7

Performance Analysis of Smart Card-Based Fingerprint Recognition For Secure User Authentication

Youn-Hee Gil[1], Yongwha Chung[1], Dosung Ahn[2], Jihyun Moon[2] and Hakil Kim[2]

1 Biometrics Technology Research Team, Electronics and Telecommunications Research Institute, 161 Kajong-dong, Yusong-gu, Daejon, 305-350, Korea {yhgil,ywchung}@etri.re.kr

2 Department of Automation Engineering, Inha University, Yonghyun-dong 253, Nam-gu, Incheon, 402-751, Korea, dosung@email.com, jieney@innocent.com, hikim@inha.ac.kr

Abstract: In the modern electronic world, authentication of a person is an important task in many areas of day-to-day life. Using a biometrics to authenticate a person's identity has several advantages over the present practices of Personal Identification Number stored in smart cards. However, there is an open issue of integrating biometrics into the smart cards. Typical authenticating algorithms by using biometrics may not be executed in real-time on the resource-constrained smart cards. In this paper, we analyse first the performance requirement of the biometric authentication on the smart cards. Then, to satisfy the requirement, we have developed a light-weighted finger recognition algorithm. Finally, we investigate the possibility of integration of the algorithm into the smart card. Based on our simulation results, a smart card can be designed such that the card can encapsulate all the critical information including the biometrics data, and perform all the comparison securely inside the smart card without any data leaking out.

1. INTRODUCTION

Smart card, chip card, or IC card [l-3], which is a credit card sized plastic card, embedded with a special type of hardwired logic or a microprocessor to hold critical information securely, is a good choice of light-weighted hardware assisted cryptographic devices for protection at the client side when conducting some kinds of online activities, such as E-commerce, E-business, and E-government. Especially, the smart card is used as a PKI storage device because chips are tamper-resistant.

Computations on the chips involved in authentication of digital signature and key exchange are more secure as they are isolated from other parts of an operating system. The smart card also enables credentials and other private information to be portable between computers at work, home, or on the road.

In recent years, there is an increasing trend of using biometrics, which refers the personal biological characteristics used for authentication or identification. It relies on "something that you are", therefore can inherently differentiate between an authorized person and a fraudulent imposter. Compared with using the four-digit Personal Identification Number(PIN), it can be more secure to use the biometric information of size 500B for protection of the critical data. Furthermore, biometric information has no concern to be forgotten. Smart cards play an important role in biometrics[4-6]. In general identification system, the biometric templates are often stored in a central database. With the central storage of a biometric data, there is an open issue of misuse of the biometric information for the purpose the owner may not be aware of. We can decentralize the database storage part into millions of smart cards and give it to the owners after authorization of it.

However, most of these systems have a common characteristic that the biometrics authentication process is solely accomplished out of the smart card processor[3]. For example, in fingerprint-based smart card system, the critical fingerprint master template information needs to be insecurely released into the external fingerprint reader from the card to be compared with the input fingerprint template. To heighten security level, the comparison of the master template with the fingerprint sample needs to be performed by the in-card processor, i.e., match-on-card[6], not the external reader.

In this paper, we examine whether the in-card processor can execute the entire authentication and verification steps. If so, all of the critical information including the biometrics data can be encapsulated in the card, and all the computation can be performed securely inside the card without any data leaking out. Although it's not possible, at least, the in-card processor should be capable of the matching and verification process. That is, the smart card does not only perform the ordinary PIN verification and certification storage, but also involve the biometrics authentication. However, the processing power of the in-card processor is very limited. Thus, performance requirements of the biometric authentication steps on the in-card processor should be analyzed first. Then, to satisfy the requirement, we have developed a light-weighted finger recognition algorithm. To investigate this feasibility of the in-card processor, we conducted performance analysis of our fingerprint matching algorithm with an instruction set simulator. The simulation results showed that the card could encapsulate the biometrics data and perform the comparison securely inside the card without any data leaking out.

The organization of the paper is as follows. Overview of the smart card system considered in this paper is given in Section 2. In Section 3, the selected biometrics

authentication system is explained. Performance analysis results are shown in Section 4, and concluding remarks are made in Section 5.

