VLSI Architecture of the Fuzzy Fingerprint Vault System

Sung Jin Lim[1], Seung-Hoon Chae[1], and Sung Bum Pan[1,2,*]

[1] Dept. of Information and Communication Engineering, Chosun Univ., 375, Seosuk-dong Dong-gu, Gwangju, 501-759, Korea
[2] Dept. of Control, Instrumentation, and Robot Engineering, Chosun Univ., 375, Seosuk-dong Dong-gu, Gwangju, 501-759, Korea
{gigasj83,ssuguly}@gmail.com, sbpan@chosun.ac.kr

Abstract. User authentication using fingerprint information provides convenience as well as strong security at the same time. However, serious problems may cause if fingerprint information stored for user authentication is used illegally by a different person since it cannot be changed freely as a password due to a limited number of fingers. Recently, research in fuzzy fingerprint vault system has been carried out actively to safely protect fingerprint information in a fingerprint authentication system. In this paper, we propose hardware architecture for a geometric hashing based fuzzy fingerprint vault system. The proposed architecture consists of the software module and hardware module. The hardware module performs the matching for the transformed minutiae in the enrollment and verification hash table. We also propose a hardware architecture which parallel processing technique is applied for high speed processing.

Keywords: Fingerprint authentication, fuzzy vault, geometric hashing, fuzzy fingerprint vault.

1 Introduction

The authentication system based on biometric information offers greater security and convenience than the traditional methods of personal verification. The biometrics such as fingerprint, iris, and voice has been received considerable attentions, which refers the personal biological or behavioral characteristics used for verification or identification. Since biometrics cannot be lost or forgotten like passwords, biometrics has the potential to offer higher security and more convenience for the users. The fingerprint is chosen as the biometrics for verification in this paper. Owing to their uniqueness and immutability, fingerprints are today the most widely used biometric features. If the biometric data are compromised, the user may quickly run out of the biometric data to be used for authentication and cannot re-enroll[1-2]. Recently, study on fuzzy fingerprint vault which fuzzy vault theory is applied to fingerprint authentication has been carried out after Juels and Sudan proposed the fuzzy vault

* Corresponding author.

F. Schwenker and N. El Gayar (Eds.): ANNPR 2010, LNAI 5998, pp. 252–258, 2010.

theory. Fuzzy fingerprint vault is a cryptology method which secret key and fingerprint information of the user are combined to obtain a secret key for only the right user. Real minutiae of the user is protected by creating a polynomial using the user's secret key and organizing the fingerprint template of the user with the user's real minutiae after creating the chaff minutiae randomly[3].

While study to simultaneously protect user fingerprint information and secret key of that user using fuzzy fingerprint vault is being reported, they cannot be realized because the alignment process was omitted due to absence of the fingerprint. To solve this problem, a method which geometric hashing technique is applied to the fuzzy fingerprint vault system was proposed[4-7]. The geometric hashing technique is an object authentication algorithm which object information is extracted and stored in a database and searched after being geometrically transformed[8].

In this paper, we propose hardware architecture for a fuzzy fingerprint vault system based on geometric hashing. The proposed architecture is performed by combining the software and hardware modules. The software module consists of modules for fingerprint minutiae information extraction, fingerprint template generation, fingerprint hash table generation and database storage. The hardware module consists of the matching module, verification module and memory to store the enrollment hash table and verification hash table. The matching module compares the transformed minutiae of the enrollment hash table and transformed minutiae of the verification hash table. The verification module takes the role of the calculation according to the result of the matching module. In addition, we propose a hardware architecture which parallel processing technique is applied for high speed processing of the fuzzy fingerprint vault system. Software module is identical to the previously proposed hardware architecture. The hardware module consists of the memory for storing the enrollment hash table and verification hash table, matching modules and the verification module.

The organization of the paper is as follows. Section 2 introduces fuzzy fingerprint vault based on the geometric hashing. Section 3 explains the hardware architecture for fuzzy fingerprint vault based on the geometric hashing, Section 4 shows the experimental results and Section 5 concludes.

