An Efficient Selective Encryption of Fingerprint Images for Embedded Processors

Daesung Moon, Yongwha Chung, Sung Bum Pan, Kiyoung Moon, and Kyo Il Chung

Biometric-based authentication can provide a strong security guarantee of the identity of users. However, the security of biometric data is particularly important as any compromise of the biometric data will be permanent. In this paper, we propose a secure and efficient protocol to transmit fingerprint images from a fingerprint sensor to a client by exploiting the characteristics of the fingerprint images. Because the fingerprint sensor is computationally limited, a standard encryption algorithm may not be applied to the full fingerprint images in real-time to guarantee the integrity and confidentiality of the fingerprint images transmitted. To reduce the computational workload on the resource-constrained sensor, we apply the encryption algorithm to a nonce for integrity and to a specific bitplane of each pixel of the fingerprint image for confidentiality. Experimental results show that the integrity and confidentiality of the fingerprint images can be guaranteed without any leakage of the fingerprint ridge information and can be completed in real-time on embedded processors.

Keywords: Biometrics, fingerprint protection, partial encryption.

I. Introduction

Traditionally, verified users have gained access to secure information systems, buildings, or equipment via multiple personal identification numbers, passwords, smart cards, and so on. However, these security methods have important weaknesses in that they can be lost, stolen, or forgotten. In recent years, there is an increasing trend of using biometrics, which refers to the use of personal biological or behavioral characteristics for verification of identity [1], [2].

The fingerprint is chosen as the biometric information for verification in this paper, as it is more mature in terms of algorithm availability and feasibility [2]. Current examples of fingerprint verification include their use in contexts such as social service databases, wherein individuals must be prevented from using multiple aliases; watch list checks in immigration offices; and identity card issuance. Furthermore, we consider a sensor-client-server model [3] for remote user authentication. In this model, the sensor captures a fingerprint image, the client extracts some features from the image, and finally the server compares the extracted features with the stored features.

In this model, however, security issues reguire that the opponents will neither be able to access the individual information/measurements nor be able to pose as other individuals by electronically interjecting stale and fraudulently obtained biometrics measurements into the system [2]-[4]. When the system and/or its communication channels are vulnerable to open physical access, cryptographic methods should be employed to protect the fingerprint information. Although either the system or the communication channel can be attacked in this model, we focus on protecting against attacks on the communication channels only.

The straightforward approach to guaranteeing the integrity

Manuscript received Jan. 23, 2006; revised June 13, 2006.

The second author of this research was supported by the Ministry of Information and Communication, Korea, under the Chung-Ang University HNRC (Home Network Research Center)-ITRC support program supervised by the Institute of Information Technology Assessment.

Daesung Moon (phone: + 82 42 860 1083, email: daesung@etri.re.kr), Kiyoung Moon (email: kymoon@etri.re.kr), and Kyo Il Chung (email: kyoil@etri.re.kr) are with Information Security Research Division, ETRI, Daejeon, Korea.

Yongwha Chung (phone: + 82 41 860 1343, email: ychungy@korea.ac.kr) is with Department of Computer Information, Korea University, Chungnam, Korea.

Sung Bum Pan (email: sbpan@chosun.ac.kr) is with Department of Information, Control and Instrumentation Engineering, Chosun University, Gwangju, Korea.

and confidentiality of the fingerprint images transmitted is to employ standard cryptographic techniques [5], [6]. This approach can work for the communication between the client and server. However, a typical fingerprint sensor either does not have a processor or it has a low-end, embedded processor. In this paper, we assume the sensor has a low-end processor. Thus, it may not be possible for such low-end processors to apply the standard cryptographic techniques to the full fingerprint images in real-time.

To reduce the computational workload of the resource-constrained sensor and to guarantee the integrity and confidentiality of the fingerprint image during the communication between a sensor and a client, we apply a standard encryption algorithm to a nonce, instead of applying it to the image itself. Also, to guarantee the confidentiality of the fingerprint image and to reduce the encryption time, we have developed an image-based selective bitplane encryption algorithm for the resource-constrained sensor. We select the least significant bit (LSB) of each pixel in the fingerprint image (called an LSB bitplane) as random noise and take the exclusive-OR of the LSB bitplane and all the pixels of the fingerprint image. Because an opponent cannot recover the original fingerprint image without knowledge of the LSB bitplane, we need to encrypt further the LSB bitplane by using a shared session key. As the sensor and client share the same session key, the client can recover the original fingerprint image by decrypting the encrypted LSB bitplane and by then applying the same exclusive-OR operation.

With this image-based selective bitplane encryption, our protocol can reduce the computational workload significantly, and can guarantee the integrity and confidentiality of the fingerprint image from an opponent without any leakage of the fingerprint ridge information. Based on the experimental results, we confirm that the proposed protocol can guarantee the integrity and confidentiality of the fingerprint images and provide real-time performance on embedded processors.

The rest of the paper is structured as follows. Section II explains the overview of a typical fingerprint verification system and the attack points in remote applications, while section III describes the proposed protocol based on a challenge-response and image-based selective encryption. The proposed selective bitplane encryption algorithm using LSB information is described in detail in section IV. The implementation details and performance evaluation are described in section V. Finally, conclusions are given in section VI.

