}^{\frac{W}{2}} \sum_{v=\frac{-W}{2}}^{\frac{W}{2}} 2\partial_x(u,v)\partial_y(u,v)
\] (2-4)

\[
\nu_y(i, j) = \sum_{u=\frac{-W}{2}}^{\frac{W}{2}} \sum_{v=\frac{-W}{2}}^{\frac{W}{2}} (\partial_x^2(u,v) - \partial_y^2(u,v))
\] (2-5)

\[
\theta(i, j) = \frac{1}{2} \tan^{-1} \left( \frac{\nu_y(i,j)}{\nu_x(i,j)} \right)
\] (2-6)

where \(\theta(i,j)\) is the estimate of the local ridge orientation at the block centered at \text{Pixel}(i,j).

4) Due to the presence of noise, in the input fingerprint image, the estimated local orientation \(\theta(i,j)\) may be not always correct. Since local ridge orientation varies slowly in local neighborhood, a low-pass filter can be used to modify the incorrect local orientation. In order to modify the incorrect local orientation, a continuous vector value is needed to derive from the local orientation \(\theta(i,j)\). The following method is used to get the continuous vector:

\[
\phi_x(i, j) = \cos(2\theta(i,j))
\] (2-7)

\[
\phi_y(i, j) = \sin(2\theta(i,j))
\] (2-8)

where \(\phi_x(i,j)\) and \(\phi_y(i,j)\) are the continuous vectors at x direction and y direction. The low-pass filter can be used as following approach:

\[
\phi_x'(i, j) = \sum_{u=-\frac{W}{2}}^{\frac{W}{2}} \sum_{v=-\frac{W}{2}}^{\frac{W}{2}} W(u,v)\phi_x(i-uw, j-vw)
\] (2-9)

\[
\phi_y'(i, j) = \sum_{u=-\frac{W}{2}}^{\frac{W}{2}} \sum_{v=-\frac{W}{2}}^{\frac{W}{2}} W(u,v)\phi_y(i-uw, j-vw)
\] (2-10)
where $W$ is the low-pass filter and $W_\phi$ is the size of low-pass filter. The size of this 2-D low-pass filter is $5 \times 5$.

5) Then compute the low-passed local orientation in the following way:

$$O(i, j) = \frac{1}{2} \tan^{-1}(\frac{\phi_x(i, j)}{\phi_y(i, j)}) \quad (2-11)$$

From the steps of estimating orientation in reference [9], we can see that the computation is very complex. Actually, there is no need to compute the local orientation in step (3). The mathematical principal of step (3) is a trigonometric function, as shown in following function:

$$\tan 2\theta = \frac{2 \tan \theta}{1 - \tan^2 \theta} = \frac{2 \cos \theta \sin \theta}{\cos^2 \theta - \sin^2 \theta} = \frac{2\partial_x \partial_y}{\partial_x^2 - \partial_y^2} \quad (2-12)$$

$$\theta(i, j) = \frac{1}{2} \arctan \left[ \frac{\sum_{u=-w/2}^{w/2} \sum_{v=-v/2}^{v/2} 2\partial_x(u, v)\partial_y(u, v)}{\sum_{u=-w/2}^{w/2} \sum_{v=-v/2}^{v/2} (\partial_x^2(u, v) - \partial_y^2(u, v))} \right] \quad (2-13)$$

In step (4) the local orientation is converted in to continuous vector values $\sin \theta$ and $\cos \theta$ again. Actually, $\sin \theta$ and $\cos \theta$ can be computed in the following way,

$$\sin \theta = \frac{\partial_y}{\sqrt{\partial_x^2 + \partial_y^2}}, \cos \theta = \frac{\partial_x}{\sqrt{\partial_x^2 + \partial_y^2}} \quad (2-14)$$

$$\sin(2\theta) = 2 \sin \theta \cos \theta = \frac{\partial_x(i, j)\partial_y(i, j)}{\partial_x^2(i, j) + \partial_y^2(i, j)} \quad (2-15)$$

$$\cos(2\theta) = \cos^2 \theta - \sin^2 \theta = \frac{\partial_x^2(i, j) - \partial_y^2(i, j)}{\partial_x^2(i, j) + \partial_y^2(i, j)} \quad (2-16)$$

So the inter-step of compute local orientation is avoided. The computation cost is decrease, especially in ASIC implementation, if local orientation estimation is implemented, with approach of LUT (Look up table), it will not only waste time, also waste memory for LUT.

### 2.1.4 Frequency field

In gray-level image, the ridges and valleys form a sinusoidal-shaped wave along the direction normal to the local orientation, where no minutiae and singular points appear (See Figure 2-9).[9] So frequency is another instinct property of fingerprint image.
The steps of estimate frequency are shown as following:

1) Divide the normalized image in to blocks of size \( w \times w \) (15 x 15).

2) For each block centered at Pixel \((i,j)\), compute an oriented window of size \( l \times w \) (32 x 16), as shown in figure 2-9.

3) Derive the x-signature, \( X[0], X[1], \ldots X[l-1] \), of the ridges and valleys in the oriented window. The x-signatures are computed in the following way:

\[
X[k] = \frac{1}{w} \sum_{d=0}^{w-1} G(u,v), \quad k = 0,1,\ldots,l-1 \tag{2-17}
\]

\[
u = i + (d - \frac{w}{2})\cos(o(i,j)) + (k - \frac{l}{2})\sin(o(i,j)) \tag{2-18}
\]

\[
v = j + (d - \frac{w}{2})\sin(o(i,j)) + (k - \frac{l}{2})\cos(o(i,j)) \tag{2-19}
\]

If no minutiae or singular points appear in the oriented window, the frequency of the sinusoidal-shaped wave formed by x-signature in the oriented window is same as the frequency of the ridges and valleys in the oriented window. So the frequency can be estimated via the frequency of x-signature. Compute the average number of pixel between two consecutive peaks of x-signature \( T(x) \), and then the frequency is \( \Omega(x) = 1/T(x) \). If there are no consecutive peaks, the frequency will be set to -1.

4) For the image scanned with certain image resolution, the frequency is fixed in a certain range. For example, with the image resolution of 500dpi, the frequency range is \([1/25, 1/3]\). Compare the estimated frequencies with the fixed range one pixel by one pixel, if the frequency is not in this range, set the frequency of the pixel to -1.
5) In the local neighborhood where minutiae or singular points existing, the ridges and valleys do not formed a sinusoidal-shaped, so the frequency estimated of that local neighborhood is not correct. The incorrect frequencies can be interpolated with the following method [13]:

\[
\Omega'(i, j) = \begin{cases} 
\Omega(i, j) & \text{if } \Omega(i, j) \neq -1 \\
\sum_{u=-w_1/2}^{w_1/2} \sum_{v=-w_2/2}^{w_2/2} W_g(u, v) \mu(\Omega(i-uw, j-vw)) \\
\sum_{u=-w_1/2}^{w_1/2} \sum_{v=-w_2/2}^{w_2/2} W_g(u, v) \delta(\Omega(i-uw, j-vw) + 1)
\end{cases}
\]

(2-20)

where

\[
\mu(x) = \begin{cases} 
0 & \text{if } x \leq 0 \\
1 & \text{otherwise}
\end{cases}
\]

In the function, \(W_g\) is a Gaussian kernel with mean and variance zero and nine respectively. The size of Gaussian kernel is \(w_1 \times w_2\), 7 x 7.

6) Frequency is also change slow in the local neighborhood, so a low-pass filter is need to process the computed frequencies to make them more precisely. The low pass filter can be a 3 x 3 mean filter or a 3 x 3 median filter.

The practical frequency of fingerprint image is shown as figure 2-10.\cite{10}

![Figure 2-10: The frequency of fingerprint image](image)
2.1.5 Region mask

In the reference [9], based on three features of fingerprint, amplitude, frequency and variance of the sinusoidal-shaped ridges and valleys, pixels of fingerprint image can be classified into recoverable and unrecoverable categories. In order to find representative patterns for the two classes, the authors of reference [9] collected 2,000 three-dimensional patterns, and fed then to a squared-error clustering algorithm and identified six clusters. Of the six clusters, four are recoverable, and the other two are unrecoverable. The authors used these six clusters to classify every $N \times N$ block in fingerprint images. If a block centered at $(i,j)$ is recoverable, then $R(i,j)=1$, else $R(i,j)=0$. If the percentage of recoverable blocks is smaller than a pre-specified threshold, then the input fingerprint image is rejected, else is accepted to the subsequent processing.

