Barman *et al. EURASIP Journal on Information Security* (2015) 2015:3
DOI 10.1186/s13635-015-0020-1

EURASIP Journal on Information Security
a SpringerOpen Journal

RESEARCH Open Access

# Fingerprint-based crypto-biometric system for network security

Subhas Barman[1], Debasis Samanta[2*] and Samiran Chattopadhyay[3]

**Abstract**

To ensure the secure transmission of data, cryptography is treated as the most effective solution. Cryptographic key is an important entity in this process. In general, randomly generated cryptographic key (of 256 bits) is difficult to remember. However, such a key needs to be stored in a protected place or transported through a shared communication line which, in fact, poses another threat to security. As an alternative to this, researchers advocate the generation of cryptographic key using the biometric traits of both sender and receiver during the sessions of communication, thus avoiding key storing and at the same time without compromising the strength in security. Nevertheless, the biometric-based cryptographic key generation has some difficulties: privacy of biometrics, sharing of biometric data between both communicating parties (i.e., sender and receiver), and generating revocable key from irrevocable biometric. This work addresses the above-mentioned concerns. We propose an approach to generate cryptographic key from cancelable fingerprint template of both communicating parties. Cancelable fingerprint templates of both sender and receiver are securely transmitted to each other using a key-based steganography. Both templates are combined with concatenation based feature level fusion technique and generate a combined template. Elements of combined template are shuffled using shuffle key and hash of the shuffled template generates a unique session key. In this approach, revocable key for symmetric cryptography is generated from irrevocable fingerprint and privacy of the fingerprints is protected by the cancelable transformation of fingerprint template. Our experimental results show that minimum, average, and maximum Hamming distances between genuine key and impostor's key are 80, 128, and 168 bits, respectively, with 256-bit cryptographic key. This fingerprint-based cryptographic key can be applied in symmetric cryptography where session based unique key is required.

**Keywords:** Symmetric cryptography; Cryptographic key generation; Biometric security; Crypto-biometric system; Network security

## 1 Introduction

Information security and a secure transmission of data become very important in information and communication technology. A third party can trap data or steal important data stored in a computer. To prevent this, it is advocated to encrypt the messages to provide information security. This type of protection is usually provided using cryptography. In cryptography, a key ($K_1$) is used to encrypt a message (called plaintext $P$) with encryption algorithm ($E$) into ciphertext ($C$). The ciphertext is converted into plaintext using a key ($K_2$) and decryption algorithm ($D$). There are two types of cryptography: symmetric cryptography and asymmetric cryptography. In symmetric cryptography (e.g., Data Encryption Standard (DES) [1], Advanced Encryption Standard (AES) [2]), same key (i.e., $K_1 = K_2 = K$) is used in encryption ($C = E_K(P)$) and decryption algorithm ($P = D_K(C)$). In asymmetric cryptography (e.g., Rivest-Shamir-Adleman (RSA) algorithm [1]), two different keys are used (i.e., $K_1 \neq K_2$), the public key is used to encrypt a message ($C = E_{K_1}(P)$), and private key is used to decrypt the ciphertext into plaintext ($P = D_{K_2}(C)$) [1].

Strength of cryptography in respect to security depends on the strength of keys used in encryption and decryption algorithms. A key is said to be strong if it is not easily guessed and not feasible to break within a real time. So, the issue that arises is the selection of cryptographic key. If the key is simple or very short in length, then for the

*Correspondence: dsamanta@iitkgp.ac.in
[2]School of Information Technology, Indian Institute of Technology Kharagpur, Kharagpur-721302, West Bengal, India
Full list of author information is available at the end of the article

attacker it is easy to guess the key. If the key is very long (128, 192, or 256 bits, for example, in AES algorithm [2]), then it is very difficult to memorize the key by a user. As a consequence, user should store it in a smart card or hardware token which can be misplaced or stolen out by an attacker. Moreover, the token or smart card is protected by password-based authentication mechanism to control the access of cryptographic key. Nevertheless, password can be forgotten or guessed by social engineering [3] and dictionary attack [4]. Both knowledge-based (e.g., password) and possession based (e.g., token) authentication systems are unable to assure non-repudiation property in traditional cryptography.