II. Background

1. Fingerprint Verification

The fingerprint verification system shown in Fig. 1 has two phases: enrollment and verification. In the off-line enrollment phase, an enrolled fingerprint image for each user is preprocessed, and the minutiae are extracted and stored in a server. In the on-line verification phase, the input minutiae are compared to the stored template, and the result of the comparison is returned.

In general, there are three steps involved in the verification phase [2]: image pre-processing, minutiae extraction, and minutiae matching. Image pre-processing refers to the refinement of the fingerprint image against the image distortion obtained from a fingerprint sensor. Minutiae extraction refers to the extraction of features from the fingerprint image. After this step, some of the minutiae are detected and stored into a database, which includes the position, orientation, and type (ridge ending or bifurcation) of the minutiae. Based on the minutiae, the input fingerprint is compared with the enrolled database in the minutiae matching step.

Note that the image pre-processing and minutiae extraction steps require a lot of integer computations, and the computational workload of both steps occupies 96% of the total workload of fingerprint verification [7]. Thus, it is reasonable to assign the time-consuming steps to the client, rather than to the resource-constrained sensor. This kind of task assignment can be found in the combination of a smart card and card reader [8], [9]. That is, the time-consuming steps are assigned to the more powerful card reader, rather than the resource-constrained smart card.

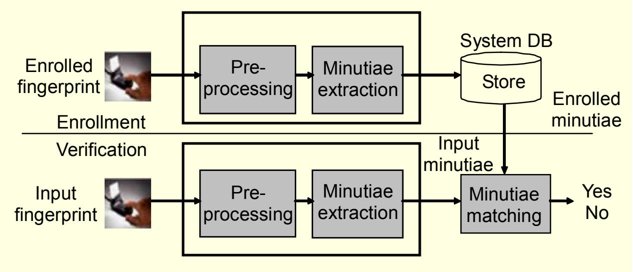

Fig. 1. Illustration of the Fingerprint Verification.

2. Attack Points

As shown in Fig. 2, many of the possible attacks in fingerprint verification systems were identified [4]: ① attack at the sensor, ② attack on the channel between the sensor and the feature extractor, ③ attack on the feature extractor, ④ attack on the channel between the feature extractor and the matcher, ⑤ attack on the matcher, ⑥ attack on the system database, ⑦ attack on the channel between the system database and the matcher, and ⑧ attack on the channel between the matcher and the application requesting the verification. Details of these attacks are explained in [4].

Note that attacks ②, ④, and ⑦ are launched against the communication channels; they are similar in nature and can be collectively called replay attacks [2]. In this paper, we focus on these replay attacks, especially attack ②, where the resource-constrained sensor is involved. In general, the large-scale adoption of a security model based on an open network requires the resolution of several practical problems relative to security and information reserve issues. In order to have a high acceptability, the mechanism of the basic tasks should be easy, fast, and inexpensive [6].

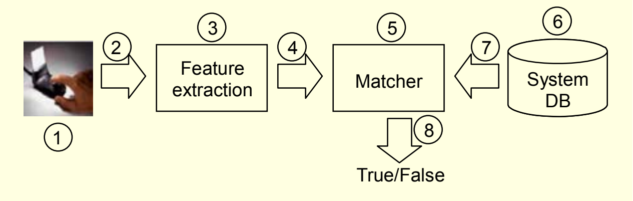

Fig. 2. Illustration of the attack points [4].

3. Related Works

A typical approach to protecting fingerprint information is to employ standard cryptographic techniques such as an encryption, hash, and digital signature. For instance, standard cryptographic techniques have been employed to protect biometric information in open networks [6], [10] and communication channels between smart cards and card readers [11].

Although many standard cryptographic techniques could be applied independently of the biometrics, some fingerprint-specific techniques have been recently reported. For instance, invisible watermarking of fingerprint images may assure the database administrators that all of the images in the database are authentic and have not been tampered with by an opponent [12]. Such mechanisms of protection reduce the risk of unauthorized insertion of spurious records into the database. Invisible watermarking can also be used to protect biometric information over insecure communication channels by inserting it at the sender and verifying it at the receiver [13].

Treating a fingerprint as a key has been reported recently. For instance, by combining an error correcting code with the *IrisCode*, a canonical iris feature can be generated and standard hash functions can be used [14]. Furthermore, because only a hash of the iris feature and the error correcting digits are stored in the database record, the original iris cannot be reconstructed.

Also, several researches [15]-[18] have been published on image and video encryption including selective (or partial) encryption methods. R. Pfarrhofer and others [15] proposed a selective encryption of Joint Bi-level Image Experts Group (JBIG) encoded visual data. This exploited the interdependencies among resolution layers in the JBIG hierarchical progressive coding mode and showed some attack resistance against common image processing attacks and replacement attacks. In particular, it modified the selective bitplane encryption algorithm of M. Podesser and others [18]. Also, S. Lian and others [16] proposed a selective encryption scheme for JPEG2000 encoded images, which encrypts only parts with high sensitivity while others are left unencrypted. Z. Liu and others [17] presented a lightweight video encryption algorithm based on the Huffman error diffusion that is capable of working on mobile devices such as pocket PCs and mobile phones.

Because most of these researches are to encrypt general multimedia contents, they use either selective bitplane encryption including the most significant bit (MSB) information or selective frequency band encryption including the DC value after executing some transformation such as a discrete cosine transform (DCT) and fast Fourier transform (FFT).