As mentioned, 2000 three-dimensional patterns are needed to collected, in order to classify fingerprint images. This is time-cost business, and impossible for a time-limited project. But in order to achieve the same effect as the region mask method proposed in reference [9], we use frequency field as a standard to classify the pixel (if the frequency of one pixel is in the range of $[1/25, 1/3]$, the intensity of the pixel is set to 255, else keep the normalized intensity), then use mathematical morphology method to dilate and erode the normalized fingerprint image.

2.1.6 Gabor filter

As mentioned before, Gabor filters are both frequency-selective and orientation-selective band-pass, and also have optimal joint resolution in both spatial and frequency domains. The configuration of fingerprint images has well-defined orientation and frequency in local neighborhood. So it is appropriate to use Gabor filter as a band-pass filter to remove the noise and make ridges and valleys clearer.

The general function of 2-D even-symmetric Gabor filter is shown as following:
\[ h(x, y; \phi, f) = \exp \left( -\frac{1}{2} \left[ \frac{x_x^2}{\delta_x^2} + \frac{y_y^2}{\delta_y^2} \right] \right) \cos(2\pi f x_\phi) \] (2-21)

\[ x_\phi = x \cos \phi + y \sin \phi \] (2-22)

\[ y_\phi = -x \sin \phi + y \cos \phi \] (2-23)

where \( \phi \) is the orientation of the Gabor filter, \( f \) is the frequency of a sinusoidal plane wave, and \( \delta_x \) and \( \delta_y \) are the space constants of Gaussian envelope along \( x \) and \( y \) axes, respectively. The modulation transfer function of the Gabor filter in reference [9] is:

\[
H(u, v; \phi, f) = \exp \left( \frac{1}{2} \left[ \frac{(u - u_0)^2}{\delta_u^2} + \frac{(v - v_0)^2}{\delta_v^2} \right] \right)
\]

\[ u_\phi = u \cos \phi + v \sin \phi \] (2-24)

\[ v_\phi = -u \sin \phi + v \cos \phi \] (2-25)

\[ u_0 = \frac{2\pi \cos \phi}{f} \] (2-26)

\[ v_0 = \frac{2\pi \sin \phi}{f} \] (2-27)

In the function [2-24], \( \delta_u = \frac{1}{2\pi \delta_x} \) and \( \delta_v = \frac{1}{2\pi \delta_y} \). When Gabor filter are applied to a fingerprint image, three parameters must be specified: 1) the frequency of the sinusoidal plane wave, \( f \); 2) the orientation of filters; 3) the standard deviations of the Gaussian envelope, \( \delta_x \) and \( \delta_y \).

It's obvious that the frequency of the filters is determined by the local ridge frequency, and the orientation of the filters is determined by the local ridge orientation. The selection of the value \( \delta_x \) and \( \delta_y \) is a trade-off. The larger the values, the more robust to noise, but the more likely the filters will create spurious ridges and valleys. On the other hand, the smaller the values, the less likely the filters create spurious ridges and valleys; but they will be less robust in removing the noise. As in reference [9], we set the values of \( \delta_x \) and \( \delta_y \) at 4.0 and 4.0 respectively based on the empirical data. As mentioned before, \( G \) is the normalized fingerprint images, \( O \) is the orientation image, \( F \) is the frequency image, \( R \) is the recoverable mask, and the
enhanced image $E$ is obtained as follows:

$$E(i,j) = \begin{cases} 255 & \text{if } R(i,j) = 0 \\ \sum_{u=-w_x/2}^{w_x/2} \sum_{v=-w_y/2}^{w_y/2} h(u,v; O(i,j), F(i,j)) G(i-u, j-v) & \text{otherwise} \end{cases}$$  \hspace{1cm} (2-29)$$

2.2 Fingerprint thinning

After fingerprint enhancement, the clarity of fingerprint ridges and valleys is better, but the width of ridges is not one-pixel, and suitable for minutiae detection. In minutiae detection, the ridges should be one-pixel width. How to make ridges of fingerprint single-pixel width? Fingerprint thinning. What is thinning? Thinning usually involves removing points or layers of outline from a pattern until all the lines or curves are of unit width, or a single-pixel width.\cite{14} The operation of thinning plays an important role in digital image processing and pattern recognition, especially for line drawing and patterns that are irrelevant to their thickness such as character recognition.\cite{15}

A large number of papers proposed thinning algorithms. Based on that whether iterative operation is used, two types exist: 1) non-iterative algorithm, skeleton of fingerprint image is generated at one time, such as the mechanism based on transformation of distance; 2) iterative algorithm, deleting the pixels on the edges of ridges against a set of criteria to decide whether the edge points should be removed or not, and finally the ridge skeleton of one-pixel width is gotten. Most thinning algorithms are iterative.\cite{16} Rosenfeld classified iterative thinning algorithms into two types, parallel and sequential.\cite{17} With parallel algorithms, only the result from previous iteration affects the decision to remove a point in the current iteration, making it suitable for preprocessing by parallel hardware such as an array processor; with sequential algorithms, the result of last previous iteration and the results obtained so far in current iteration decide the processing of current pixel. As developing of classical thinning algorithms, the thinning algorithms developing from morphology are also being developed very fast.

Hilditch, Pavlidis, Rosenfeld thinning algorithms: with these algorithms,
computations are taken in program directly to decide whether or not delete the edge pixels. The differences are different deciding conditions with different computations. Hilditch algorithm are used for binary image, is general algorithm; Pavlidis algorithm is implemented with fusing processings of parallel and sequential operations, and do matching via bitwise computation; Rosenfeld algorithm is a parallel algorithm, used for binary image. The processing performance of Rosenfeld algorithm is better, so this algorithm is selected in this paper.

2.3 Minutiae detecting and fingerprint matching

The goal of minutiae detecting is to extract the minutiae (ending and bifurcation) of fingerprint image, locate them, detecting their frequency, and delete the spurious minutiae; the goal of fingerprint matching is to match the minutiae from input fingerprint image and template image, and determine if they are from the same fingertip.

2.3.1 Minutiae detecting

Many methods of minutiae detecting have been proposed, some detects minutiae directly on gray-scale image,[18] some detects minutiae on skeleton fingerprint image.[19] The gray-scale image based approaches detect the minutiae along the ridges on the gray-scale image directly. The skeleton fingerprint image based approaches convert the original image into skeleton image with image processing algorithms first, and then extract the minutiae on the single-pixel width skeleton fingerprint image. The skeleton image based approaches has less time cost if considering the extraction step lonely, but it needs more image processing steps. It also has higher precision. The gray-scale image based approaches has less time cost compared with skeleton image based approach, if fingerprint processing is considered in skeleton image based approach. But it has lower precision.[20] This paper chooses the skeleton image based method with the consideration of the precision and the independent of feature
As described in figure 2-11 and function (2-30), if the count is equal to 6, the detected minutia is bifurcation; if the count is equal to 2, the detected minutia is ending.\[21\]

Because of fingerprint quality and limitation of fingerprint pre-processing algorithms, some of the true minutiae in fingerprint image are missed while many spurious minutiae are generated. Many papers about minutiae-based approaches have been proposed. Several heuristic rules to eliminate the ridge breaking, spike, and boundary effects, based on the minutiae structural relationship was proposed;\[22\] a w\(\times\)w window around the minutiae were used to delete the spike, island, etc;\[23\] and the duality property of fingerprint image is also used to delete the bridge, ladder and wrinkle. All of these approaches can be used for fingerprint, and solve certain situations. But they are hard to meet the practical fingerprint applications. So a fusing of these methods is necessary. There are several spurious minutiae existing: spike, bridge, island, breaking, short ridge, double bifurcation, and boundary effect, as shown in figure 2-12.

For ridge breaking as shown in figure 2-12 (d), the two detected ridge endings must be in a w\(\times\)w small range which is centered on one of the ridge endings, and because they are generated for the break of on ridge in a local neighborhood, the
orientation of these two endings is smaller than a specified value. After detecting the ridge breaking, the two ridge endings are both deleted.