2. SMART CARD SYSTEM

A smart card resembles a credit card in terms of physical look and size with one or more semiconductor devices attached to a module embedded in the card. More specifically, the smart card is a portable, very secure, low cost, and intelligent device capable of manipulating and storing data. This intelligence is due to an in-card processor that is suitable for use in a wide range of applications[1-3].

Figure. 1 shows the smart card system we are developing[7], and its characteristics are summarized as follows:

Hardware. The in-card processor is a 32-bit ARM7TDMI[8] to manipulate and interpret data. The memory in the smart card consists of three different types. The ROM is used for the smart Card Operating System(COS) and is usually embedded during manufacture. The RAM is used by the COS as temporary storage area. The user available data segments are allocated in the EEPROM. The size of each memory type is 64KB, 1KB, 40KB for ROM, RAM, EEPROM, respectively. The first two types of memory are not available for user access. Several levels of access security are supported in the EEPROM. The methods of assigning access security can be controlled through use of a PIN or a biometric template or using cryptography. The smart card also include the Crypto-Coprocessor and the Random Number Generator(RNG) to perform cryptographic algorithms in real-time. Finally, for the contact interface, the external interface module is included.

Software. From the time of smart card manufacture to the end of loading application and usage by consumers, different kinds of software are used to handle smart cards. The Card OS(COS) is a vendor dependent component of the software, and supports a file system on the EEPROM storage, command interpretation, and security options for the data stored on the smart card. During initialization and personalization, application specific data structures are loaded. During the usage of the smart card, the card interacts with the application through the Application Programming Interface(API). The smart card also includes the Java Virtual Machine to support multiple applications. The details of the targeted smart card can be found in [7].

Note that, because of the area restriction of the smart card chip, we select ATM7TDMI. However, the maximum performance of the in-card processor is 60 Million Instructions Per Second(MIPS) and the maximum clock rate of it is 66 MHz. This processing power is very limited compared with the typical PCs having 800 MHz Pentium III. Thus, very careful performance analysis is required to integrate the biometrics into the resource-constrained smart card system.

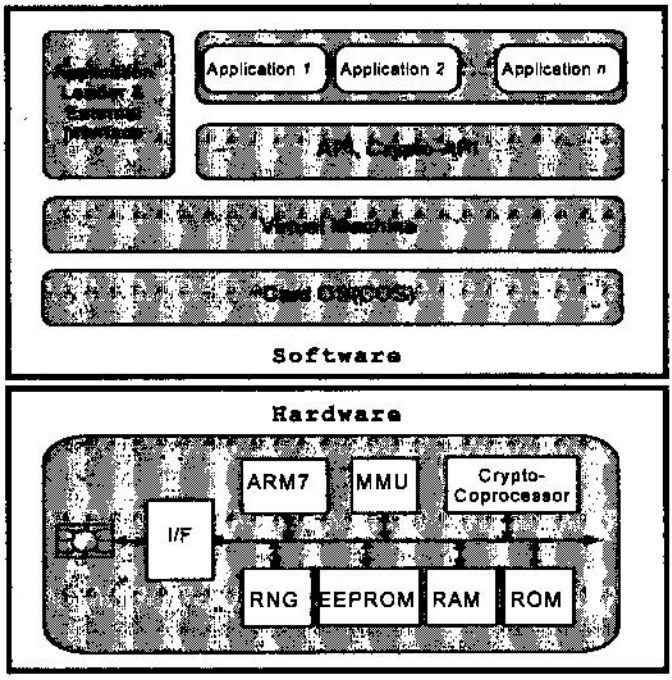

Figure 1. Target Smart Card System [7]

3. SELECTED BIOMETRICS AUTHENTICATION

In this paper, fingerprint is chosen as the biometrics for authentication as it is more mature in terms of the algorithm availability and feasibility, while the other kinds of biometrics, such as the iris and face, may not be well suited to an ordinary smart card processor with respect to its limited processing power. Note that the problem of resolving the identity of a person can be categorized into two distinct types of problems. Authentication or verification refers to the problem of confirming or denying a person's claimed identity, whereas identification or recognition refers to the problem of establishing a subject's identity. That is, the authentication system matches a person's claimed identity to his/her previously enrolled pattern(i.e., "one-to-one" comparison). However, the identification system identifies a person from the entire enrolled population by searching a database for a match(i.e., "one-to-many" comparison). In this paper, we only focus on the authentication system by using smart cards with biometrics.