In this paper, we focus on protecting fingerprint information over insecure networks by using standard cryptographic techniques on the selective bitplane and simple bit-wise operations only, especially for attack ② shown in Fig. 2.

III. Secure and Efficient Transmission of Fingerprint Images

The goal of this research is to transmit fingerprint images securely with a reduced amount of computational workload. Thus, a resource-constrained sensor can transmit fingerprint images securely in real-time by employing the proposed protocol. In this paper, we assume that the sensor is connected physically to a specific client, and both the sensor and the client share the same master key for symmetric encryption.

1. Challenge-Response Protocol

With current technologies, a resource-constrained sensor cannot apply all the security components such as an encryption module, a digital signature module, a hash function module, and a random number generator to the full fingerprint image in real-time. Furthermore, implementing asymmetry key encryptions such as Rivest-Shamir-Adleman (RSA) is infeasible for the resource-constrained sensor as it is slow. Symmetry key encryptions such as Advanced Encryption Standard (AES), on the other hand, can be much faster and simpler to implement. To guarantee the integrity and confidentiality of the fingerprint image and execute the required security components in real-time on the resource-constrained sensor, we propose a simple and effective protocol in terms of the computational workload and security level. We will explain how we can achieve that goal with a standard symmetric encryption algorithm. For the purpose of explanation, we define first the following notations:

N : a nonce generated randomly in the client and used as a 'challenge'
Bio : biometric data such as a fingerprint
K_m : a master key shared by both the sensor and the client
$f_1(K_m, N)$: a simple function to generate a session key with K_m and N
K_s : a shared session key generated for each transmission
$f_2(N)$: a simple function to generate a 'response'

Figure 3 illustrates an example of a simple challenge-response protocol to transmit the biometric data from a sensor to a client using a standard symmetric key algorithm only. As we mentioned, we assume that both the sensor and the client share a master key, K_m. For instance, the master key can be distributed when the sensor is installed on the client. As shown in Fig. 3, the client sends first a nonce N encrypted with K_m. After receiving this message, the sensor generates a session key K_s by using a simple function, $f_1(K_m, N)$. Then, the sensor encrypts the input fingerprint image with the session key. The sensor also computes a response and encrypts it with the session key, and sends $\mathrm{E}_{Ks}(Bio)$ and $\mathrm{E}_{Ks}(f_2(N))$ back to the client. Because the client can also generate the session key, the client can confirm the source of the message.

This simple protocol requires only two steps of communication and guarantees both the integrity and confidentiality of the fingerprint image transmitted. However, the sensor may not execute this protocol in real-time if it encrypts the full fingerprint image. To complete the protocol in real-time on the resource-constrained sensor, we need to further reduce the computational workload of encrypting fingerprint images.

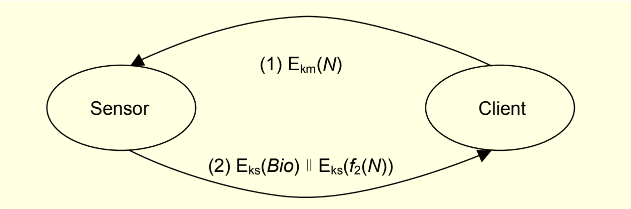

Fig. 3. Illustration of the challenge-response protocol.

2. Image-Based Selective Encryption

To reduce the computational workload of encrypting fingerprint images further, two approaches are possible for image-based selective encryption that encrypts part of the fingerprint image instead of the full fingerprint image: selective spatial encryption and selective bitplane encryption**.** Note that we consider spatial-domains rather than frequency-domains of fingerprint images because the transform to a frequency-domain is a time consuming operation for the resource-constrained processors.

Selective spatial encryption is a straightforward approach to partitioning the image and encrypting the central region of the partition, instead of the full fingerprint image. Because the central region of a fingerprint image may include a lot of significant information of the fingerprint such as cores and deltas, this approach may be considered an efficient way to conceal the fingerprint information from an opponent. However, an opponent can intercept multiple selectively encrypted messages and generate a composed image by using the mosaic technique [2]. Because the composed image looks similar to the original fingerprint image, this selective spatial encryption cannot provide the confidentiality of the fingerprint image. Figure 4 shows the results of selective spatial encryption and a composed image. Note that, to show the effect of the composition, the encrypted central region is shown as a white box.

The other approach is selective bitplane encryption [18]. In general, a fingerprint image is given in an 8-bit/pixel (bpp) precision. We consider the 8 bpp data in the form of 8 bitplanes, where each bitplane is associated with a position in the binary representation of the pixels. The selective bitplane encryption approach is to encrypt a subset of the bitplanes only, starting with the bitplane containing the MSB of the pixels.

Figure 5 illustrates the selective bitplane encryption approach.

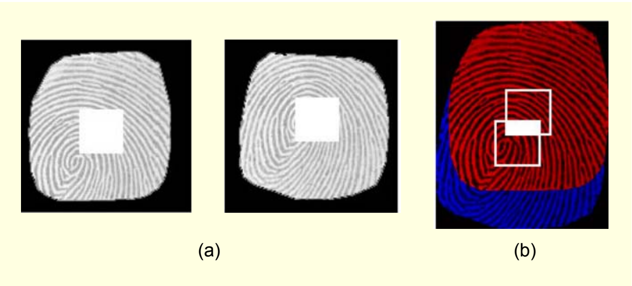

Fig. 4. (a) Results of the selective spatial encryption, and (b) a composed image by using the mosaic technique.