For bridge as shown in figure 2-12 (b), the two detected bifurcations must in a w*w small range which is centered on one of the bifurcations. So in this paper, a bridge is detected if it satisfies one condition: 1) one bifurcation is in the range of rectangle which is size of w*w (15*15), and centered on another ending. After detecting the bridge, delete both bifurcations. This method can also be used to delete short ridge, as shown in figure 2-12 (e) and figure 2-12 (c).

For spike as shown in figure 2-12 (a), the ending which is detected is at the ending of a short ridge beginning at a bifurcation and ending at then ending. So in this paper, we track the ridges beginning at a bifurcation, and if in the range of 75-pixel length, the ending is a spike. Then delete both the bifurcation as the beginning and ending at the end of the spike.

2.3.2 Fingerprint matching

Fingerprint matching is the core step of fingerprint recognition system, and many researches have been done in this field. D.K.Isenor etc, proposed a matching method based on the graph;[24] Andrew K.Hrecha, etc, proposed a matching method based on fingerprint structure.[25] But now the most developed method is to use minutiae matching based on the model of fingerprint minutiae which was proposed by FBI. As mentioned before, the fingerprint minutiae are fingerprint ridge ending, and bifurcation, and these two factors of fingerprint are used to determine whether two fingerprint images are from the same fingertip. The fingerprint images are represented with point pattern, an automatic fingerprint matching is converted in to a problem of point pattern matching. Point matching is a famous difficult problem in pattern recognition, and many algorithms on point pattern matching have been proposed, such as relaxation algorithm proposed by Sanjay Ranade, etc;[26] triangular fingerprint
matching algorithm proposed by Kovács-Vajna, Z.M.,[27] and matching algorithm based on local and global factors proposed by Xudong Jiang, etc. Anil Jain proposed a matching algorithm in which the minutiae in Cartesian coordinate system are converted into polar coordinates, matching the input fingerprint with template fingerprint via point pattern matching in polar coordinate. Xiping Luo, etc, proposed a matching system by modifying the algorithm proposed by Anil Jain. They modified the algorithm on three aspects: 1) a more effective and simpler fingerprint alignment method was adopted; 2) different from Anil Jain using ridge information during alignment stage, the algorithm of Xiping Luo uses ridge information during matching stage; 3) Anil Jain using a bounding box of fixed size, but the algorithm of Xiping Luo uses a bounding box of adaptive size, so this algorithm can effectively resolve the problem of non-linear distortion. Because of the advantage of the algorithm proposed by Xiping Luo, his algorithm was used in this paper. There are two stages of fingerprint matching proposed by Xiping Luo, fingerprint alignment, and fingerprint matching.[21]

### 2.3.2.1 Fingerprint alignment

Supposed \( P = ((x_1^p, y_1^p, \theta_1^p), \ldots, (x_M^p, y_M^p, \theta_M^p))^T \) represent minutiae of number \( M \) in template fingerprint image, and \( Q = ((x_1^o, y_1^o, \theta_1^o), \ldots, (x_N^o, y_N^o, \theta_N^o))^T \) represents minutiae of number \( N \) in input fingerprint image. In order to convert the minutiae into polar coordinates, a pair of minutiae (one from template minutiae and one from input minutiae) must be located, and compute the polar location in polar coordinates relative to the pair of aligned minutiae.

\( P_i (1 \leq i \leq M) \) is a minutia from template fingerprint image, and \( Q_j (1 \leq j \leq N) \) is a minutia from input fingerprint image, \( \text{rotate}[i][j] \) is a rotating angle from input image to template image if considering \( P_i (1 \leq i \leq M) \) and \( Q_j (1 \leq j \leq N) \) as a pair of aligned minutiae. If they are regarded as a pair of minutiae, the related ridges with them are similar enough, then \( \text{rotate}[i][j] \) is a value between \( 0^\circ \sim 360^\circ \), else \( \text{rotate}[i][j] \) is set to 400 to express that they can not be regarded as a minutiae pair.

If \( P_i (1 \leq i \leq M) \) and \( Q_j (1 \leq j \leq N) \) are the same type of fingerprint minutiae, the points abstracted on the related ridges are same, then
whether \( P_i (1 \leq i \leq M) \) and \( Q_j (1 \leq j \leq N) \) are aligned fingerprint minutiae and the value of \( \text{rotate}[i][j] \) are decided by the following computation.\(^{[21]}\)

\[
\begin{align*}
\text{Diff}_{\text{dist}} &= \frac{1}{L} \sum_{i=0}^{L} \left| R(d_i) - r(d_i) \right| \\
\text{Diff}_{\text{ang}} &= \frac{1}{L} \sum_{i=0}^{L} \left| R(\alpha_i) - r(\alpha_i) \right|
\end{align*}
\]

\[(2-31)\]

In function (2-31), \( R \) is the ridge connected with \( P_i \), \( r \) is the ridges connected with \( Q_i \). Function (2-31) represents the difference between the two ridges. \( L \) is the number of points sampled at the ridges, \( R(d_i) \) and \( r(d_i) \) represent the distance from point \( i \) to the relative minutiae, and \( R(\alpha_i) \) and \( r(\alpha_i) \) represent the orientation differences between point \( i \) to the relative minutiae, as shown in figure 2-13.\(^{[21]}\)

![Figure 2-13: Alignment of input ridge and template ridge](image)

If \( \text{Diff}_{\text{dist}} \) is smaller that a specified threshold \( T_d \), and \( \text{Diff}_{\text{ang}} \) is smaller than a specified threshold \( T_a \), we can take them as a pair of ridges which are similar in shape, and \( P_i \) and \( Q_i \) are a pair of minutiae, then \( \text{rotate}[i][j] \) can be estimated with following function:

\[
\text{rotate}[i][j] = \text{dir}_\text{temp} - \text{dir}_\text{in}
\]

\[(2-32)\]

In the function (2-32), \( \text{dir}_\text{temp} \) is the orientation of template minutiae \( P_i \) and \( \text{dir}_\text{in} \) is the orientation of input minutiae \( Q_i \). If \( \text{Diff}_{\text{dist}} \) is larger that a specified threshold \( T_d \), or \( \text{Diff}_{\text{ang}} \) is larger than a specified threshold \( T_a \), the two ridges are
not a pair of ridges, the two minutiae are not a pair of minutiae, then \( \text{rotate}[i][j] \) will be set to 400.

After the relative minutiae found from template image and input image, then just convert the minutiae in template image and input image into polar coordinates, which are centered on minutiae \( P_i \) and \( Q_j \) respectively. The following function shows the way of converting:[28]

\[
\begin{pmatrix}
    r_i \\
    e_i \\
    \theta_i
\end{pmatrix} = \begin{pmatrix}
    \sqrt{(x_i - x')^2 + (y_i - y')^2} \\
    \tan^{-1}\left(\frac{y_i - y'}{x_i - x'}\right) \\
    \theta_i - \theta'
\end{pmatrix}
\] (2-33)

In function (2-33), \( (x_i, y_i, \theta_i)^T \) is the minutiae coordinate, which is going to be converted into polar coordinates; \( (x', y', \theta')^T \) is the referenced minutiae coordinate; \( (r_i, e_i, \theta_i)^T \) is the minutiae coordinate represented in polar coordinates. In \( (r_i, e_i, \theta_i)^T \), \( r_i \) is polar radius, \( e_i \) is polar angle, and \( \theta_i \) is the orientation between minutiae and referenced minutiae. In order to rotate the input fingerprint image to the same orientation as template fingerprint image, \( \text{rotate}[i][j] \) should be added into every \( e_j \).

2.3.2.2 Fingerprint matching:

The matching steps which this paper used and are proposed in reference [21] are discussed following:

(1) For every \( i(1 \leq i \leq M) \) and every \( j(1 \leq j \leq N) \), if \( \text{rotate}[i][j] = 400 \), which means minutiae \( P_i \) and \( Q_j \) can not be regarded as minutiae pair, step (1) should be repeated; else if \( \text{rotate}[i][j]! = 400 \), then go to step (2). If all of minutiae have been done in step (1), then go to step (5).

