PRIVACY PROTECTION OF FINGERPRINT DATABASE USING LOSSLESS DATA HIDING

Sheng Li and Alex C. Kot

School of Electrical and Electronic Engineering
Nanyang Technological University, Singapore 639798
Email: {lish0016,eackot}@ntu.edu.sg

ABSTRACT

In this paper, we introduce a fingerprint authentication system for protecting the privacy of the fingerprint template stored in a database. The template, which is a binary fingerprint image after thinning, will be embedded with private personal data in the user enrollment phase. In the user authentication phase, these hidden personal data can be extracted from the stored template for verifying the authenticity of the person who provides the query fingerprint. A novel lossless data hiding scheme is proposed for a thinned fingerprint. By adopting "embeddability criterion", data is hidden into the template by just adding some boundary pixels in the template. These boundary pixels can be extracted and removed to reconstruct the original thinned fingerprint so that fingerprint matching accuracy is not affected. Compared with using existing binary image data hiding techniques, our scheme has a better performance for a thinned fingerprint.

Keywords— fingerprint, database, privacy, lossless, data hiding

1. INTRODUCTION

Nowadays, biometric techniques such as fingerprint, face, signature, iris and voice, are widely used in authentication systems. While biometric data provides uniqueness, it cannot be lost or forgotten, further, it is difficult to be shared or distributed. Hence, biometrics-based authentication systems are more useful and reliable than traditional ones which are mainly based on passwords.

In general, biometrics needs to be stored in a database for subsequent authentication. However, templates stored in the database may be stolen, removed or modified. The stolen templates with user identity are difficult to be replaced like passwords, which also create difficulties for authorized person to enter the system. Thus, biometric templates should be stored in the database such that both the security of the template and the privacy of the user are not compromised under various attacks.

In the past few years, various data hiding techniques are proposed to hide data in grayscale biometric images or color biometric images for the authenticity of the biometric data or for secure transmission [1, 2, 3]. We focus in this paper on using data hiding to ensure the privacy of the fingerprint template stored in a database. Instead of keeping grayscale fingerprint images for authentication, we store thinned fingerprint image files which are much smaller in size and all key features are maintained. Also, extracting minutiae features from a thinned fingerprint image is much faster. Keeping minutiae alone as the template won't be sufficient to reconstruct the ridge valley of the original fingerprint. In order to preserve all the features of the fingerprint template in data embedding, a novel lossless data hiding method for thinned fingerprint is proposed. Using the proposed scheme, we are able to reconstruct the original thinned fingerprint from the marked-fingerprint without data extraction. Therefore, the fingerprint matching accuracy is not affected after data embedding.

The organization of the paper is as follows. Section 2 gives a brief review on the data hiding techniques for binary image. Section 3 introduces the proposed method including the fingerprint authentication system and the lossless data hiding scheme. Section 4 presents the experimental results and discussions, followed by our conclusions in the last section.

2. DATA HIDING IN BINARY IMAGE

Most binary image data hiding techniques focus on finding appropriate "flipping" locations to minimize the visual distortion caused by data hiding [4, 5, 6]. Wu and Liu [4] compute flippablity scores of pixels in binary image by observing the smoothness and connectivity. Pixels with high flippablity scores have the priority for data embedding. Yang and Kot [5] propose a pattern based data hiding technique which is able to preserve the connectivity of pixels in a local neighborhood. The flippability of a pixel is determined by some criteria in a 3×3 window centered at the pixel. Blocks containing flippable pixels are identified as embeddable blocks and chosen for data embedding. An interlaced morphological binary wavelet transform is proposed in [6] to track the shifted edges according to the observation that flipping an edge pixel equals to shifting the edge location one pixel horizontally and vertically. Suitable locations for data embedding are identified by utilizing the relationship between the coefficients obtained from the proposed transform. These schemes have a good visual quality, however, they cannot recover the original image after data embedding.