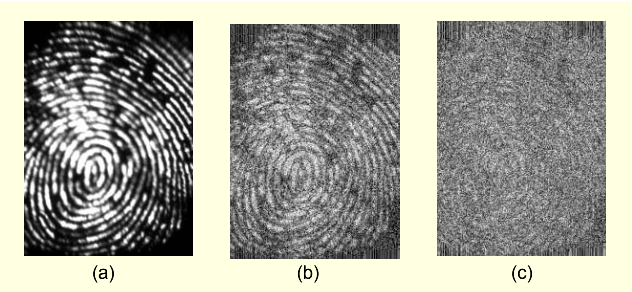

Fig. 5. Results of the selective bitplane encryption: (a) original fingerprint image, (b) result of encrypting the MSBs only, and (c) result of encrypting the MSBs and the next significant bits.

Figure 5(a) represents an original fingerprint image, Fig. 5(b)represents a distorted image after encrypting one bitplane, and Fig. 5(c) represents a distorted image after encrypting two bitplanes. As shown in Fig. 5, this approach can avoid the mosaic attack by distorting the whole pixel values of a fingerprint image. Also, compared to the full bitplane encryption, the computational workload of this encryption is reduced significantly.

Note that some structural information is still visible after encrypting the MSB only, whereas encrypting two bitplanes leaves no useful information. However, the two-bitplane encryption is not safe for fingerprint images either. For instance, a replacement attack [18] replaces the encrypted bitplanes by constant 0's and compensates for the decreased luminance by adding 96 to each pixel. As shown in Fig. 6, the replacement attack can reveal the fingerprint ridge information even from the two-bitplane encryption. Details of the replacement attack can be found in [18].

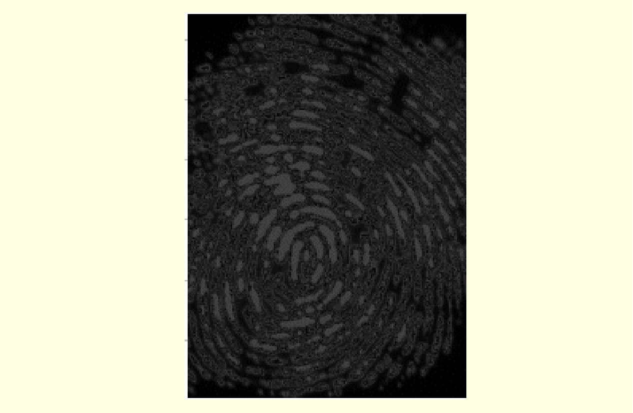

Fig. 6. Result of replacing the MSBs and the next significant bits of Fig. 5(c) with constant 0's.

IV. A Proposed Selective Bitplane Encryption Algorithm

To solve these problems, we propose another selective bitplane encryption algorithm that uses the LSB as the selective bitplane. Generally, it is possible to discover correlations between the LSB and other bitplanes in the images generated by graphic tools such as Photoshop, Paintshop, and so on. However, the LSBs of the images are not correlated with other bitplanes if the images are acquired by various sensors such as a digital camera, scanner, and other devices [19], [20] (note that the MSBs are still correlated with other bitplanes even for those images).

To understand the characteristics of the LSBs acquired from fingerprint sensors, we measured the difference between the two LSBs of the fingerprint images acquired sequentially from a single touch to the sensor. Even though each pair in Figs. 7(a) and 7(b) and Figs. 7(c) and 7(d) seems to show similar values, we can see the difference between the two LSBs in Fig. 7(e) (note that the difference should be 0 and shown as black if the two LSBs are the same). That is, even if two fingerprint images are acquired sequentially from a single touch to the sensor, the LSB strings of the two fingerprint images are quite different. Furthermore, the LSB looks similar to a random number field [19], [20] and is more suitable for our algorithm than the MSB, although the LSB contains the least significant information of the image. Thus, it is natural to select the LSB as our random noise, and the LSB itself needs to be protected from an opponent. In the following, we will describe the proposed selective bitplane encryption algorithm consisting of two steps: image distortion and LSB encryption.

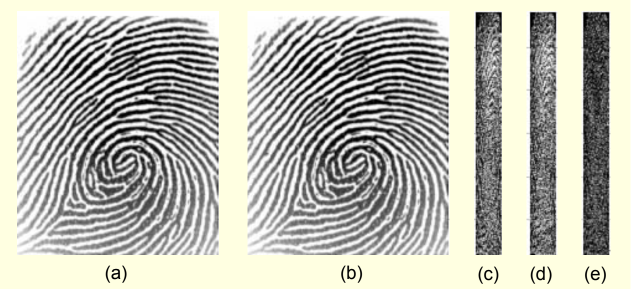

Fig. 7. LSB characteristics of the fingerprint images acquired sequentially by the single touch to the sensor: (a) first-acquired fingerprint image; (b) second-acquired fingerprint image; (c) LSB bitplane of (a); (d) LSB bitplane of (b); and (e) difference between (c) and (d).