(2) Using minutiae \( P_i \) and \( Q_j \) as referenced minutiae in template image and input image respectively, convert the minutiae coordinate of template image and input image into polar coordinate centered on \( P_i \) and \( Q_j \) respectively.

(3) Sort the minutiae of template image and input image in polar coordinates based on increasing order of polar angle respectively, and respect in
sequence, as shown in following:

\[
P_i^r = ((r_1^p, e_i^p, \theta_1^p)^\top, ..., (r_M^p, e_M^p, \theta_M^p)^\top),
\]

(2-34)

\[
Q_j^o = ((r_1^o, e_j^o, \theta_1^o)^\top, ..., (r_N^o, e_N^o, \theta_N^o)^\top),
\]

(2-35)

(4) Using the later discussed matching method to match \( P_i^r \) and \( Q_j^o \), find and record the minutiae matching score \( m_{score}[i][j] \), then go back to step (1).

(5) Find the largest \( m_{score}[i][j] \), which is regarded as matching score of template image and input image. If it is larger than a specified threshold, then the template image and the input image are considered from the same fingertip, else from different fingertips.

Before discussing matching method, a bounding box and its size should be introduced first, as shown in figure 2-14. A bounding box is a box on a minutia, and the size of bounding box is defined by \( \text{radius} \_ \text{size} \) and \( \text{angle} \_ \text{size} \).

![Figure 2-14: Bounding box of fixed size and changeable size](image)

In reference [21], a bounding box of changeable size is proposed, which indicates that the values of \( \text{radius} \_ \text{size} \) and \( \text{angle} \_ \text{size} \) are changeable with the value of polar radius of minutiae. If the polar radius of minutiae is longer, \( \text{radius} \_ \text{size} \) of the bounding box is longer and \( \text{angle} \_ \text{size} \) of the bounding box is smaller; if the polar radius of minutiae is shorter, \( \text{radius} \_ \text{size} \) of the bounding box is shorter and \( \text{angle} \_ \text{size} \) of the bounding box is larger, as shown in figure 2-14. The \( \text{radius} \_ \text{size} \) and the \( \text{angle} \_ \text{size} \) of minutiae with radius \( r \) can be computed with following functions:
In functions from (2-36) to (2-39), $r$ is the radius of template minutiae, $r_{\text{small}}$, $r_{\text{large}}$, $a_{\text{small}}$, $a_{\text{large}}$ are the upper boundary and lower boundary of $radius\_size$ and the $angle\_size$ respectively, and $\alpha$ is pre-specified constant value.

In order to make matching algorithm more robust to non-linear distortion, the bounding box of changeable size is used, not the bounding box of fixed size. Non-linear distortion is very obvious in some region, and non-linear to expand out. When the radius of minutiae is shorter, small distortion can cause large distortion of polar angle, and small distortion of polar radius. So in this situation, the $angle\_size$ of bounding box should be larger, and the $radius\_size$ of bounding box should be shorter. On the other hand, when the radius of minutiae is longer, small distortion can bring small distortion of polar angle, and large distortion of polar radius. So in this situation, the $radius\_size$ of bounding box should be longer, and the $angle\_size$ of bounding box should be smaller, as shown in figure 2-14.[21]

The algorithm of matching $P_i'$ and $Q_j'$ is discussed in following section:

(1) The size of bounding box of every template minutiae is computed by functions (2-36)~(2-39).

(2) The following round loop is taken:

While $1 \leq k \leq M$ do

While $1 \leq l \leq N$ and $r_{\text{low}} < angle\_high[k]$ do

if template \_ point$[k]$ and input \_ point$[l]$ satisfy condition1, then

$m\_score[i][j] = m\_score[i][j]+1$;

Adjust the size of bounding box;
end if
Increase \( l \);
end while
Increase \( k \);
end while

The condition \( 1 \) in upper section is:

\[
condition 1 = \begin{cases} 
    \text{true} & \text{if } \left\{ \begin{array}{l}
    \text{radius}_\text{low}[k] < (r^p_i - r^0_i) < \text{radius}_\text{high}[k] \\
    \text{radius}_\text{low}[k] < \Delta \theta < \text{radius}_\text{high}[k] \\
    \text{rotate}[k][l] < 400
    \end{array} \right.
    \\
    \text{false} & \text{otherwise}
\end{cases}
\tag{2-40}
\]

\[
\Delta \theta = \begin{cases} 
    a & \text{if } (a = (\theta^p_i - \theta^0_i + 360) \mod 360) < 180 \\
    a - 180 & \text{otherwise}
\end{cases}
\tag{2-41}
\]

\[
\Delta \theta = \begin{cases} 
    a & \text{if } (a = (\theta^p_i - \theta^0_i + 360) \mod 360) < 180 \\
    a - 180 & \text{otherwise}
\end{cases}
\tag{2-42}
\]

During implement matching algorithm from reference [21], we found that for minutiae \( \text{template}_p\text{oint}[k] \), there may be more than one \( \text{input}_\text{oint}[l] \) can be matched with it, but in practical, it is impossible that one minutia from template image can be matched with more than one minutia from input image. Also the time of matching computation is wasted for more than one minutia is matched. So we modified the matching algorithm in two aspects: 1) compare the numbers of template minutiae \( M \), and input minutiae \( N \), find the larger one \( \text{Larger} \) and smaller one \( \text{Smaller} \), the larger one named as \( \text{Larg}_p\text{oint} \) related with \( P^i \) or \( Q^j \), and the smaller one named as \( \text{Small}_p\text{oint} \) related with \( P^s \) or \( Q^s \); 2) modify the round loop as following sequence:

While \( 1 \leq k \leq \text{Smaller} \) do

While \( 1 \leq l \leq \text{Larg}_p\text{oint} \) and \( \text{Smaller}, \angle_{\text{high}}[k] < \angle_{\text{low}}[l] \) do

if \( \text{Small}_p\text{oint}[k] \) and \( \text{Larg}_p\text{oint}[l] \) satisfy condition \( 1 \), then

\[
m_{\text{score}}[i][j] = m_{\text{score}}[i][j] + 1;
\]

adjust the size of bounding box;
break;
end if
Increase \( l \);
end while

Increase $k$;

end while

break the second level round loop, once there is minutia matched with $Small\_point[k]$. Once $Small\_point[k]$ is matched with $Large\_point[l]$, the second round loop is broken, and $Small\_point[k]$ does not need to be matched with $Large\_point[l+1] \sim Large\_point[larger]$, the comparing time is saved. The smaller minutiae number as the number of outer round loop, the whole round number is saved, so the computation save is saved. With these two modified aspects, the efficiency of matching algorithm is improved.
Chapter 3: Security authentication based on cryptology mechanism

As the fast developing of wireless network, security authentication is becoming more critical than in wire network. Cryptography is becoming increasingly important mechanism of security authentication, with the proliferation of information exchange across the wireless network, and the storage of sensitive data on open networks. Many cryptographic algorithms are available for securing information, such as RC4, RSA, AES, SMS4, etc. In general, data will be secured using a symmetric cipher system; in this paper shared-key system was used.

As mentioned before, biometric authentication offers a new mechanism for security network authentication. However, biometric image is only a cluster of bits, there is no essentially different if compared it with security key, from the point of view of computer. So in wireless network, fingerprint recognition is not safe without protection: 1) replay attack, attacker intercepts the fingerprint information, and send it to authentication server directly while bypassing fingerprint capture equipment, and fingerprint processing and matching; 2) attack to fingerprint template storied on authentication server, attacker is vicious to attack the database on authentication server to obtain fingerprint template information; 3) attack to fingerprint template storied on authentication client, attacker steal the authentication device of client to obtain fingerprint template information.\[29]\n
In this paper, fingerprint verification and shared-key authentication are merged to resolve the upper issues existing in fingerprint recognition system.

3.1 Shared –key authentication

Shared-key authentication seeks to authenticate stations as either a member of those who know a shared secret key or member of those who do not. Shared-key authentication can be used if and only if WEP has been selected in WLAN.\[7\] In following discussions, the station initiating the authentication exchanges is referred to
as the requester, and the station to which the initial frame in the exchange is addressed is referred to as the responder.[7]

From figure 1-13, in chapter one, we can see that shared-key authentication has four frames: authentication request, challenge number, response, and success/fail.