Of late, biometric is being integrated with cryptography (called crypto-biometric system) to alleviate the limitations of the above-mentioned systems [5,6]. Biometric is the unique measure of the identity of individuals with their behavioral and physiological traits like face, fingerprint, iris, retina, palm-print, speech, etc. [7]. Many researchers are trying to use biometric traits in the authentication component of cryptography to remove the requirement of password-based authentication. The integration of biometric with cryptography, deals with either cryptographic key release [8-10] or cryptographic key generation [6,11-17], is promising in many aspects. As biometric is directly linked with the owner, it removes the problem to memorize the cryptographic key and confirms the non-repudiation of users.

Crypto-biometric system, however, has some issues. Any biometric system needs to provide biometric template protection which confirms the privacy and security of biometric data [18]. The biometric data used in a biometric system should not leak any information about the biometric features. It is also required to provide revocability to the irrevocable biometric data. In password-based authentication systems or token-based authentication systems, passwords or tokens are easy to change while it is compromised. But, biometric traits are inherent and fixed forever, that is, the biometric data is irrevocable [7]. The owner of biometric traits is not able to revoke her biometric when it is compromised. As a result, the biometric data become useless forever [6]. To overcome this problem, it demands a cancelable transformation [18,19] of biometric template to provide revocability to the irrevocable biometric. Simultaneously, it would ensure the privacy of biometric data [5], so that the transformed template does not leak any information about the original template. Moreover, biometric data is required to be transmitted over non-secure communication channels for remote use. Therefore, there is a need to generate cryptographic key, which is revocable and non-invertible from the biometrics of two different users without compromising the privacy and security of the biometrics involved in key generation process.

This work aims to address the above-mentioned concerns and proposes a solution to develop a crypto-biometric system. Our proposed solution includes the following: 1) how to generate cancelable fingerprint template so that biometric features of neither communicators are never disclosed to anyone, 2) how to generate a unique cryptographic key for encryption (decryption) of messages using the cancelable fingerprint templates of both sender and receiver, and 3) how to generate revocable session key from irrevocable biometric traits prior to each session. In this paper, we propose an approach to generate, share, and update cryptographic key for symmetric cryptography from the fingerprints of sender and receiver at their sites for encryption and decryption, respectively. Initially, sender shares two secret keys namely stego key ($K_g$) and shuffle key ($K_{\text{shuf}}$) with receiver. Stego key is generated from a password (*pwd*) by sender and receiver using pseudo random number generator (PRNG) [1]. Shuffle key ($K_{\text{shuf}}$) is generated randomly, which is a binary stream of bits and stored in token. In this work, sender shares $K_{\text{shuf}}$ and *pwd* with receiver using public key cryptography. With our proposed approach, asymmetric cryptography is proposed to exchange an initial shuffle key $K_{\text{shuf}}$ and a password *pwd* between sender and receiver. For session keys, we propose biometric-based cryptographic key generation to establish a link of users biometric with cryptographic key. In our approach, biometrics of both communicating parties are integrated to generate cryptographic keys so that we can avoid the complex random number generation and alleviate the issue of storing the random cryptographic keys in the custody of sender and receiver. Moreover, revocable key generation in every session and protecting the privacy of biometric templates are the challenge which has been addressed in this work. Both sender and receiver exchange their cancelable fingerprint template with each other using key-based steganography. Both cancelable templates are then merged together using concatenation-based feature level fusion technique [20] to generate a combined template. Shuffle key is used to randomize the elements of the combined template. Finally, cryptographic key is generated from this shuffled template using a hash function. In our approach, fingerprint identity of sender is not disclosed to receiver and vice versa as cancelable template is exchanged between them to derive cryptographic key. Moreover, in our approach, cryptographic key or fingerprint template or both can be revoked easily if required. The revocability is provided to the cryptographic key with cancelable template and or with updated shuffle key.

The rest of the paper is organized as follows. A brief review of related research is given in Section 2. The proposed approach for cryptographic key generation from the fingerprints of sender and receiver is given in Section 3. The experimental results and security analysis

are discussed in Sections 4 and 5, respectively. Finally, the paper is concluded in Section 6.

## 2 Literature survey

Our work consists of mainly three sub-tasks: i) transformation of biometric template, ii) secure transmission of biometric data, and iii) crypto-biometric system. There exist few work in the literature related to each sub-task, which are discussed in this section.