Fingerprint is especially suitable as a method to authenticate users to use smart cards. This can be elaborated by considering the time complexity of the algorithm.

Whether the in-card processor is capable to execute the entire fingerprint matching algorithm in real-time depends on the time complexity. In the following, we briefly describe the fingerprint matching system used in our research.

The technique for fingerprint matching has been developed in the field of image processing. Generally, when we want to compare two fingerprint images, it is needless and wasteful to accomplish this by repeating pixel-by-pixel checkup. On the contrary, it is better to pre-process the images so that the unique features of the fingerprint images are extracted first, and then simply compare these features instead. Here, such kinds of domain specific features for fingerprint matching are called minutiae[4]. Minutiae refer to the ridge ends and ridge bifurcations of a fingerprint.

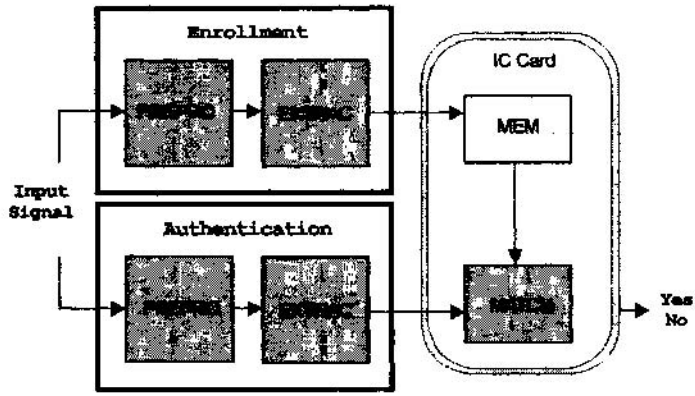

Figure 2. Fingerprint Matching System using Smart Card

Figure. 2 shows the general steps in the fingerprint matching system using smart cards. The system operates in two phases. In the off-line enrollment phase, minutiae are extracted and stored in the smart card. In the on-line authentication phase, the stored minutiae and the minutiae from live fingerprint are presented to the system, and then the similarity between them is examined. This authentication phase consists of the following three steps:

Image Pre-Processing. This refers to the refinement of the original fingerprint image against image distortion obtained from the fingerprint sensor. It consists of three stages. Binary conversion stage applies low-pass filter to smooth the high frequency regions of the print and threshold to each sub-segment of the image. Thinning stage generates one-pixel-width skeleton image by considering each pixel with its neighbors. In positioning stage, the skeleton obtained is transformed and/or rotated such that valid minutiae information can be extracted.

Minutiae Extraction. This refers to the extraction of feature in the fingerprint image. After this step, some of the minutiae are selected and stored into a template

file, which includes the position, orientation, and type(ridge ending or bifurcation) of minutiae. However, false minutiae can be extracted due to the noise during the image acquisition step and/or information loss in the pre-processing step. The false minutiae deteriorate overall accuracy significantly in the succeeding matching step. Therefore, we have developed a method to remove such false minutiae using ridge distance information and various types of noises. Another advantage of this removing method is to reduce the execution time by eliminating unnecessary computation due to false minutiae. The details of our false-minutiae removing algorithm can be found in [9].

Minutiae Matching. When user's fingerprint image is obtained, we use image processing techniques before turning the image into a skeleton image as explained above. After getting thinned image, we find out minutiae points in it. Note that the correction for scaling, translation, and rotation of the image are needed before starting the matching step.