3.1.1 First frame of shared-key authentication

During this stage, the authentication message is sent from requester to responder, which is actually an authentication initializing message. The information of the message includes: 1) Station identity assertion; 2) Authentication algorithm identification = “Shared Key”; 3) Authentication transaction sequence number = 1; 4) Authentication algorithm dependent information (none).[7]

3.1.2 Second frame of shared-key authentication

Before sending back the second frame to requester in the shared-key authentication sequence, the responder generates a string of octets as authentication text, and verifies the first frame from requester.

If verification is not successful, the following information items are sent back to requester as second frame: 1) Authentication algorithm identification = “Shared Key”; 2) Authentication transaction sequence number = 2; 3) Authentication algorithm dependent information = the authentication result; 4) The unspecified challenge text. If verification is successful, the following information items are sent back to requester as second frame: 1) Authentication algorithm identification = “Shared Key”; 2) Authentication transaction sequence number = 2; 3) Authentication algorithm dependent information = the challenge text; 4) 128-octet authentication result.[7]
3.1.3 Third frame of shared-key authentication

Before sending third frame to responder, the requester copies challenge text from second frame into third frame, and then encapsulates the third frame via authentication algorithm.

The information items sent from requester to responder include: 1) Authentication algorithm identification = “Shared Key”; 2) Authentication transaction sequence number = 3; 3) Authentication algorithm dependent information = The challenge text from the second frame.[7]

3.1.4 Final frame of shared-key authentication

After receiving the final frame of shared-key authentication, the responder decapsulate the third frame, and compare the decrypted challenge text with the challenge text sent in the second frame. If they are same, the responder will response with successful code, else unsuccessful code.

The items of final frame sent from the responder to the requester include: 1) Authentication algorithm identification = “Shared Key”; 2) Authentication transaction sequence number = 4; 3) Authentication algorithm dependent information = The authentication result; 4) The result code of the requested authentication, which is a fixed length item with value “successful” and “unsuccessful”.

3.2 Authentication algorithm

In this paper, we modified shared-key authentication process, which will be discussed in next chapter, and used SMS4 cryptograph algorithm instead of RC4 algorithm.

SMS4 algorithm is the first Chinese encryption standard for WLAN released by
the Office of State Commercial Cryptography Administrator (OSCCA). It is a symmetric-key cipher using Feistel net structure. The data block length and the key length are both fixed 128 bits. The encryption algorithm and decryption algorithm both have 32-round non-linear iterative structure, as shown in figure 3-1, but the sequence of round keys involved in computation is reversed.

Assume that the input plain text is \((X_0, X_1, X_2, X_3)\), the output cipher text is \((Y_0, Y_1, Y_2, Y_3)\), and the round key is \(r_{k_i}, i = 0, 1, 2, ..., 31\), where the word length of \(X, Y, r_k\) is 32 bits. The encryption transformation is:

\[
X_{i+4} = F(X_i, X_{i+1}, X_{i+2}, X_{i+3}, r_{k_i}) \\
= X_0 \oplus T(X_{i+1} \oplus X_{i+2} \oplus X_{i+3} \oplus r_{k_i}), i = 0, 1, ..., 31
\]  

(3-1)

\((Y_0, Y_1, Y_2, Y_3) = (X_{35}, X_{34}, X_{33}, X_{32})\)

As mentioned above, the encryption and decryption use reverse-sequence round keys. The round-key sequence for encryption is \((r_{k_0}, r_{k_1}, ..., r_{k_{31}})\), and the round-key sequence for decryption is \((r_{k_{31}}, r_{k_{30}}, ..., r_{k_{0}})\).

![Figure 3-1: SMS4 encryption flow chart](image)

The combined transformation \(T\) is a reversible transformation, combining one non-linear transformation \(\tau\) and one linear transformation \(L\), \(T(.) = L(\tau(.))\). The non-linear transformation is composed by 4 parallel S-box. The output of S-box is 8 bits with 8-bit input. The \(\tau\) transformation can be given as:
\[ B = \tau(A) = (Sbox(a_0), Sbox(a_1), Sbox(a_2), Sbox(a_3)) \] (3-2)

where \( A = (a_0, a_1, a_i, a_j) \), and \( a_0 - a_3 \) are 8-bit data. The \( L \) transformation can be given as

\[ L(B) = B \oplus (B \ll 2) \oplus (B \ll 10) \oplus (B \ll 18) \oplus (B \ll 24) . \] (3-3)

The round keys \((rk_0, rk_1, ..., rk_{31})\) used in the encryption and decryption described above are generated in the key expansion schedule. The process of key expansion can be divided into 32 rounds which have the similar structure as those in encryption process as shown in figure 3-1, except for the linear transformation

\[ L'(B) = B \oplus (B \ll 13) \oplus (B \ll 23) . \] (3-4)
Chapter 4: Proposed hardware-software cooperating embedded verification system

As mentioned before, an authentication system merging fingerprint authentication and shared-key authentication is proposed. Shared-key authentication was modified in our system, which will be discussed later. The proposed system was implemented with both software and hardware-software cooperation.

4.1: Software implementation of embedded authentication system

In order to easily estimate the embedded authentication system, which we proposed and the fingerprint authentication algorithms we selected, the embedded authentication system implemented with software was designed firstly before software-hardware cooperating embedded authentication system.

The embedded authentication system includes two steps: 1) fingerprint enrollment; 2) security authentication. Fingerprint verification is processed at client end, and the selected algorithms have been discussed in last chapter.

4.1.1 Fingerprint enrollment

The enrollment process was shown in figure 4-1, the shown steps are: 1) Firstly,
client gets the initial secret key, *KeyID*, *SQN*, *IDD*, *IDS*; 2) Client sends the *KeyID/SMS40* (*SQN*/*IDD*/*IDS*) to server; 3) After server receives *KeyID*, it looks up the secret key of SMS4 from the key table, decrypts the cipher text *SMS40* (*SQN*/*IDD*/*IDS*), gets *SQN*, *IDD*, *IDS*, and then verifies the correctness of them; 4) Server sends the verifying result to client end, and if the result is “successful”, client device will scan fingerprint image, extract minutiae, and generate template, else do nothing. So the client end embedded device must include fingerprint sensor to scan fingerprint, and have capacity to process the scanned fingerprint image. In figure 4-1, *KeyID* is the *ID* of initial secret key; initial *SQN* is the initial enrollment sequence number; *IDD* is the *ID* of client; *IDS* is the *ID* of server; *SMS40* means that SMS4 encryption using the initial secret key.

During the enrollment process, client has to get the correct secret key, *KeyID*, initial *SQN*, *IDD*, *IDS*, and completes the enrollment process after the authentication of server. So if the device of client embedded device (for example the credit card with fingerprint authentication module) is stolen, there is no necessary to worry about the re-enrollment which may cause financial loss.

### 4.1.2 Fingerprint authentication

Figure 4-2 shows the authentication process of proposed authentication system. Supposed that client has gotten the embedded authentication device and the enrollment process has been finished. If client needs to access the remote server, the authentication steps are:
1) Client sends $\text{KeyID}/\text{SMS40} (SQN/IDD/IDS)$ to server, where $SQN$ is the sequence number of current time. $SQN$ is generated randomly at every time of authentication, not a number increasing order by order, so it is not easy to figure the rule of $SQN$, not easy to guess, and the security of the system is improved. $SQNs$ of client end and server end are generated with the same seed and random code generation algorithm, and synchronized at enrollment stage.

2) Server looks up the secret key from the secret key table related with the client based on the $\text{KeyID}$, decrypt the cipher text of $SQN$, $IDD$, and $IDS$, and compare the decrypted $SQN$, $IDD$, and $IDS$ with the local $SQN$, $IDD$, and $IDS$. If the result of comparision is successful, server will generate 128-bit rand code of current authentication, RAND; else current authentication is over, and authentication is fail. $\text{Message\_key}$, and $\text{hash\_key}$, which are derived from $RAND$ is generated, if verification is successful. The rule of generating $\text{message\_key}$ and $\text{hash\_key}$ is shown as functions (47) and (48), where 0 and 1 mean 128-bit 0 vector and 128-bit 1 vector respectively, and $\text{Message\_key}$ is the secret key for encryption and decryption, and $\text{hash\_key}$ is the hash key for hash verification.