In order to reconstruct the original image after data embedding, some lossless binary image data hiding techniques are proposed in [7, 8, 9]. Tsai *et al.* [7] propose a lossless data hiding technique based on pair-wise logical computation. In

978-1-4244-7493-6/10/$26.00

data embedding, proper host image bit pairs are chosen based on some criteria, one reference bit and one secret bit are hidden into each chosen bit pair. In the receiving side, the stego bit pairs are identified and the secret bits are extracted. Furthermore, the host image bit pairs can be recovered from the stego bit pairs. Ho *et al*. [8] propose a approach based on pattern substitution in the difference image. Two groups of patterns are utilized in their technique, i.e., patterns with high frequency and patterns with low frequency. During data embedding, patterns with high frequency will be changed to their corresponding low frequency patterns or vice versa. The locations of the patterns with low frequency will be recorded and used for reconstructing the original image. Similar to [8], Xuan *et al*. [9] make use of the statistic features of run-length histogram in the binary image. A black run-length histogram pair is chosen for data embedding. These lossless data hiding schemes suffer from poor visual quality when applied to the thinned fingerprint, as they are not able to preserve the connectivity of the pixels in a local neighborhood in data embedding.

In this paper, we propose a novel lossless data hiding scheme for thinned fingerprint with a good visual quality. In this scheme, data is hidden into the thinned fingerprint by adding some boundary pixels. These boundary pixels can be located and removed to reconstruct the original thinned fingerprint, thus the fingerprint matching performance is not affected after data embedding.

3. THE PROPOSED METHOD

3.1. The fingerprint authentication system

Figure 1 shows the fingerprint authentication system considered in this paper. In the user enrollment, an original thinned fingerprint image **F** is produced after some enhancement and thinning process for the input fingerprint. Note that in our system, the original thinned fingerprint refers to an image that keeps the one-pixel width skeleton of a fingerprint, i.e., no boundary pixel exists in an original thinned fingerprint. Then, the user identity **U** is hidden into **F** based on a data embedding key **K** which is generated from a key generation function (e.g. MD5) with **U** as the input. After data embedding, the fingerprint template **T** with hidden identity is stored in an online database for subsequent authentication. Meanwhile, the user identity **U** is stored in a local storage kept by the authority. In the user authentication, a query fingerprint is compared with the templates stored in online database to find a best match. If the similarity between the query fingerprint and the best match is over a certain threshold, the hidden identity **U** will be extracted from the matched template based on a query identity $\mathbf{U}'$. The authenticity of the person who provides the query fingerprint and identity is further verified by comparing **U** and $\mathbf{U}'$.

In addition, some extra user biometric data such as facial information, iris code or voice features can also be hidden into the thinned fingerprint in the user enrollment. These hidden biometric data could be extracted and served as an additional source for user authentication. For example, we can embed both the

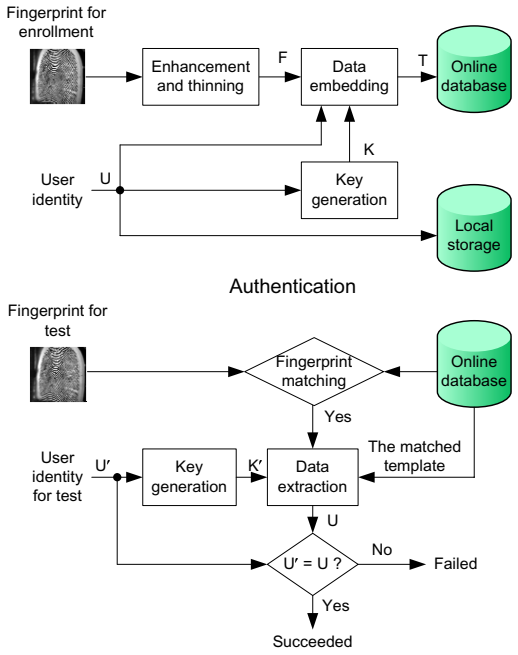


Fig. 1. The fingerprint authentication system

$(i\text{-}1, j\text{-}1)$ P_1	$(i\text{-}1, j)$ P_2	$(i\text{-}1, j\text{+}1)$ P_3
$(i, j\text{-}1)$ P_8	(i, j) P_0	$(i, j\text{+}1)$ P_4
$(i\text{+}1, j\text{-}1)$ P_7	$(i\text{+}1, j)$ P_6	$(i\text{+}1, j\text{+}1)$ P_5