Based on the minutiae, we compare the input fingerprint image with the template file. Actually, minutiae matching is composed of the alignment stage and matching stage. Alignment is the most time consuming step in the whole fingerprint recognition, thus success of optimizing the alignment stage is the key to achieve real-time performance. Therefore, we have developed an alignment algorithm using specific data structure called as "clique", which can optimize the search space. "Clique" is the data structure consisted of information derived from the triangle-shaped three minutiae, such as three minutiae points and orientation, and the radius of circumcircle. Especially, the radius of circumcircle can be used as the search key because there exists only one in one traingle-shaped form.

In order to get the clique data structure, we select three minutiae by picking up the first and second nearest minutiae point from one minutia using Euclidean distance measurement. Figure 3 represents these three minutiae and the circumcircle formed using them. After getting the center of the circumcircle and three inter angles between three minutiae points, the three minutiae can be arranged according to the angles. The right side point of the largest inter angle (α) is the first, and the second and third point are chosen as clockwise direction from it.

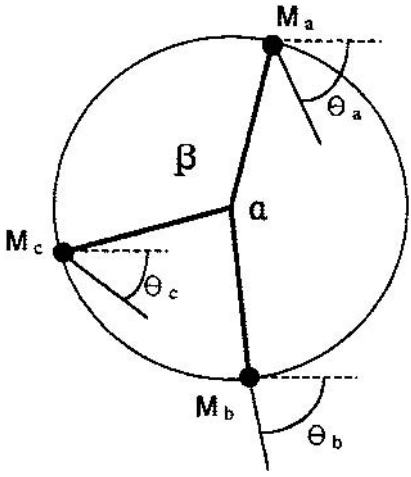

Figure 3. Geometry of 3 minutiae clique

The structure of clique is consist of the radius of circumcircle (r) , two inter angle (α, β), three orientations ($\theta_a, \theta_b, \theta_c$), and three types ($\zeta_a, \zeta_b, \zeta$) as below:

$Ci = \{r, \alpha, \beta, \theta_a, \theta_b, \theta_c, \zeta_a, \zeta_b, \zeta_c\}$, *where* i = 1, 2, … , N , and N is the number of clique.

Given a fingerprint image, we extract all of the possible triple minutiae sets to be get ready for construction clique. Here, the number of clique is defined as $n \times {}_k C_2$, $2 \le k \le n - 1$ when n is the number of feature points. Selecting meaningful k is important and affects to the performance. The complexity of blind search between two fingerprint images, each of which has n_t, n_s minutiae points, is between the range $\{\Theta(n_t n_s), \Theta(n_t^3 n_s^3)\}$. By using the radius of circumcircle as the search key, search space can be reduced compared with the exhaustive search. After using our minutiae matching algorithm, the number of points that matches can be obtained. Then, the match score could be measured by normalizing with the number of input minutiae. This score is in the range {0,1}, where 1 corresponds to a perfect pattern match. The details of the clique algorithm can be found in [10].

4. PERFORMANCE REQUIREMENT ANALYSIS

Our light-weighted minutiae matching algorithm developed to satisfy the resource constraint is applicable to the smart card environment perfectly because of its real-time performance.

To establish an objective assessment of our system, sample image pairs of size 832 × 768 selected from the NIST Special Database 14[11] have been used to estimate the performance numbers. The simulator we used is the Simple Scalar[12] which models the behavior of a microprocessor in software on a host system.

To characterize the computational requirement of each step in the fingerprint matching system, we break down the instructions into six components as shown in Table 1. The low-level step(Pre-Processing) needs a lot of integer computations. However, the high-level step(Minutiae Matching) involves less ALU operations and more load/store operations.

Table 1. Distribution of Instructions Executed

Step	Load	Store	Uncond Branch	Cond Branch	Int Comp	FP Comp
Pre-Processing	67,937,581 (18%)	11,720,360 (3%)	3,075,362 (1%)	56,968,411 (15%)	227,089,206 (62%)	92,242 (1%)
Minutiae Extraction	19,004,017 (23%)	2,873,022 (3%)	817,493 (1%)	11,010,208 (13%)	50,034,953 (59%)	1,115,595 (1%)
Minutiae Matching	3,640,392 (18%)	2,600,846 (13%)	219,109 (1%)	1,933,178 (10%)	8,738,570 (43%)	2,992,937 (15%)

To show the performance requirement of the in-card processor, the number of instructions and the estimated execution time on the 8-bit Intel-8051 and 32-bit ARM7-based smart cards are summarized in Table 2.