After that, sever send $RAND/\text{Hash1(SQN/RAND/IDD/IDS)}$ to client. $\text{Hash1}$ is hash encryption using $\text{hash\_key}$ via SMS4.
3) Client generates `message_key` and `hash_key` using functions (4-1) and (4-2) based on `RAND`, and then verify the receiving `SQN`, `RAND`, `IDD`, `IDS` via hash verification. If hash verification is successful, the embedded authentication device at client end is beginning to scan fingerprint image, extracting minutiae, and matching input minutiae with template minutiae; and then sending $\text{SMS41(result/SQN)/Hash1(result/SQN)}$ to server; else the embedded device do nothing to end current authentication. With this step, the client end can also authenticate the server end, so the risk of attack from fake server is reduced. During this step, $\text{SMS41}$ means encryption which uses `message_key` to encrypt via SMS4; result is the matching core of fingerprint verification.

4) Server decrypts the cipher text of result, and `SQN`, and then verify the integrity of `result/SQN` via hash verification. If verification is successful, server will compare the result with a pre-specified threshold to determine whether the client is legal or not, and then send “pass”, or “deny” to client end; else end the authentication process and inform client end authentication failed.

5) The client end gets the authentication result from server to access the server.

This authentication mechanism can improve currently widely used authentication mechanism, which merges the authentication mechanisms of user-password and physical medium (bank card), by adding fingerprint recognition module to client end. In this way, the security of network authentication will be improved significantly.

4.2: Software-hardware cooperating implementation

As shown in table 1-2 in chapter 1, the computation time of orientation and frequency field, and Gabor filter take most of the enhancement time. Also as shown in table 4-1, the time cost of enhancement takes most of the fingerprint recognition time. To improve efficiency of fingerprint recognition, one of most useful ways is to
implement the time-cost part in ASIC (Application Specific Integrated Circuit).

Table 4-1: The time of fingerprint enhancement

<table>
<thead>
<tr>
<th>Enhancement (seconds)</th>
<th>Thinning (Seconds)</th>
<th>Matching (Seconds)</th>
<th>Total (Seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.218</td>
<td>0.375</td>
<td>0.469</td>
<td>9.062</td>
</tr>
</tbody>
</table>

4.2.1 Introduction of ASIC

An ASIC is basically an integrated circuit designed specifically for a special purpose of application. Strictly speaking, it implies that an ASIC is built only for one and only one customer. ASICs are usually classified into one of three categories: full-custom, semi-custom, and structured.

Full-custom ASICs are those that are entirely tailor-fitted to a particular application from the very start. Since its ultimate design and functionality is pre-specified by the user, it is manufactured with all the photolithographic layers of the device already fully defined, just like most off-the self general purpose ICs. The use of predefined masks for manufacturing leaves no option for circuit modification during fabrication, except perhaps for some minor fine-tuning or calibration.[28]

Semi-custom ASICs, on the other hand, can be partly customized to serve different functions within its general area of application. Unlike full-custom ASICs, semi-custom ASICs are designed to allow a certain degree of modification during the manufacturing process. A semi-custom ASIC is manufactured with the masks for the diffused layers already fully defined, so the transistors and other active components of the circuit are already fixed for that semi-custom ASIC design.[28]

Structured or platform ASICs, which are a relatively new ASIC classification, are those which have been designed and produced from a tightly defined set of: 1) design methodologies; 2) intellectual properties (IPs); 3) well-characterized silicon, aimed at shortening the design cycle and minimizing the development costs of ASIC.[28]
4.2.2 The flow of ASIC design

There are three steps of ASIC design flow as shown in figure 4-3. From figure 4-3, we can see that the first step of ASIC design is to do research and plan on the whole and in detail.

4.2.3 System design

The whole system includes three modules, fingerprint enhancement, fingerprint thinning, and fingerprint matching. The structure of software-hardware cooperating system is shown in figure 4-4. As shown in figure 4-4, the time-cost parts, orientation, frequency, and Gabor filter, are implemented in ASIC in order to improve the efficiency of the embedded authentication system. The part of region mask is also implemented in ASIC in order to decrease the communication between software part
and hardware part. The communication time is saved, the efficiency is further improved.

Figure 4-4: The structure of embedded authentication system

4.2.4 Orientation field estimation

In figure 4-5, “orientation control” is the control module; “ram_0” is the ram to save fingerprint image after normalization; “gradient” is the module to compute the gradient of fingerprint image; “ram_1” is the ram to save the gradient of x direction; “ram_2” is the ram to save the gradient of y direction; “lori_addr” is the module to compute the address for look up table of rom_0 and rom_1; rom_0 is the rom to save sin(x) value between 0° ~ 90°; rom_1 is the rom to save cos(x) value between 0° ~ 90°; “lori” is the module to convert the sin(x) and cos(x) value between 0° ~ 90° to 0° ~ 360° based on the signal phase that have been estimated in the module, “lori”; “ram_3” is the ram to save the value of sin(x) between 0° ~ 360°; “ram_4” is the ram to save the value of cos(x) between 0° ~ 360°; “low pass” is a low pass filter module to reduce the noise of frequency, and to compute the address for rom_2; “rom_2” is a rom to save the theta value between 0° ~ 45°; “renew_ori” is the module to convert the value between 0° ~ 45° to 0° ~ 360° based on the value of
“phase_ori”, which is computed in the module, “low pass”.

As mentioned in orientation field estimation of fingerprint enhancement algorithm, we did not estimate the local orientation of fingerprint image before low pass filter in this paper, but the $\sin(x)$ value and $\cos(x)$ value for orientation low pass processing and between $0^\circ$ ~ $90^\circ$ were estimated via look up table. If the local orientation of fingerprint image before low pass filter was estimated instead of $\sin(x)$ value and $\cos(x)$ value, the modules to compute $\tan^{-1}$ and estimate the address for looking up the values of $\sin(x)$ and $\cos(x)$ are needed, but in the way of this paper, the module to compute $\tan^{-1}$ is saved, the hardware cost is lowered. More than that, the module to compute $\tan^{-1}$ is very difficult to implement in VLSI, and time cost of process is long, so the time cost of this system is shortened.

Figure 4-5: The structure or orientation field estimation module
Why the $\sin(x)$ and $\cos(x)$ values between $0^\circ \sim 90^\circ$ is saved in the look up table, rom_0 and rom_1, not the values between $0^\circ \sim 360^\circ$? Why the $\tan^{-1}$ values between $0^\circ \sim 45^\circ$ is saved in the look up table, rom_3, not the values between $0^\circ \sim 360^\circ$? The answers to the two questions are same. Take the example of sin values: 1) $\sin(x)$ is a periodic function, the period is $2\pi$ as shown in figure 4-6(a); 2) and $|\sin(x)|$ is also a periodic function, the period is $\pi$ as shown in figure 4-6(b); 3) the value of $|\sin(x)|$ in one period is symmetrical with the middle vertical axis in that period, as shown in figure 4-6(b). For the discussed three reasons, just the values of a quarter of $\sin(x)$ in one period are enough for estimating the value of $\sin(x)$. The steps to estimate the values of $\sin(x)$ are: 1) estimating the angle $x$ in which quadrant, and record the information of quadrant in the register of phase[1:0]; 2) converting the angle $x$ which is not in the first quadrant to the first quadrant $x_{\_first}$, and compute the address $addr_{\_theta}[14:0]$ to look up the $\sin(x_{\_first})$; 3) converting the looked up value $\sin(x_{\_first})$ back to $\sin(x)$ based on phase[1:0]. In this way the rom space to save the value of $\sin(x)$ for looking up was saved by three quarters.