Fig. 2. Notation of a single pixel and its eight neighbors

user's eigen-face coefficients and the identity into the thinned fingerprint in the user enrollment. In the user authentication, the user's face could be captured and matched with the eigen-face coefficients extracted from the matched fingerprint template.

3.2. The lossless data hiding scheme

3.2.1. Data embedding

Unlike existing block-based approach, we process the thinned fingerprint pixel by pixel in our proposed data hiding scheme. We first determine the embeddablity of each pixel, only embeddable pixels are chosen for data embedding. The embeddablity of a pixel depends on this pixel and its eight neighbors in a 3×3 block. Consider a single pixel P_0 located at ith row and jth column of the fingerprint template. We denote the eight neighbors of P_0 in Figure 2 as P_1, P_2, P_3, P_4, P_5, P_6, P_7 and P_8. In what follows, we denote "1" as the foreground (black) pixel and "0" as the background (white) pixel.

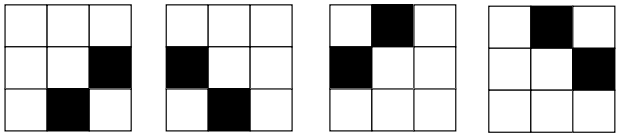
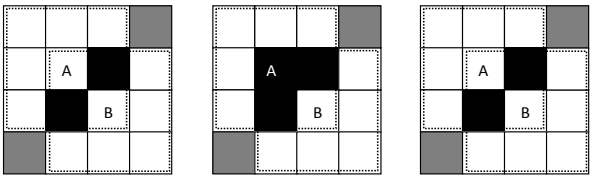


Fig. 3. Patterns satisfy the embeddability criterion



(a) (b) (c)

Fig. 4. Illustration of data embedding process (a) a 4×4 block in an original thinned fingerprint, (b) pixel B is not an embeddable pixel after pixel A is flipped, (c) pixel B is still an embeddable pixel. Pixels in gray represent "don't care" pixels.

We define the "Embeddablity Criterion" $C(i,j)$ according to

$$C(i,j) = \bar{P}_0 \cdot (\prod_{w=1}^{4} \bar{P}_{2\cdot w-1}) \cdot \sum_{w=1}^{4} (P_{2\cdot w} * P_{2\cdot w+2}) \quad (1)$$

where $P_{10} = P_2$, $\bar{P}$ is logical "not P" and "$*$" is logical "and" operator, P_0 is an embeddable pixel if $C(i,j) = 1$. Figure 3 shows four different patterns that fit such embeddablity criterion with the center pixel in these four patterns embeddable.

Given an original thinned fingerprint **F**, we develop the following data embedding steps.

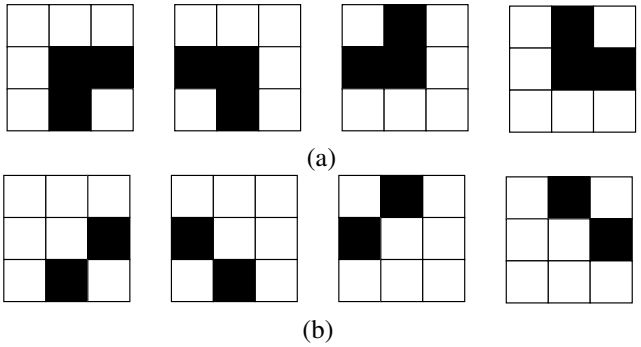
1) Select a pixel $F(i,j)$ from **F** randomly by the key **K**.
2) Determine the embeddablity of the pixel by computing $C(i,j)$.
3) Modify $F(i,j)$ according to $C(i,j)$ and the secret bit m by

$$F(i,j) = \begin{cases} m & if\ C(i,j) = 1\ \ (holds\ 1\ bit) \\ F(i,j) & if\ C(i,j) \neq 1\ \ (holds\ 0\ bit) \end{cases} \quad (2)$$

4) Repeat steps 1) to 3) until all pixels are processed.