2 Fuzzy Fingerprint Vault

Juels and Sudan proposed a scheme for crypto-biometric system called fuzzy vault. This is method which can protect the user's important secret key and biometric information using fuzzy concept. Clancy et al. proposed a fuzzy fingerprint vault based on the fuzzy vault of Juels and Sudan[4-5]. Using multiple minutiae location sets per finger, they first find the canonical positions of minutia, and use these as the elements of the set A. They added the maximum number of chaff minutiae to find R that locks. However, their system inherently assumes that fingerprints are pre-aligned. This is not a realistic assumption for fingerprint-based authentication schemes.

The architecture of the fuzzy fingerprint vault system of Chung et al. [6] consists of two processes: enrollment and verification processes as shown in Fig. 1. Enrollment process consists of minutiae information acquisition stage, enrollment hash table generation stage again. In minutiae information acquisition stage, minutiae information includes real minutiae of a user and chaff minutiae generated randomly.

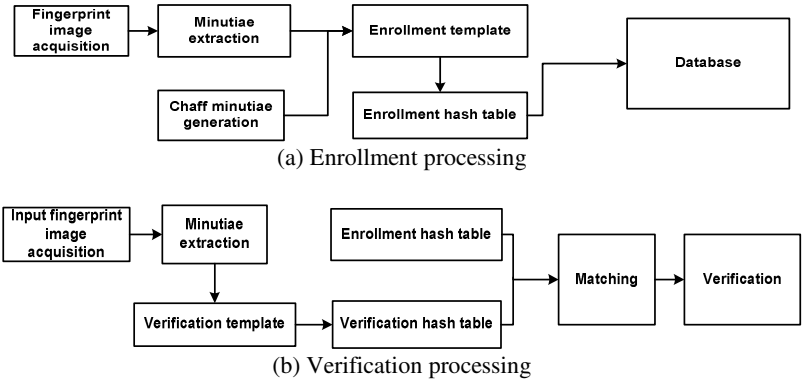

Fig. 1. Fuzzy fingerprint vault system

It is challenging to perform fingerprint verification with the protected template added by chaff minutiae. And then, Chung et al. applies modified geometric hashing. According to the geometric characteristics of the minutiae information, a table, called an enrollment hash table, is generated. Let $m_i = (x_i, y_i, \theta_i, t_i)$ represent a minutia and $\boldsymbol{L} = \{m_i \mid 1 \le i \le r\}$ be a locking set including the real and chaff minutiae. In $\boldsymbol{L}$, the real and chaff minutiae can be represented by $\boldsymbol{G} = \{m_i \mid 1 \le i \le n\}$ and $\boldsymbol{C} = \{m_i \mid n+1 \le i \le r\}$, respectively. Note that, the enrollment hash table is generated from $\boldsymbol{L}$. In the enrollment hash table generation stage, an enrollment table is generated in such a way that no alignment is needed in the verification process for unlocking vault by using the geometric hashing technique. That is, alignment is pre-performed in the enrollment table generation stage.

After the enrollment process, the verification process to separate the chaff minutiae($\boldsymbol{C}$) from the real minutiae($\boldsymbol{G}$) in the enrollment minutiae table should be performed. In the verification process, minutiae information(unlocking set $\boldsymbol{U}$) of a verification user is obtained and a table, called verification table, is generated according to the geometric characteristic of the minutiae. Then, the verification table is compared with the enrollment minutiae table, and the subset of real minutiae is finally selected. Note that, the verification table generation stage is performed in the same way as in the enrollment process. In comparing the enrollment and verification minutiae tables, the transformed minutiae pairs with the same coordinates, the same angle, and the same type are determined. The minutiae pairs having the maximum number and the same basis are selected as the subset of real minutiae($\boldsymbol{G}$). Also, any additional alignment process is not needed because pre-alignment with each minutia is executed in the enrollment and verification minutiae table generation stages.