In Table 2, we present estimated result per each step.

Table 2. Summary of Simulation Results

Step	Total No of Instructions	Estimated time on 8051	Estimated Time on ARM7
Pre-Processing	366,883,251	158.6 sec	6.1 sec
Minutiae Extraction	84,856,108	36.4 sec	1.4 sec
Minutiae Matching	20,125,037	7.8 sec	0.3 sec

According to the Table 2, it is impossible to assign minutiae extraction or matching step as well as pre-processing to the 8051 chip. This is because computation using biometric information requires much memory and time. However, ARM7 shows improved result. Using this can make match-on-card to be

realized. Currently, 32-bit smart card is somewhat expensive to be applied for ordinary system. Nevertheless, it can be good solution for the system that should be guaranteed very high-level security such as E-Commerce, E-Business, and E-Government.

With respect to the limited processing power of the in-card processor, all of the three steps above can't be assigned to the in-card processor. Instead, we consider assigning only the third step to the in-card processor, which is the minutiae matching. This is because the first two steps involve rigid image processing computation, which is too exhaustive to be executed in the in-card processor. These computation steps can be easily carried out in real-time by a fingerprint capture device or a smart card terminal equipped with at least a 500 MIPS processor. Therefore, the whole computational steps can be performed in real-time, and the smart card can encapsulate the biometrics data and perform the comparison securely inside the card without any data leaking out.

5. CONCLUDING REMARKS

Smart card is a model of very secure storage, and biometrics is the ultimate technology for authentication. The two can be combined in many applications to enhance both the security and authentication. However, a careful analysis is required to integrate the biometrics into the smart cards because the smart cards have very limited resources.

Our performance analysis shows that the use of a 32-bit smart card processor is feasible in order to conduct the fingerprint authentication actively with respect to the advanced techniques in fingerprint image comparison. In contrast to the traditional PIN verification currently being used, this further enhances the security issues in adopting the smart card into many emerging applications such as:

- making legally binding digital signatures for E-commerce, on-line documentation signing, contracts, taxation, legal applications
- access to the system that should be guaranteed very high level security in government especially such as Department of Defense
- user authentication used in internet banking or electronic commerce
- gaining access to secure websites.

6. REFERENCES

[1] Dreifus, H. and Monk, T.: Smart Cards. John Wiley & Sons (1997)

[2] CardTech/SecureTech.: Proc. of the CardTech/SecureTech 2000 Conference (2000)

[3] Mearns, C. and Jones, D.: The Smart Card. SJB Research (1999)

[4] Jain, A., Bole, R., and Panakanti, S.: Biometrics – Personal Identification in Networked Society. Kluwer Academic Publishers (1999)
[5] Pakanti, S., Bolle, R., and Jain, A.: Biometrics: The Future of Identification. IEEE Com-puter, Vol. 33, No. 2 (2000) 46-49
[6] Biometric Consortium.: Proc. of the Biometric Consortium 2000 Conference (2000)
[7] Kim, H. et al.: Specification for the Next-Generation IC Card System(Korean). Technical Report, ETRI (2000)
[8] Furber, S.: ARM System-on-Chip Architecture. Addison-Wesley (2000)
[9] Kim, H. and Kim, H.: Rotation-Scale-Translation-Intensity Invariant Algorithm for Finger-print Identification(Korean). Journal of The Institute of Electronics Engineers of Korea, Vol. 35, No. 6 (1998) 838-850
[10] Ahn, D. and Kim, H.: Fingerprint Recognition Algorithm using Clique(Korean). Technical Report, Inha University (2000)
[11] http://www.itl.nist.gov /iaui/vip/fing/fing.html
[12] Burger, D. and Austin, T.: The SimpleScalar Tool Set, Version 2.0. Technical Report, University of Wisconsin (1997)