It should be noted that the modification of $F(i,j)$ may affect the embeddability of its neighboring pixels in a 3×3 block. For example, Figure 4(a) shows a 4×4 block in the original thinned fingerprint. We know that both pixel A and pixel B are embeddable pixels from the embeddablity criterion. Suppose pixel A is the first pixel to be processed in this block during the data embedding process. If it is flipped to embed a secret bit "1", pixel B will not be embeddable anymore, as shown in Figure 4(b). On the other hand, if pixel A is used to embed a secret bit "0", pixel B is still an embeddable pixel, as shown in Figure 4(c).

Such embedding process only adds some boundary pixels to the patterns shown in Figure 3 to satisfy the embeddablity criterion. Key features are all preserved for fingerprint matching.



(a)

(b)

Fig. 5. Illustration of pattern reconstruction process, (a) patterns contain boundary pixels that are produced in data embedding process, (b) corresponding reconstructed patterns.

3.2.2. Data extraction

The original thinned fingerprint can be reconstructed by removing boundary pixels in the marked-fingerprint, which is the fingerprint template after data embedding. We define $R(i,j)$ below as the relationship between P_0 and its eight neighbors

$$R(i,j) = P_0 \cdot (\prod_{w=1}^{4} \bar{P}_{2\cdot w-1}) \cdot \sum_{w=1}^{4} (P_{2\cdot w} * P_{2\cdot w+2}) \quad (3)$$

where $P_{10} = P_2$. $R(i,j) = 1$ implies that P_0 is a boundary pixel which is produced in data embedding. All these pixels in the marked-fingerprint will be flipped to background pixels to recover the original thinned fingerprint, as shown in Figure 5.

Next we explain why the original thinned fingerprint **F** can be successfully reconstructed. According to Figure 4(b), once the embeddable pixel A is flipped to a foreground pixel, all its eight neighbors are not embeddable pixels from the embeddability criterion, i.e., these neighboring pixels cannot be modified anymore. After the flipping of pixel A, the relationship $R(i,j)$ between pixel A and its eight neighbors equals to one. As we have already discussed, such foreground pixel will be flipped to background pixel during the reconstruction process.

Given a marked-thinned fingerprint **T**, the data extraction process is summarized as follows.

1) Reconstruct the original thinned fingerprint **F** from **T**.
2) Select a pixel $F(i,j)$ from **F** by the same key used in data embedding.
3) Determine the embeddability of this pixel by computing $C(i,j)$. If the pixel is embeddable, extract one bit $m = T(i,j)$. Otherwise, no message can be extracted.
4) Modify $F(i,j)$ according to

$$F(i,j) = T(i,j) \quad (4)$$

5) Repeat steps 2) to 4) until all the pixels are processed.

The hidden data is obtained by concatenating all the extracted bits.

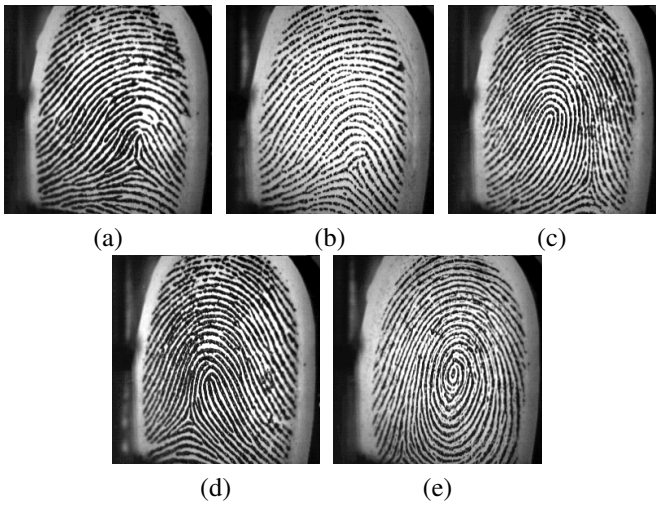


Fig. 6. Fingerprints for enrollment: (a) tented arch, (b) arch, (c) right loop, (d) left loop, (e) whorl.