3 Hardware Architecture of the Fuzzy Fingerprint Vault

To implement the hardware system of the fuzzy fingerprint vault, the proposed architecture was performed by integrating software and hardware modules as shown in Fig. 2. The enrollment processing for the fuzzy fingerprint vault system consists of steps for real minutiae extraction, chaff minutiae generation, fingerprint template generation,

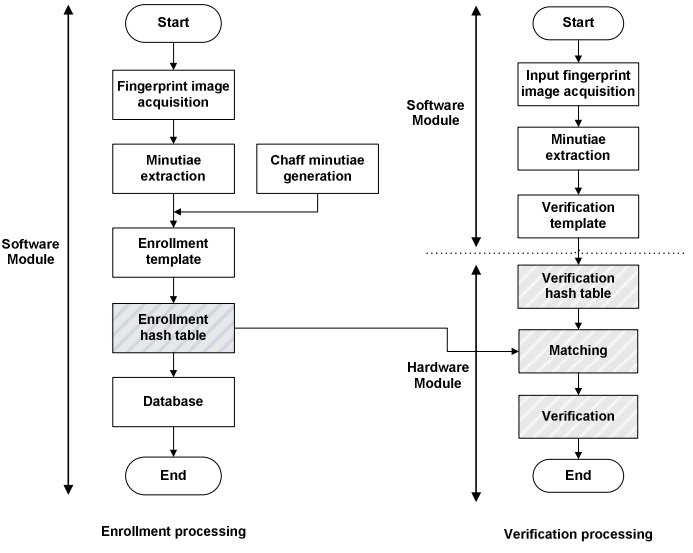

Fig. 2. Flow diagram of the fuzzy fingerprint vault system

fingerprint hash table generation and fingerprint database storage. The verification processing consists of steps for input fingerprint minutiae extraction, fingerprint hash table generation, matching and verification (Candidate list generation step).

The matching step and verification step of the verification processing is performed in hardware. Since the amount of computation of the matching step for the verification processing increases as the number of chaff minutiae increases in the enrollment processing, it is advisable for the matching step and verification step to be performed in hardware. On the other hand, the enrollment processing and fingerprint hash table generation step in the verification processing are performed in software. The proposed hardware module consists of two memories(enrollment and verification hash table), matching module and verification module as shown in Fig. 3. Hash table for each is organized of transformed minutiae. The enrollment fingerprint transformed minutiae is created by geometric transformation of user's real minutiae and chaff minutiae that was inserted to protect this in the enrollment processing. The verification fingerprint transformed minutiae is created by geometric transformation of minutiae of the fingerprint for verification.

The enrollment hash table can be expressed as $E = \{tr_i \mid 0 \le i \le r\text{-}1\}$ where r is the number of enrollment fingerprint templates. tr_i is the hash table which is created through geometric hashing after selecting m_i among the fingerprint template as the reference point and consists of r-1 transformed minutiae. The verification fingerprint transformed minutiae can be expressed as $V = \{tr_j \mid 0 \le j \le s\text{-}1\}$ where s is the number of verification fingerprint templates. The matching module consists of the Compare module and Count module. The Compare module is the module that compares the enrollment fingerprint transformed minutiae and verification fingerprint transformed minutiae and the Count module calculates the number of corresponding

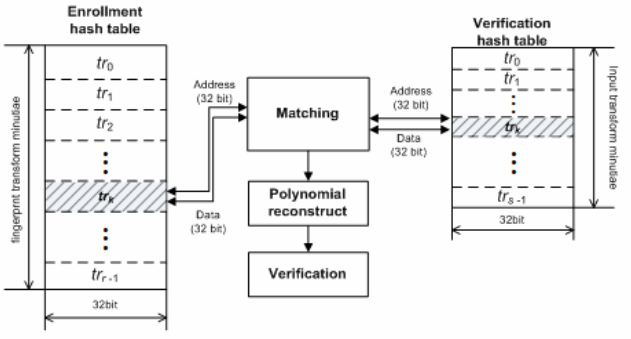

Fig. 3. Structural diagram of the proposed hardware module

transformed minutiae. The verification module is the module that aligns calculated similarity in high similarity order.