In image distortion, we distort the full fingerprint image by using very simple operations. For each pixel, we select its LSB as a random noise and generate the LSB bitplane. Then, we take a simple exclusive-OR of the LSB bitplane and all the pixels of the fingerprint image, as shown in Fig. 8. In this design, the LSB bitplane works as a one-time pad, and the length of the LSB bitplane is too long to predict. Without knowledge of the LSB bitplane, an opponent cannot recover the ridge structure of the fingerprint image from the result of image distortion. Therefore, in LSB encryption, we only need to encrypt further the LSB bitplane by using a shared session key. As the client and sensor share the same session key, the client can recover the original fingerprint image by decrypting the encrypted LSB bitplane and then applying the same exclusive-OR operation shown in Fig. 8.

Figure 9 shows the proposed protocol to transmit fingerprint images by using the image-based encryption. We call this protocol an image-based selective bitplane encryption protocol. The only difference between Figs. 9 and 3 is the second step. Instead of encrypting the full fingerprint image, the sensor generates a distorted fingerprint image $Bio \oplus LSB$ in image distortion. Then, the sensor encrypts the LSB with the session key, and sends $Bio \oplus LSB \| \mathrm{E}_{Ks}(LSB) \| \mathrm{E}_{Ks}(f_2(N))$ to the client.

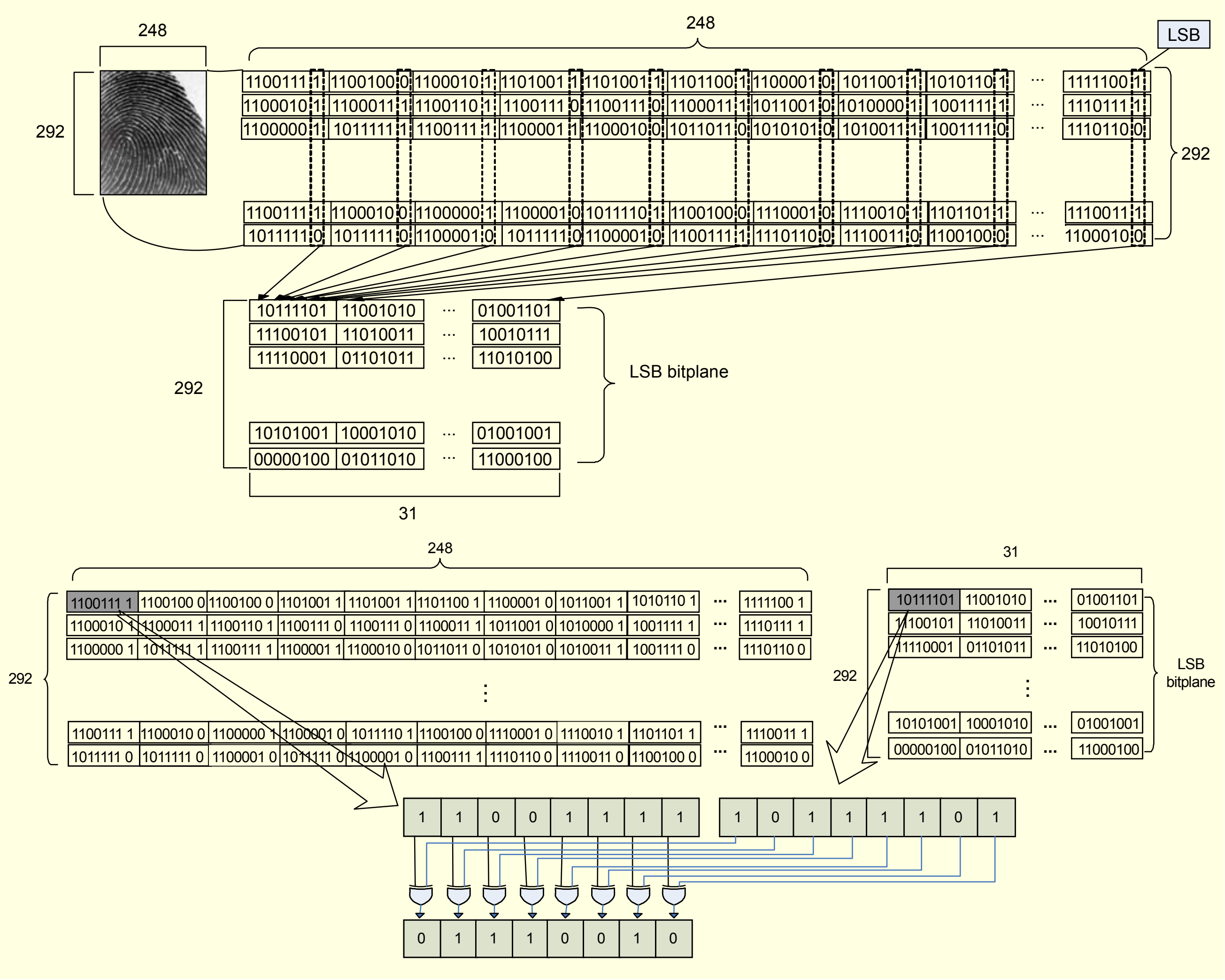

Fig. 8. Illustration of image distortion using LSB.