4. EXPERIMENTAL RESULTS AND DISCUSSIONS

Figure 6 shows five 256×256 grayscale fingerprint images that are used as fingerprints for enrollment. These images represent main fingerprint image classes: tented arch, arch, right loop, left loop and whorl [10]. We then apply fingerprint enhancement [11] and thinning [12] for these fingerprint images to get their corresponding thinned versions. After determining the region of interest (ROI) of the thinned versions, we obtain five thinned fingerprint images with the size 181×229, as shown in Figure 7. These thinned fingerprint images serve as the original thinned fingerprint for evaluating the performance of our proposed data hiding technique.

To evaluate the performance of the proposed scheme for thinned fingerprint, we compare the data hiding capacity and visual quality of the proposed scheme with other data hiding schemes proposed by Ho *et al*. [8], Xuan *et al*. [9] and Yang *et al*. [5, 6]. Comparisons of the hiding capacity for thinned fingerprint among our proposed scheme and the four existing schemes are shown in Table 1. In order to achieve the highest capacity for Yang *et al*.'s [5, 6] schemes, 4×4 interlaced block and double processing are chosen respectively. It can be seen that our scheme has a higher capacity than the schemes proposed by Ho *et al*. and Xuan *et al*. for tented arch and arch, and it has a much higher capacity than the schemes proposed by Yang *et al*.. Note that both Ho *et al*.'s and Xuan *et al*'s schemes require correct data extraction before reconstructing the original thinned fingerprint. On the contrary, our scheme can reconstruct the original thinned fingerprint from the marked-fingerprint without data extraction.

The comparison of visual quality between our scheme and the four existing data hiding schemes for thinned fingerprint are shown in Figure 8, where 1000 bits are hidden into the right loop thinned fingerprint (the one shown in Figure 7(c)) using the proposed scheme and the other four schemes. It can be seen that the visual quality of our scheme has a great advantage over the other two lossless data hiding schemes [8, 9], and it is comparable with the schemes proposed by Yang *et al*..

Table 1. Comparison of hiding capacity among various data hiding schemes for thinned fingerprint

Original thinned fingerprint	Capacity (bits)				
	Proposed	Ho *et al*. [8]	Xuan *et al*. [9]	Yang *et al*. [5]	Yang *et al*. [6]
tented arch	1867	1588	1531	914	1252
arch	1786	1509	1523	862	1131
right loop	1740	2449	3982	1064	1255
left loop	1929	2213	3669	1094	1384
whorl	1103	2415	4001	846	1017

Next we discuss the security of our proposed system. Once the online database is stolen, the system would be vulnerable if an attacker has some prior knowledge of system, i.e., the key generation function and the data extraction algorithm. For each stolen template, the attacker can perform unlimited attempts in his computer until the guessed identity is the same as the extracted identity. However, we think it would be difficult for an attacker to perform such attack, because the majority of attackers would not realize that there is something hidden in the templates thanks to the good visual quality of our data hiding algorithm. It should be noted that stealing both the online database and the local storage is not sufficient to break the system, as the linkage between the online database and the local storage is not available to the attacker.

Generally, the length of a user identity shall be less than 20 bytes (160 bits). Therefore, our data hiding techniques have a sufficient capacity for the proposed fingerprint authentication system. In addition, some extra biometric data could also be hidden into the thinned fingerprint. For example, the extra biometric data could be 14 eigen-face coefficients using four bytes per coefficient (totally 448 bits). Note that 14 eigen-face coefficients are sufficient for face verification. These hidden biometric data may further improve the security of our system.