With $\tan^{-1}(x)$, we know that it is impossible to save all of the $\tan^{-1}(x)$ values in look up table. In this paper, we used one of properties of $\tan(x)$, in function (4-3), and also the periodic properties of $\tan(x)$. The steps are: 1) converting the angle $x$ which is not between $0^\circ \sim 45^\circ$ to $0^\circ \sim 45^\circ$, and recording the phase information in the register phase_\_ori[2:0]; 2) looking up the value of $\tan^{-1}(x)$ between $0^\circ \sim 45^\circ$; 3) converting the looked up angle back to the original quadrant based
on $phase_{ori}[2:0]$. In this way the rom space to save $\tan^{-1}(x)$ was saved significantly.

$$
\tan(x) = \frac{1}{\tan\left(\frac{\pi}{2} - x\right)} \tag{4-3}
$$

4.2.4.1 Gradient Module

In order to accelerate the computation, intensities of four pixels in one row are saved in one address memory, and four intensities can be gotten in one clock, so the time to fetch the intensities is decreased; after get the intensities, the pipeline structure to compute the gradient also accelerates the computation, as shown in figure 4-7.

![Figure 4-7: The pipeline structure of gradient module](image)
As shown in figure 4-7, Sobel_X and Sobel_Y is the module to calculate the gradients of x direction and y direction; reg0 ~ reg5 are the registers to save the input data; Gradient_Xreg and Gradient_Yreg are the registers to save the gradients of x direction and y direction. The pipeline process is:

(1) Initialize the registers of reg0, reg2, and reg4.

(2) Compute the first gradients of x direction and y direction, the input is shown in figure 4-8, and then save the outputs to Gradient_Xreg[47:36], and Gradient_Yreg[47:36]; read next intensities to reg1[31:0].

(3) Compute the second gradients of x direction and y direction, the input is shown in figure 4-9, and then save the outputs to Gradient_Xreg[35:24], and Gradient_Yreg[35:24]; read next intensities to reg3[31:0].

(4) Compute the third gradients of x direction and y direction, the input is shown in figure 4-10, and then save the outputs to Gradient_Xreg[23:12], and Gradient_Yreg[23:12]; read next intensities to reg5[31:0].
(5) Compute the fourth gradients of x direction and y direction, the input is similar to the input of the first and second but move right by 8 bits, and then save the outputs to Gradient_Xreg[11:0], and Gradient_Yreg[11:0]; save the values of Gradient_Xreg[47:0] and Gradient_Yreg[47:0] to ram_1 and ram_2 respectively; read the next intensities to reg0[31:0].

(6) Compute the fifth gradients of x direction and y direction, the input is similar to the input of the first and second but move right by 8 bits, and then save the outputs to Gradient_Xreg[47:36], and Gradient_Yreg[47:36]; read the next intensities to reg2[31:0].

(7) Compute the sixth gradients of x direction and y direction, the input is similar to the input of the first and second but move right by 8 bits, and then save the outputs to Gradient_Xreg[35:24], and Gradient_Yreg[35:24]; read the next intensities to reg4[31:0].

(8) Back to step (2).

From the description, we can see that with size clocks, we can calculate size pair of gradients, gradient_x and gradient_y, and the rate is $1\text{pair/clock}$; if we compute the gradients one by one, the time to compute is the time to read the input, 3 clocks, so the rate is $1/3\text{pair/clock}$. With analyze, the efficiency is improved by 2 times with the pipeline structure.

4.2.5 Frequency field estimation

In this module, “frequency control” is the control module; “ram_5” is the ram to save the orientation of fingerprint image; “sin_cos_addr” is used to compute the address to look up the values of $\sin(o(i, j))$ and $\cos(o(i, j))$ used in the functions, (2-18) and (2-19); “sin_rom” and “cos_rom” are the roms to look up the values
of \sin(o(i, j)) and \cos(o(i, j)), respectively; “sub_xsig_add0” and “sub_xsig_add1” are the modules to compute the address to look up the \(G(u, v)\) used in the function, (2-17); “ram_00” and “ram_11” are the rams the save the normalized intensity of fingerprint pixel, and one cell of “ram_00” and “ram_11” is size of 32 bits; “x_signature0” and “x_signature1” are the modules to get \(G(u, v)\) which is 8-bit from the 32-bit value looked up from “ram_00” and “ram_11” and add the 16 \(G(u, v)\)s to get \(X[k]\) as in function (2-17); “frequency0” and “frequency1” are the modules to estimate the frequency using the method mentioned in the section 2.1.4.; because two frequencies are computed synchronously, the aim of “sort_fre” is to sort the two frequencies in order to save them one by one; the goal to “low_pass” is to filter to noise of high frequency, because the frequencies of fingerprint image changes slowly in local neighborhood.

![Figure 4-11: The structure of frequency field estimation module](image)

The frequency estimation is a time-cost module as mentioned, so in this paper, two frequencies are parallel to estimated, the time is saved by half, as shown in figure 4-11. There are two frequencies of fingerprint image in one cell of ram_5, so the “sin_cos_addr” module can estimate two addresses at the same time,
“sub_xsig_add0”, “ram_00”, “x_signature0”, “frequency0” are the first modules to compute the frequency; “sub_xsig_add1”, “ram_11”, “x_signature1”, “frequency1” are the second modules to compute the frequency.

4.2.5.1 sin_cos_addr module

In order to save the LUT table memory, “sin_rom” and “cos_rom”, just the values of sin(x) and cos(x) are saved in the roms, however the orientations of fingerprint image are \((-\pi/2, \pi/2]\), so the modules to convert the original orientations to \((0, \pi/2]\) and recover the looked up values are needed. The structure of “sin_cos_addr” is shown in figure 4-12, and there are two outputs of this module, one is the address to look up values; the other is the phase of fingerprint orientation to recover the real value.

![Figure 4-12: The structure to compute sin and cos address](image)

4.2.6 Gabor filter
The detail process of Gabor filter computation module is shown in figure 4-13. Before discussing figure 4-13, the mathematical principle of Gabor filter computation will be discussed firstly. The function of Gabor filter is shown as (2-29), and from the function, we can see that the critical step of this filter is to compute the filter coefficients of pixel intensities. The functions to calculate the filter coefficients are shown as (2-21)~(2-23). As mentioned in this paper, the values of $x_8$ and $y_8$ were set to 4.0 and 4.0. Then the functions to calculate the filter coefficients can be simplified as the following steps:

\[
\frac{x^2}{16} + \frac{y^2}{16} = \frac{(x \cos \phi + y \sin \phi)^2 + (y \cos \phi - x \sin \phi)^2}{16}
\]

\[
= \frac{x^2 + y^2}{16}
\]  

(4-4)
The frequency $f$ and the location of pixel $(u,v)$ is easy to get compared with the values of $\cos \phi$ and $\sin \phi$, so the values of $fu$ and $fv$ used in $2\pi fu \cos \phi + 2\pi fv \sin \phi$ are computed firstly; the exponential computation is also simplified to have no relationship with orientation to make the computation much easier. The VLSI implementation of Gabor filter is based on the simplified format of the Gabor filter functions, and the structure is shown in figure 4-13. From figure 4-13, we can see that the computation is pipelining, including four stages: 1) the first stage, Stage0, is to get the values of $\cos \phi$ and $\sin \phi$, compute the values of $fu$ and $fv$, and the time cost is one clock; 2) the second stage, Stage1, is to compute the values of $2\pi fu \cos \phi + 2\pi fv \sin \phi$ via multiplier, and the time cost of the multiplier is two clocks; 3) the third stage, Stage2, is to compute the value of $\cos(2\pi f_x \phi)$ via LUT, and the time cost is one clock; 4) the forth stage, Stage3, is to compute the value of $\exp\left(-\frac{u^2 + v^2}{32}\right) \cos(2\pi f_x \phi)G(i-u, j-v)$ via a multiplier, and the time cost of the multiplier is two clocks. The module to compute the value of $\exp\left(-\frac{u^2 + v^2}{32}\right)G(i-u, j-v)$ is a special module: 1) it is timing parallel with Stage0 and Stage1, and the time cost is two clocks; 2) the input signal to this module is late by one clock compared the signals to Stage0; 3) in order to make the output of this module reach Stage 3 at the same time with the output of Stage2, a register is added between the output of this module and the input of Stage3. The multipliers used in Stage1, Stage2 and Stage3 are Booth Encoder multiplier, and it is a pipeline multiplier, including Booth Encoder and Wallace Tree, the time costs of this multiplier is two clocks, as shown in figure 4-14. So the pipeline flow of Gabor filter can be constructed as figure 4-15.
Figure 4-14: The structure of multiplier

Figure 4-15: The timing sequence of Gabor filter
Chapter 5: Metric and security analysis

The software part has been implemented with C language, and the hardware part has been implemented with VLSI. The purpose of this system is to achieve high verification rate, efficiency and resolve some security issues existing in the current authentication systems. From the following performance evaluation and theory analyze, we can see that this system does achieve the goals.