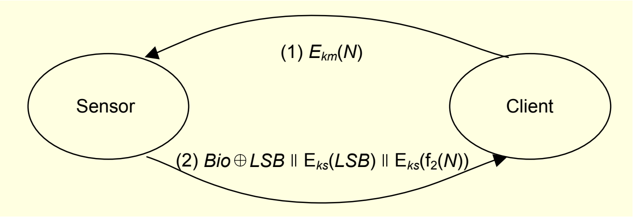
(1) $E_{km}(N)$
Sensor
Client
(2) $Bio \oplus LSB \parallel E_{ks}(LSB) \parallel E_{ks}(f_2(N))$

Fig. 9. Illustration of the proposed protocol.

V. Implementation Details and Performance Evaluation

For the purpose of evaluation, we used the standard symmetric encryption algorithm (128-bit *AES*) to guarantee both the integrity and confidentiality, and we have tested our selective bitplane encryption algorithm on the fingerprint images captured by using three types of fingerprint sensors: TBS220 [21], AT77C101B [22], and OPU01M [23]. Characteristics of these fingerprint sensors are explained in Table 1. Because the size of the fingerprint image captured by AT77C101B was variable, we have reconstructed the fingerprint image to the size of 288×432.

Table 2 shows the system specifications of the secure fingerprint sensor that we have implemented for the secure fingerprint transmission between the sensor and the client. The fingerprint sensor employed a 400 MHz CPU with 2 Mbytes of ROM and 16 Mbytes of RAM. Note that the selected CPU is one of the widely used CPUs in real door-lock systems. Also, two types of fingerprint sensors such as TBS220 and AT77C101B were employed as shown in Figs. 10(a) and 10(b), respectively. However, the encryption time for the fingerprint image captured by the OPU01M sensor was measured after downloading to our secure fingerprint sensor. Figure 10(c) shows a photograph of the evaluation environment including the client module, the debugging tool, and our secure fingerprint sensor.

Figure 11 shows an input image, the LSB bitplane, and the

Table 1. Characteristics of three types of fingerprint sensors.

	Testech TBS220 [21]	ATMEL AT77C101B [22]	Nitgen OPU01M [23]
Sensor type	Light emitting touch	Thermal array sweep	Optical
Resolution	500 dpi	500 dpi	500 dpi
Image size	320×440	Variable	248×292

Table 2. System specifications of the secure fingerprint sensor.

CPU	16-bit RISC processor (ADSP-BF531 [24], 400 MHz)
ROM	FLASH ROM 2 MBytes
RAM	SDRAM 16 MBytes
Sensors	AT77C101B, TBS220

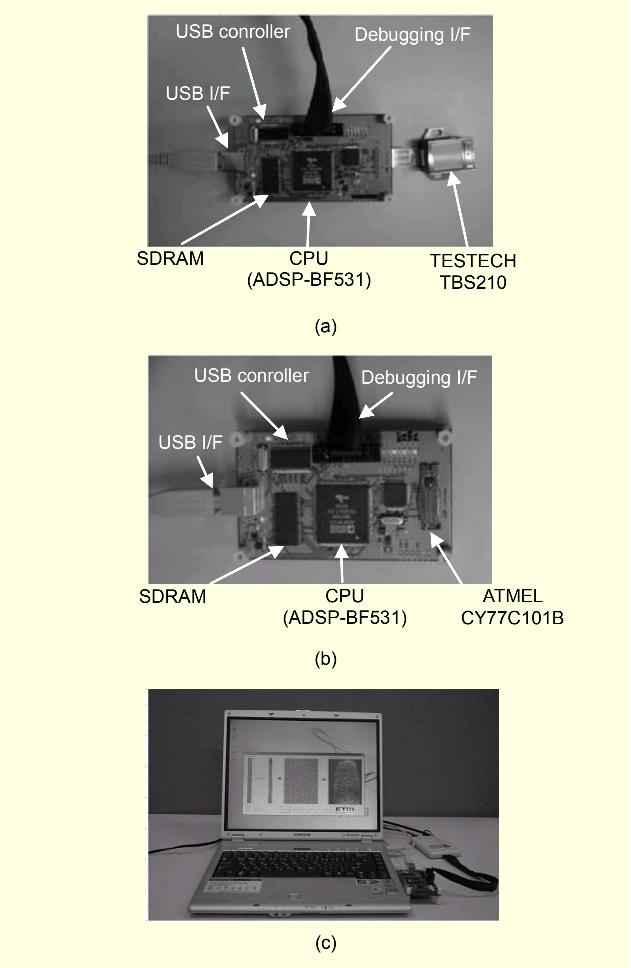

Fig. 10. Photograph of the evaluation system: (a) secure fingerprint sensor employing TBS220, (b) secure fingerprint sensor employing AT77C101B, and (c) simulation environment.

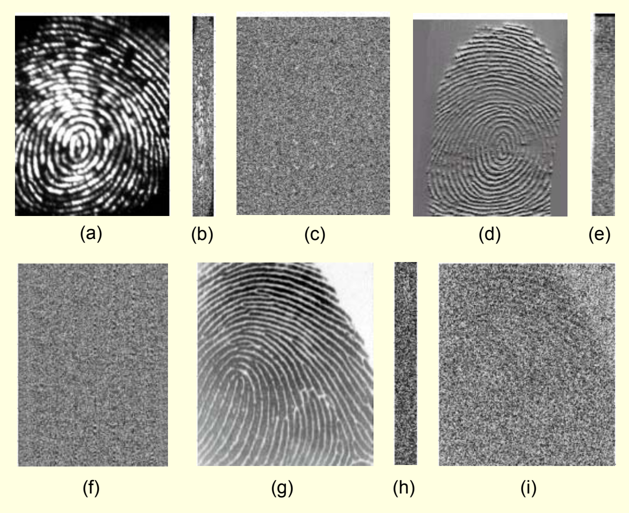

Fig. 11. Results of the proposed selective encryption: (a) input image captured by TBS220; (b) LSB bitplane of (a); (c) result of the sistortion of (a), (d) input image captured by AT77C101B; (e) LSB bitplane of (d); (f) result of the distortion of (d); (g) input image captured by OUP01M; (h) LSB bitplane of (g); and (i) result of the distortion of (g).