### 2.1 Biometric template transformation

Biometric systems require a transformation of biometric template to ensure privacy, security, and revocability of biometric data. The technique which can meet this requirement is called cancelable or revocable biometric. This privacy enhancement problem is identified, and conceptual frameworks of biometric templates are presented in [21,22]. Ratha et al. [23] formally defined the problem of cancelable biometric. Uludag et al. [6] provides a comprehensive review on privacy and revocability of biometrics with some corrective measures. Recently, Ratha et al. [19] proposed three practical solutions to cancelable biometrics and generate cancelable fingerprint templates. These three template transformation approaches are Cartesian, polar, and functional transformations on feature domain. In Cartesian transformation approach, the minutiae space is divided into rectangular cells which are numbered with sequence. A user-specific transformation key (i.e., matrix) is used to shift the cells to a new location, and the minutiae points are relocated to the new cells. In polar transformation, coordinate space is divided into polar sectors that are numbered in sequence. The sector position is changed with the help of translation key, and it changes the minutiae location also. In functional transformation, Ratha et al. [19] model the translation using a vector-valued function $\overrightarrow{F}$ (x,y) which is an electric potential field parameterized by a random distribution of charges. The phase angle of the resulting vector decides the direction of translation and the magnitude $|\overrightarrow{F}|$ of this vector function parameterizes the extent of movement. In an alternate formulation, Ratha et al. [19] use the gradient of a mixture of Gaussian kernels to determine the direction of movement and the extent of movement is determined by the scaled value of the mixture. Some researchers proposed shuffling-based transformation to generate cancelable templates using a user-specified random key [24,25]. In these works, the iris code is divided into blocks and then the blocks are shuffled with a user-specified random shuffling key to generate cancelable iris template.

Jain et al. [18] reviewed the existing work of fingerprint template protection such as encryption, template transformation, and crypto-biometric systems. They analyzed the practical issues involved in applying these techniques for biometric template protection. They compared the existing solutions of template protection on the basis of template security and matching accuracy of biometrics in transformed domain.

### 2.2 Biometric data transmission

There are many work reported in the current literature where the biometric data is transmitted over communication channels for the purpose of remote authentication. Existing work [26-28] consider hiding of biometric data within another media called cover media using data hiding technique. Different types of data hiding techniques are used for secure transmission of biometric data using steganography. Minutiae points of fingerprint are hidden within face or synthetic fingerprint using watermarking technique and sent to other user via insecure communication channel [26]. Similarly, fingerprint is also used as cover media to hide other biometric data (i.e., face) in watermarking technology and used as carrier image for secure transmission of biometric data [27]. Note that in the data hiding concept, use of real biometric as cover media is risky as it reveals sender's biometric identity to receiver. Agrawal and Savvides [29] propose a biometric data hiding approach where a biometric (iris and fingerprint) data is encrypted with a key and the encrypted biometric data is encoded with error correcting code. The encoded biometric data is embedded bit by bit using the sign of discrete cosine transform (DCT) coefficients of a random cover image.

### 2.3 Crypto-biometric systems

Biometric-based cryptosystems are classified into two types, namely key release and key generation. In the first approach, a randomly created cryptographic key is protected from unauthorized access with users' biometric data. Fuzzy vault [8-10] and fuzzy commitment scheme [24,25,30] fall under this category. In fuzzy vault scheme, biometric data (e.g., minutiae points) is considered as an unordered set $s^E = x_1, x_2, \ldots x_r$ of $r$ elements. The secret key ($k_v$) of $k$ bits is transformed into a polynomial of degree $k$. All elements of $s^E$ are evaluated on the polynomial and the polynomial evaluation value $P(x_i)$ and $x_i$, that is, $(x_i, P(x_i))$ points are secured with some randomly generated chaff points $(a_j, b_j)^q_{j=1}$ which do not lie on the polynomial $P$ (i.e., $b_j \neq P(a_j)$ and $a_j \notin s^E, \forall j = 1, 2, \ldots, q$). The genuine $((x_i, P(x_i)))$ and chaff $(a_j, b_j)$ points constitute the fuzzy vault. The security of this vault depends on the computational difficulties of solving polynomial reconstruction problem. Now, the secret key is released only when the query biometric is close to the set $s^E$. Most of the existing fingerprint-based fuzzy vault use the $(x, y)$ coordinate values of minutiae points [9,10] whereas, Nandakumar et al. [8] propose a fingerprint-based fuzzy

vault where both $(x, y)$ coordinates and orientation $(\theta)$ of minutiae points are used. On the other side, in fuzzy commitment scheme, biometric data is represented in a binary vector $b^E$ and the vector is locked by a random secret key of less or equal bits of $b^E$ with XOR operation. In [24,25], iris code is combined with a random key using XOR operation and using the query iris code, the secret key is extracted from the combined iris code.