By using our proposed fingerprint authentication system, when the fingerprint templates stored in the online database are stolen, the impostor or intruder will find it difficult to obtain the identity of the enrolled fingerprints. The identity of these users with their stolen fingerprints will not be compromised.

5. CONCLUSION

This paper introduce a fingerprint authentication system to protect the privacy of fingerprint database. In the user enrollment, the user identity is hidden into the user's thinned fingerprint template. The template with hidden identity will then be stored in an online database for subsequent user authentication. A novel lossless data hiding scheme is also proposed for a thinned fingerprint. In this scheme, data is hidden by adding some boundary pixels in the thinned fingerprint. The added bound-

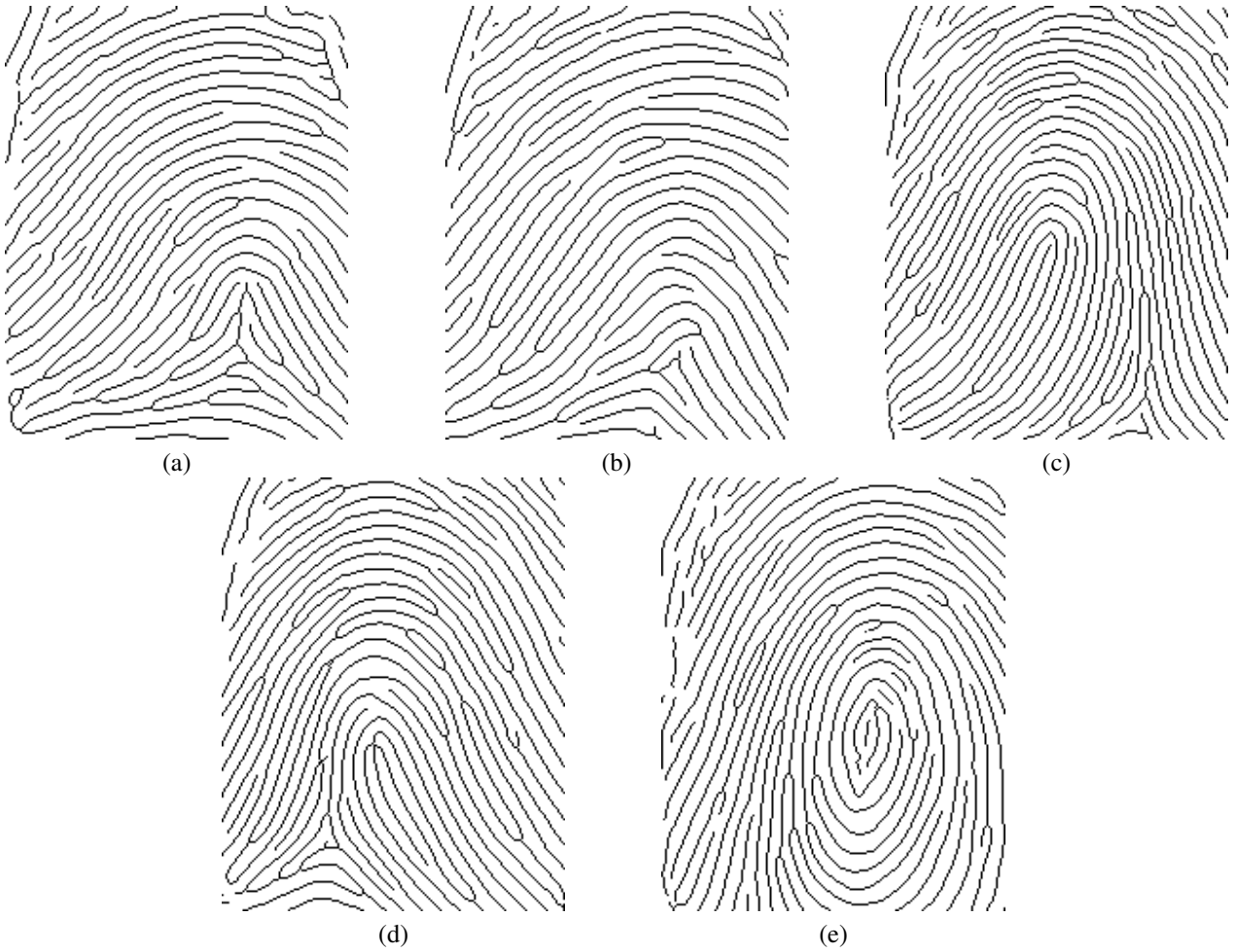


Fig. 7. Original thinned fingerprints: (a) tented arch, (b) arch, (c) right loop, (d) left loop, (e) whorl.

ary pixels can be identified and removed to recover the original thinned fingerprint, thus the fingerprint matching accuracy is not affected after data embedding. Experimental results show that our scheme has a sufficient capacity for hiding common personal data. In addition, our scheme has a better performance for thinned fingerprint than existing binary image data hiding techniques.

6. REFERENCES