5.1: Metric of authentication system

The test system was tested on the platform formed by PC as server and FPGA development board as client, as shown in figure 5-1 and figure 5-2. The system on the FPGA board is RISC processor of MIPS structure, the software is running on the RISC-D4, the hardware part is running as a cooperating processor, “Enhancement”, and the communication between MIPS core and PC is via network port.

![Figure 5-1: Test platform diagram](image)
5.1.1 Fingerprint recognition performance

The number of tested fingerprint image is 200, which are scanned from 20 fingertips, and scanned 10 images from one fingertip. The estimation method is from method proposed by Xiping Luo, etc.\cite{21} During estimation, define 10 fingerprint images from the same fingertip as a template, and every image is matched with the template which includes itself (the other 9 images) and the other templates. If more than 3 images in a template can be matched with the input image successfully, it can be regarded that it is from the same fingertip as the images in the template. If an image can not be matched successfully with the template including itself, this situation will be regarded as false rejection, else correct verification; if a image can be matched with a template which does not include it successfully, this situation is regarded false access.\cite{18}

Define reject \_num as the number of image rejected, correct \_num as the number of correct verification, and false \_num as the number of false verification, the functions to compute verification rate and rejection rate as shown in (4-3) and (4-4). The testing result showed that the verification rate is 99.47%, and the rejection rate is
6%.

\[
\begin{align*}
\text{Verification Rate} &= \frac{\text{correct num}}{\text{correct num} + \text{false num}} \times 100\% \\
\text{Rejection Rate} &= \frac{\text{reject num}}{200} \times 100\%
\end{align*}
\]

(5-1)
(5-2)

5.1.2 Hardware part performance

The hardware part is implemented with Verilog, and logic synthesized based on SMIC 0.13 \(\mu\)m standard CMOS technology via Design Compiler of Synopsys. The result shows that the design can achieve the 125MHz clock frequency, the cost is 7.1M gates.

The time to finish orientation estimation with VLSI implementation is shown as following:

\[
T_{ori} = (W_g \times W_g) \times w \times h \times T \\
= (15 \times 15) \times 256 \times 304 \times T \\
\approx 17.51 \times 10^6 \times T \\
= 17.51 \times 10^6 \times \frac{1}{125 \times 10^6} \\
= 0.1401(\text{sec})
\]

(5-3)

The time to finish frequency estimation with VLSI implementation is shown as following:

\[
T_{fre} = (W_g \times W_g) \times w \times h \times T \\
= (15 \times 15) \times 256 \times 304 \times T \\
\approx 17.51 \times 10^6 \times T \\
= 17.51 \times 10^6 \times \frac{1}{125 \times 10^6} \\
= 0.1401(\text{sec})
\]

(5-4)

The time to finish Gabor filter with VLSI implementation is shown as following:

\[
T_{gabor} = (W_g \times W_g) \times w \times h \times T \\
= (11 \times 11) \times 256 \times 304 \times T \\
\approx 9.42 \times 10^6 \times T \\
\approx 9.42 \times 10^6 \times \frac{1}{125 \times 10^6} \\
= 0.0754(\text{sec})
\]

(5-5)
Table 5-1 shows the comparison of the time costs of the modules with software implementation and the modules with VLSI implementation.

Table 5-1: Time-cost comparison of software and VLSI

<table>
<thead>
<tr>
<th></th>
<th>orientation</th>
<th>frequency</th>
<th>Gabor filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>software (second)</td>
<td>1.546</td>
<td>2.641</td>
<td>3.875</td>
</tr>
<tr>
<td>VLSI (second)</td>
<td>0.1401</td>
<td>0.1401</td>
<td>0.0754</td>
</tr>
</tbody>
</table>

(The system implemented with software runs on on Intel Pentium 2 E2140 @1.60GHZ, and the system implemented with VLSI is synthesized with 0.13 standard CMOS technology)

With the hardware-software cooperating implementation, the total time cost of this system is 1.3339 seconds. The time can totally satisfy the time tolerance of authentication system, compared with 9.062 seconds from software implementation.

5.2: Security analysis

The authentication structure used in this paper and cryptography mechanism can resist many attacks, include replay attack, attack on fingerprint template on server, attack on fingerprint template on client.

5.2.1 Replay attack

In this attack, attacker intercepts the information transmitted on the public channel, and then re-broadcasts the information to server or client. Because SQN is inserted, the replay attack can be resisted. When attacker re-broadcasts the intercepted information, and server or client verifies SQN, SQNs from different times of authentication times are different. When client or server verifies SQN fail, it will end the authentication process with fail result, so the replay stack can be resisted. Compared with system proposed by David D. Hwang, client sends cipher text and hash text of matching result and SQN, so the security and integrity of matching result
and SQN is protected.

After attacker intercepts the matching result, it bypasses the shared-key authentication and fingerprint authentication, sends the matching result to server end directly. Server will recognize the client is legal based on the replayed result, and this may cause the leak of critical information and financial lose. Because $SQN$ is inserted, server not only verifies matching result, also verifies $SQN$. Server can estimate the whether the result is from replaying. On the other hand, server sends $SQN$ and the authentication result to client, and client can verify whether the authentication result is from legal server. In this mechanism, server and client can verify each other, while in current authentication systems, client can not verify whether server is legal, and the interest of client can not be protected well.

5.2.2 Attack to fingerprint template on server

In this attack, attacker is vicious to steal the templates stored on server. This is similar to the attack to the database of bank. Attacking to the database of server can cause very fearful affection, because the invariance of fingerprint, and the number of fingertips is limited, once the fingerprint template is stolen, lose of client is impressible to remedy. While in the system of this paper, fingerprint template is stored on client end, so possibility of encountering this attack is zero.

5.2.3 Attack to fingerprint template on client

Attacker can get the fingerprint template of client by stealing the embedded device of fingerprint. The possibility of this attack exists in this system. For this attack, some technology resisting tamper can be taken (when the attack happens, clear the memory immediately); some other technology protecting the information can be taken (encode the information of fingerprint template). These technologies can be easily expanded in this system.
Chapter 6: Conclusion and future work

As fast developing of wireless network, the requirement to security authentication is becoming more and more urgent. In this paper, we proposed a hardware-software cooperating authentication system which fused fingerprint authentication and shared-key authentication. The researches of this paper include:

a) Compared with the traditional embedded authentication system, this paper proposed a hardware-software cooperating mechanism. The traditional hardware implementation mechanism is not flexible, configurable, and not easy to expand the functions; the traditional software implementation costs higher, and the power consumption is higher compared with hardware implementation. This paper takes the advantages traditional hardware implementation on function, performance, power consumption, cost, and the advantages of software implementation, flexible, configurable, and easy to expand. The time-cost parts in the system, orientation field and frequency field estimation, and Gabor filter, are implemented with VLSI, and the other parts, shared-key authentication, median filter, normalization, thinning, and matching are implemented with software. The test result shows that this system not only satisfies requires of performance, function, efficiency, also flexible, configurable.

b) Biometrics authentication is the most secure mechanism of security authentication, it connects the authentication medium with the user directly, but it is not suitable to use in wireless network directly, because the transfer medium of wireless network is open, when the biometric information is transferred on the wireless network, it is open to many attacks. Cryptology mechanism is now widely used in security filed, but the password is easy to be forgotten and stolen, and the physical authentication medium is also easy to be stolen. In a word, the cryptology mechanism can not connect the authentication medium with user. In this paper, an authentication mechanism fusing biometrics authentication and cryptology mechanism is proposed. The proposed system connects the authentication medium with the user directly, and protects the biometric information transferred on the public wireless channel.
As the developing and widely using of biometric authentication, the performance of fingerprint recognition algorithms is still a critical issue, because the verification rate is still not high enough and the false verification rate is still a little high in practice. So in future, the modification of fingerprint recognition algorithms is an issue which is necessary to pay attention on.
Reference