result of the proposed encryption algorithm. Figures 11(a), 11(d), and 11(g) represent the input images captured by TBS220, AT77C101B, and OPU01M, respectively. As shown in Figs. 11(b), 11(e) and 11(h), the LSB bitplane is smaller than a full fingerprint image by a factor of eight. The LSB bitplane looks like random noises and can vary at every fingerprint acquisition. Thus, unlike selective spatial encryption, it is difficult for an opponent to discover the ridge structure of the fingerprint image even if he obtains multiple images distorted by the LSB bitplane. Also, the results of our selective bitplane encryption algorithm using the LSB information leave no useful information.

The proposed algorithm is safe under a known attack (for example, a replacement attack) because every bitplane is modified by the exclusive-OR operation before encrypting the LSB bitplane. Although the result of the distortion shown in Fig. 11(i) seems to show some ridge information, for example, the attack cannot get any useful information from it. Figure 12 shows the results of the attack on the distorted image shown in Fig. 11(i). Compared to Fig. 6, which shows some useful information through the attack, Fig. 12 does not show any useful information. Also, the results of the attack on Figs. 11(c) and 11(f) were similar to the results shown in Fig. 12.

Finally, we summarize the evaluated execution times of the proposed selective encryption and the full encryption in Table 3. As shown in Table 3, the LSB encryption can reduce the execution time of the full encryption by a factor of six and can be executed in real-time. In spite of requiring an additional operation for image distortion, both the LSB extraction and the

XORing operations are simple bit-wise and shift operations. If the current pixel values are 10101100 and 10101101, for example, we can extract the LSB using a simple code as follows.

10101100 & 00000001 = 00000000 (LSB = 0)
10101101 & 00000001 = 00000001 (LSB = 1)

In particular, the results of Table 3 show that the execution time of the image distortion is negligible even with low-power, embedded-processor environments.

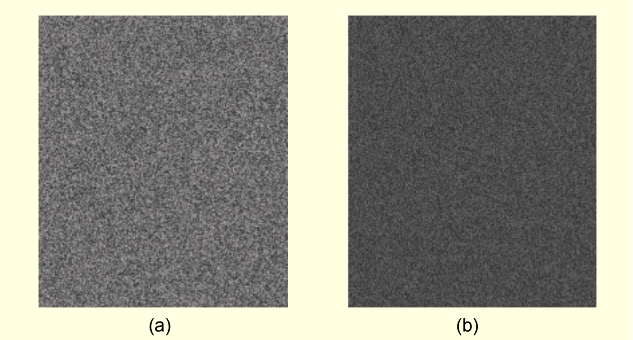

Fig. 12. Results of replacement attack: (a) result of replacing the MSBs of Fig. 11(i) with constant 0's, and (b) result of replacing the MSBs and the next significant bits of Fig. 11(i) with constant 0's.

Table 3. Encryption times of fingerprint images under ADSP-BF531 [24] (400 MHz).

			TBS220	AT77C101B	OPU01M
Selective encryption (proposed)	Image distortion	LSB extraction	0.06 s	0.05 s	0.03 s
		XORing	0.09 s	0.08 s	0.05 s
	LSB encryption		0.58 s	0.51 s	0.30 s
	Total time		0.73 s	0.64 s	0.38 s
Full encryption (typical)			4.69 s	4.15 s	2.41 s

VI. Conclusions

The use of biometrics in a network is expected to increase widely in conjunction with other techniques such as cryptography. In this paper, a secure and efficient protocol has been proposed to transmit fingerprint images from a fingerprint sensor to a client. To guarantee both the integrity and the confidentiality of the fingerprint images transmitted, only a standard encryption algorithm has been employed. To reduce the computational workload on the resource-constrained sensor, however, we applied the encryption algorithm to a nonce and to the LSB bitplane of the fingerprint image.

The proposed selective encryption shows promising results. Because the full fingerprint image is distorted by applying the simple exclusive-OR operations with the LSB bitplane, and then the LSB bitplane itself is encrypted, our protocol can reduce the execution time of full encryption by a factor of six. Therefore, it can guarantee both the integrity and the confidentiality of the fingerprint image between a sensor and a client in real-time. Note that our protocol can also be applied to other types of biometric data such as facial features, iris, or vein structures, although in this paper, we considered fingerprints only.

References

[1] A. Jain, R. Bole, and S. Panakanti, *Biometrics: Personal Identification in Networked Society*, Kluwer Academic Publishers, 1999.

[2] D. Maltoni, D. Maio, A. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*, Springer, 2003.

[3] R. Bolle, J. Connell, and N. Ratha, "Biometric Perils and Patches," *Pattern Recognition*, vol. 35, 2002, pp. 2727-2738.