It assumes the number of enrollment fingerprint templates to be *r*, *tr* of the enrollment hash table *E* to be *r*, number of the verification fingerprint template to be *s* and *tr* of the verification hash table *V* to be s. First, tr_0 of the verification hash table and tr_0 of the enrollment hash table is input to the matching module. In the Compare module of the matching module, tr_0 of the verification hash table and tr_0 of the enrollment hash table are compared. After comparing the coordinate, angle and type of the transformed minutiae for the two tr_0 that were input, whether or not they match is sent to the Count module. The Count module calculates the number of transformed minutiae that match. Then, tr_1 of the enrollment hash table is input to the matching module and compared with the tr_0 of the verification hash table. After comparison of all *tr* in the enrollment hash table with tr_0 of the verification hash table are completed using the same method, tr_1 of the verification hash table is input to the matching module. Up to tr_{s-1} of the verification hash table is compared by executing this repeatedly. The verification module performed alignment of the calculated similarity. By using the number of corresponding transformed minutiae, similarity is measured and candidate list is generated in high similarity order.

In this paper, we also propose a hardware architecture which parallel processing technique is applied to reduce the matching time. Separation of the software and hardware modules is identical to the previously proposed hardware architecture. The proposed parallel processing hardware module consists of two memories storing the enrollment hash table and verification hash table, matching module and verification module as shown in Fig. 4. For parallel processing, the number of matching modules used in the hardware module is two. While the architecture of the matching module is similar to the previously proposed hardware architecture, it is different because the input enrollment fingerprint transformed minutiae is output after the comparison. The architecture of the verification module is identical to the previously proposed hardware architecture. When the number of the enrollment fingerprint templates is *r* and the number of the verification fingerprint templates is *s*, the enrollment hash table *E* consists of *r* and verification hash table V consists of *s*.

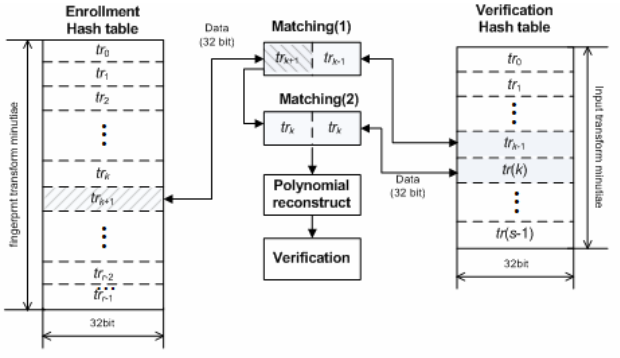

Fig. 4. Structural diagram of the proposed parallel processing technique hardware module

4 Experimental Results

The number of real minutiae that was used in the hardware architecture experiment for the fuzzy fingerprint vault system based geometric hashing proposed in this paper is between a maximum of 90 and minimum of 16. The experiment was performed by adding 100, 200, 300 and 400 chaff minutiae. The software module was realized by using C language in Visual C++ 6.0.

For hardware module implementation, a Spartan 3E starter board was used. The development board contains a Xilinx XC3S500E FPGA. The hardware module was designed by using VHDL in Xilinx ISE 9.2. Hardware simulation was performed in Moledmsim XE 6.0.

Table 1 shows the major resources that were used when the hardware architecture for the proposed fuzzy fingerprint vault and hardware architecture which the parallel processing technique was applied were implemented in the development board. Since the parallel processing hardware architecture has matching modules whose number equals that of the verification fingerprint transformed minutiae when there is one matching module in the proposed hardware architecture, the amount of resources that are used is large. It can be seen that the amount of hardware is about 56% higher for the number of slices, about 28% higher for the number of slice flip flops and 58% higher for the total number of 4 input LUTs in the parallel processing hardware architecture.