[1] A. K. Jain and U. Uludag, "Hiding biometric data," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 11, pp. 1494–1498, 2003.

[2] N.K. Ratha, J.H. Connell, and R.M. Bolle, "Secure data hiding in wavelet compressed fingerprint images," in *Proc. ACM workshops on Multimedia*, 2000.

[3] M. Vatsa, R. Singh, and A. Noore, "Feature based RDWT watermarking for multimodal biometric system," *Image and Vision Computing*, vol. 27, no. 3, pp. 293–304, 2009.

[4] M. Wu and B. Liu, "Data hiding in binary image for authentication and annotation," *IEEE Transactions on Multimedia*, vol. 6, pp. 528–538, 2004.

[5] H. Yang and A. C. Kot, "Pattern-based data hiding for binary image authentication by connectivity-preserving," *IEEE Transactions on Multimedia*, vol. 9, pp. 475–486, 2007.

[6] H. Yang and A. C. Kot, "Orthogonal data embedding for binary images in morphological transform domain-a high-capacity approach," *IEEE Transactions on Multimedia*, vol. 10, pp. 339–351, 2008.

[7] C.-L. Tsai, H.-F. Chiang, K.-C. Fan, and C.-D. Chung, "Reversible data hiding and lossless reconstruction of binary images using pair-wise logical computation mechanism," *Pattern Recognition*, vol. 38, no. 11, pp. 1993–2006, 2005.

[8] Y.-A. Ho, Y.-K. Chan, H.-C. Wu, and Y.-P. Chu, "High-capacity reversible data hiding in binary images using pattern substitution," *Computer Standards and Interfaces*, vol. 31, no. 4, pp. 787–794, 2009.