[4] N. Ratha, J. Connell, and R. Bolle, "An Analysis of Minutiae Matching Strength," *LNCS 2091 - Proc. of AVBPA*, 2001, pp. 223-228.

[5] W. Stallings, *Cryptography and Network Security: Principals and Practice*, Pearson Ed. Inc., 2003.

[6] D. Maio and D. Maltoni, "A Secure Protocol for Electronic Commerce Based on Fingerprints and Encryption," *Proc. of Conf. on Systems, Cybernetics, and Informatics*, 1999, pp. 519-525.

[7] D. Moon, Y. Gil, S. Pan, Y. Chung, and H. Kim, "Performance Analysis of the Match-on-Card System for the Fingerprint Authentication," *Proc. of International Workshop on Information Security Applications*, 2001, pp. 449-459.

[8] Y. Moon, H. Ho, K. Ng, S. Wan, and S. Wong, "Collaborative Fingerprint Authentication by Smart Card and a Trusted Host," *Electrical and Computer Engineering*, vol. 1, 2000, pp. 108-112.

[9] S. Pan, Y. Gil, D. Moon, Y. Chung, and C. Park, "A Memory-Efficient Fingerprint Verification Algorithm using a Multi-Resolution Accumulator Array for Match-on-Card," *ETRI Journal*, vol. 25, no. 3, 2003, pp. 179-186.

[10] A. Jain, S. Prabhakar, and A. Ross, "Biometrics-Based Web Access," *Technical Report, Michigan State U.*, 1998.

[11] L. Rila and C. Mitchell, "Security Analysis of Smartcard to Card Reader Communications for Biometric Cardholder Authentication," *Proc. of CARDIS*, 2002, pp. 19-28.

[12] M. Yeung and S. Pankanti, "Verification Watermarks on Fingerprint Recognition and Retrieval," *Journal of Electronic Imaging*, vol. 9, no. 4, 2002, pp. 468-476.

[13] A. Jain and U. Uludag, "Hiding Fingerprint Minutiae in Images," *Proc. of AutoID*, 2002, pp. 97-102.

[14] G. Davida, Y. Frankel, and B. Matt, “On Enabling Secure Applications through Off-Line Biometric Identification,” *Proc. of Symp. on Privacy and Security*, 1998, pp. 148-157.

[15] R. Pfarrhofer and A. Uhl, “Selective Image Encryption using JBIG,” *LNCS 3677 - Proc. of CMS*, 2005, pp. 98-107.

[16] S. Lian, J. Sun, D. Zhang, and Z. Wang, “A Selective Image Encryption based on JPEG2000 Codec,” *LNCS 3332 - Proc. of PCM*, 2004, pp. 65-72.

[17] Z. Liu, X. Li, and Z. Dong, “A Lightweight Encryption Algorithm for Mobile Online Multimedia Devices,” *LNCS 3306 - Proc. of WISE*, 2004, pp. 653-658.

[18] M. Podesser, H. Schmidt, and A. Uhl, “Selective Bitplane Encryption for Secure Transmission of Image Data in Mobile Environments,” *Proc. of the 5th IEEE Nordic Signal Processing Symposium*, 2002.

[19] R. Gonzalez, *Digital Image Processing*, Addison Wesley, 1992.

[20] M. Droogenbroeck and R. Benedett, “Techniques for a Selective Encryption of Uncompressed and Compressed Images,” *Proc. of ACIVS*, 2002, pp. 90-97.

[21] Testech, http://www.testech.co.kr

[22] ATMEL, http://www.atmel.com

[23] NiGen, http://www.nitgen.com

[24] Analog Device, http://www.analog.com

Daesung Moon received the MS degree from Busan National University, Korea, in 2001. He joined the Electronics and Telecommunications Research Institute (ETRI), Korea, in 2000, where he is currently a Senior Member of the engineering staff in the Biometric Technology Research Team. His research areas are biometrics, image processing, and security.

Yongwha Chung received the BS and MS degrees from Hanyang University, Korea, in 1984 and 1986. He received the PhD degree from the University of Southern California, USA in 1997. He worked for ETRI from 1986 to 2003 as a Team Leader. Currently, he is an Associate Professor in the Department of Computer Information, Korea University. His research interests include biometrics, security, and performance optimization.

Sung Bum Pan received the BS, MS, and PhD degrees in electronics engineering from Sogang University, Korea, in 1991, 1995, and 1999. He was a Team Leader in the Biometric Technology Research Team of ETRI from 1999 to 2005. He is now a Full-time Instructor at Chosun University. His current research interests are in biometrics, security, and VLSI architectures for real-time image processing.

Kiyoung Moon received the BS and MS degrees in electronics engineering in 1986 and 1989 from Kyungpook National University, Korea. He received the PhD degree in Computer Science from Chungnam National University, Korea in 2006. He has been a senior member of the technical staff in ETRI since 1994, where he is currently working as the Team Leader of the Biometric Technology Research Team. His research interests include biometrics, XML security, distributed systems, application security, and transactions.

Kyo Il Chung received the BS, MS, and PhD degrees in electronic engineering from Hanyang University in 1981, 1983, and 1997. He joined ETRI in 1982 and has been involved with COMSEC systems. Currently, he is a principal member of the engineering staff and his role is the Director of the Information Security Infrastructure Research Group. His research interests are in IC cards, RFID, biometrics, and information warfare.