Execution time of the hardware module for the fuzzy fingerprint vault system was measured with number of chaff minutiae equal to 100, 200, 300 and 400. As shown in Table 2, real time processing is possible even when the number of chaff minutiae is increased to improve security in the proposed hardware architecture.

Table 1. Major resources when matching module is implemented in a Xilinx Spartan 3E FPGA

Number of matching module	Number of Slices	Number of Slices Flip Flop	Total Number of 4 input LUTs
1	419 out of 4,656(9%)	161 out of 9,312(2%)	668 out of 9,312(7%)
2	496 out of 4,656(10%)	246 out of 9,312(2%)	762 out of 9,312(8%)

Table 2. Required number of cycles according to the number of chaff minutiae

Module \ Chaff minutiae	Number of chaff minutiae			
	100	200	300	400
software	30,128,840	56,376,516	75,087,761	110,009,410
1 matching module	10,023,288	12,027,946	18,763,596	26,461,481
2 matching module	8,173,713	9,808,456	15,301,191	21,578,603

5 Conclusion

While a user authentication system using fingerprint information provides convenience and strong security at the same time, serious problems may cause if the fingerprint information is used illegally or leaked. In this paper, we proposed hardware architecture for a geometric hashing based fuzzy fingerprint vault system. The matching module of the proposed hardware architecture was performed so that all transformed minutiae in the enrollment fingerprint hash table are matched with each transformed minutiae in the verification fingerprint hash table. In addition, the hardware architecture of the matching system which parallel processing technique was applied for high speed processing of the system organizes matching modules in number equal to the number of transformed minutiae in the input fingerprint hash table and matches them simultaneously. Execution time of the proposed system was 0.24 second for 36 real minutiae and 200 chaff minutiae and 0.53 second for 400 chaff minutiae. In addition, execution time for hardware architecture with 2 matching modules which parallel processing technique was applied was 0.18 second and 0.47 second respectively for the same condition. Based on the experimental result, it was verified that real-time fingerprint authentication is possible by using the hardware architecture of the proposed fuzzy fingerprint vault system and high speed processing is possible by applying parallel processing technique.

References

1. Maltoni, D., Maio, D., Jain, A.K., Prabhakar, S.: Handbook of Fingerprint Recognition. Springer, Heidelberg (2003)
2. Uludag, U., Jain, A.: Securing fingerprint template: Fuzzy vault with helper data. In: Conf. on Computer Vision and Pattern Recognition Workshop, pp. 163–172 (2006)
3. Juels, A., Sudan, M.: A fuzzy vault scheme. In: IEEE International Symposium on Information Theory, p. 408 (2002)
4. Clancy, T., Kiyavash, N., Lin, D.: Secure smartcard-based fingerprint authentication. ACM SIGMM Multim., Biom. Met. & App., 45–52 (2003)
5. Uludag, U., Pankanti, S., Jain, A.K.: Fuzzy vault for fingerprints. In: Kanade, T., Jain, A., Ratha, N.K. (eds.) AVBPA 2005. LNCS, vol. 3546, pp. 310–319. Springer, Heidelberg (2005)
6. Chung, Y., Moon, D., Lee, S., Jung, S., Kim, T., Ahn, D.: Automatic alignment of fingerprint features for fuzzy fingerprint vault. In: Feng, D., Lin, D., Yung, M. (eds.) CISC 2005. LNCS, vol. 3822, pp. 358–369. Springer, Heidelberg (2005)
7. Lee, S., Moon, D., Jung, S., Chung, Y.: Protecting secret keys with fuzzy fingerprint vault based on a 3D geometric hash table. In: Beliczynski, B., Dzielinski, A., Iwanowski, M., Ribeiro, B. (eds.) ICANNGA 2007. LNCS, vol. 4432, pp. 432–439. Springer, Heidelberg (2007)
8. Wolfson, H., Rigoutsos, I.: Geometric hashing: an overview. IEEE Computational Science and Engineering 4, 10–21 (1997)