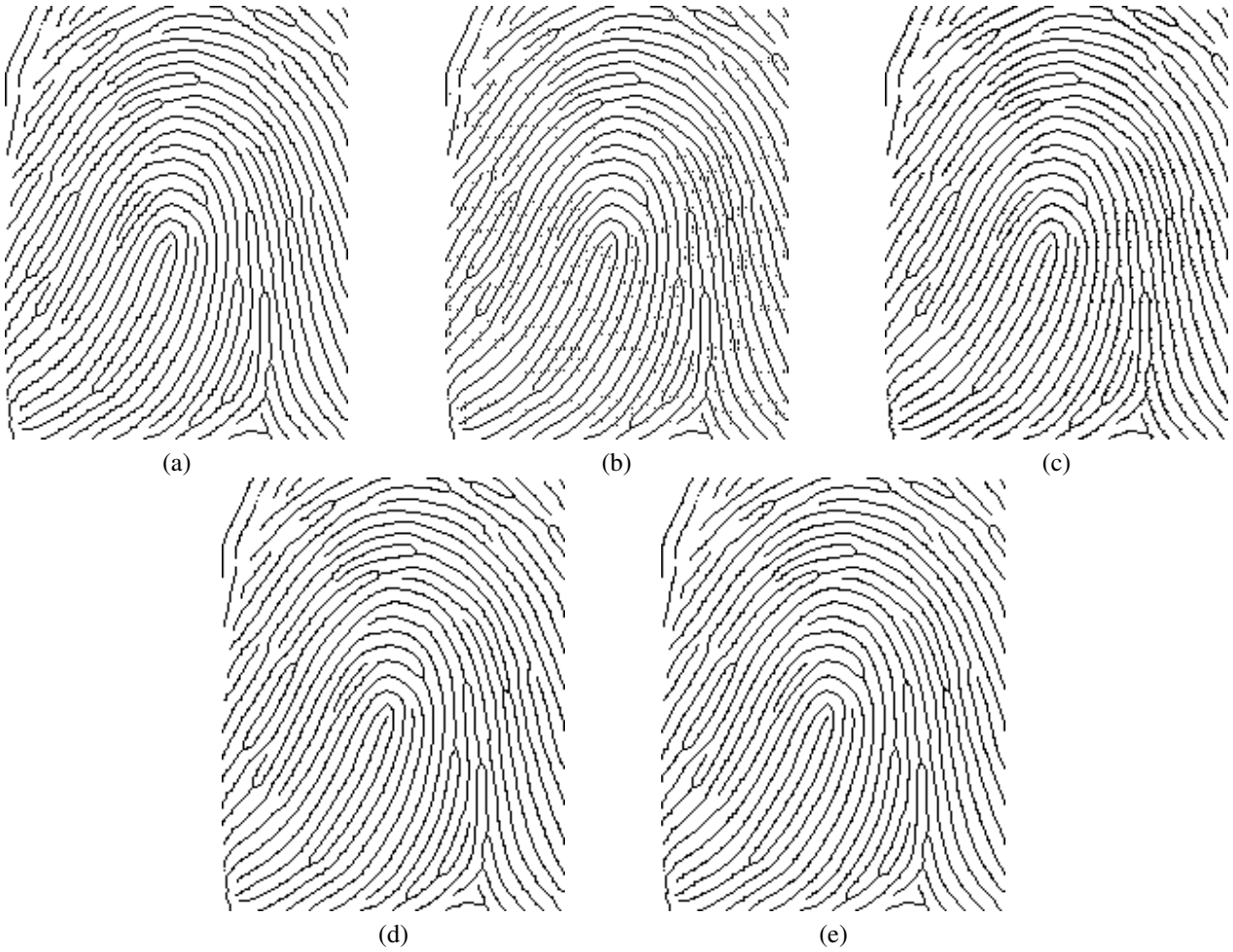


Fig. 8. Comparison of visual quality of the marked-fingerprint (right loop) by hiding 1000 bits, (a) using the proposed scheme, (b) using Ho *et al*.'s scheme [8], (c) using Xuan *et al*.'s scheme [9], (d) using Yang *et al.*'s [5] scheme, (e) using Yang *et al*.'s [6] scheme.

[9] G. Xuan, Y.Q. Shi, P. Chai, X. Tong, J. Teng, and J. Li, "Reversible binary image data hiding by run-length histogram modification," in *19th International Conference on Pattern Recognition, ICPR 2008*, 2008.

[10] G.T. Candela, P.J. Grother, C.I. Watson, R.A. Wilkinson, and C.L. Wilson, "A pattern level classification automation system for fingerprints," *NIST Technical Report NISTIR 5647*, 1995.

[11] L. Hong, Y. F. Wan, and A. Jain, "Fingerprint image enhancement: Algorithm and performance evaluation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 777–789, 1998.

[12] R. W. Zhou, C. Quek, and G. S. Ng, "A novel single-pass thinning algorithm and an effective set of performance criteria," *Pattern Recognition Letters*, vol. 16, no. 12, pp. 1267–1275, 1995.