Few approaches have been proposed to generate cryptographic key from the biometric traits [14-16,31,32]. Monrose et al. [14] propose an approach to generate cryptographic key from user's voice while speaking a passphrase. Feng et al. [15] propose a cryptosystem, that is, BioPKI, where user's online signature is used to generate a private key. In [16,32], face biometric is used to extract a suitable length cryptographic key. Iris is also used for cryptographic key generation from iris texture [12,13,31]. Rathgeb et al. [31] analyze the iris feature vector and detected the most stable or reliable bits in the binary iris code to construct cryptographic key. Fingerprint, the most universal and acceptable biometric, is also used to derive a cryptographic key from cancelable fingerprint template [17,33,34]. Main problem of the approaches [17,33,34] is that it is not able to generate revocable key for session based communication.

In recent research, multimodal or multiple biometrics are used in crypto-biometric systems [11-13]. A Jagadeesan et al. proposed a method [12] where multimodal biometrics (fingerprint and iris) are used. They applied the feature level fusion of minutiae points and texture properties of iris to generate the multimodal biometric templates and the key is generated from this template. In another work [13], Jagadeesan et al. use the same biometrics (fingerprint and iris) but different method to generate the transformed template. The exponentiation operation is performed where iris texture values are used as base numbers and minutiae coordinates are used as exponent. Then, the next prime number is calculated for each exponentiation result and multimodal template is generated using multiplication of two resultants prime numbers to generate a key of 256 bits. This approach is, however, not free from key sharing problem of traditional symmetric cryptography.

Dutta et al. reported a method of fingerprint-based cryptography and network security [11]. In this method, they work with the fingerprints of sender and receiver. The fingerprint of receiver is transmitted to sender, and it is merged with sender's fingerprint to generate cryptographic key (of 128 bits) using standard hash function (MD5). In their approach, cryptographic key along with a random vector are watermarked into the genuine fingerprint and watermarked image is sent to the recipient. This method is not secure as genuine biometric is used as the cover image for data hiding. As the key and random sequence vector are transmitted over insecure channel with data hiding technique, it causes security threats to the message transmission if the fingerprint of the user is compromised to a third party, anyway. In this approach, fingerprint of receiver is sent to sender and fingerprint of sender is used as cover image and the watermarked image containing the master key and random vector is sent to receiver. Thus, fingerprint of sender is known to receiver and vice versa. A third party (man-in-middle) can generate the key using cryptographic hash function and with the knowledge of fingerprints of sender and receiver. Further, this approach is silent about the revocability issue.

## 3 Proposed methodology

In this section, we discuss our proposed approach in details. An overview of our approach is shown in Figure 1. In our approach, both sender and receiver extract minutiae points from their own fingerprints. The minutiae points are transformed into a cancelable form called cancelable template. The cancelable templates are exchanged between them using steganography. The stego key $(K_g)$ is used by both parties for secure steganographic use. The stego key is generated from a password (*pwd*) using PRNG. Both cancelable templates are combined together and shuffled using shuffle key $(K_{\text{shuf}})$ and finally the cryptographic key is generated from it following a hash function. In this scheme, initial shuffle key $K_{\text{shuf}}$ and password *pwd* both are selected by sender. Sender uses asymmetric cryptography to share the concatenated shuffle key and password, that is, $K_{\text{shuf}}||\text{pwd}$ to receiver. Sender uses the public key $K_{\text{pub}}$ of receiver to encrypt the $(K_{\text{shuf}}||\text{pwd})$ and sends $E_{K_{\text{pub}}}(K_{\text{shuf}}||\text{pwd})$ to receiver. Receiver can decrypt the shuffle key and password using his own private key $K_{\text{prv}}$, and they are used for key generation and template sharing, respectively. The above-mentioned steps in our approach are stated in details in the following.

### 3.1 Feature extraction from fingerprint image

We consider minutiae points (ridge ending and bifurcations) as the biometric features. The features are stored as $(x, y, \theta)$ form, where $(x, y)$ denotes coordinate value and $\theta$ is the orientation of a minutiae point. In fingerprint authentication systems, inclusion of more features increase matching scores and hence the angle information is preferred. The objective of our work is to use fingerprint data as the source of randomness rather than the authentication of a user. It is observed that $x$ and $y$ coordinate values are enough to provide randomness in data. Therefore, we have considered only $(x, y)$ coordinate values as the minutiae points in our work. We extract minutiae-based features from the fingerprint images of both sender and receiver. For the reference in our subsequent discussion, we denote them as follows.

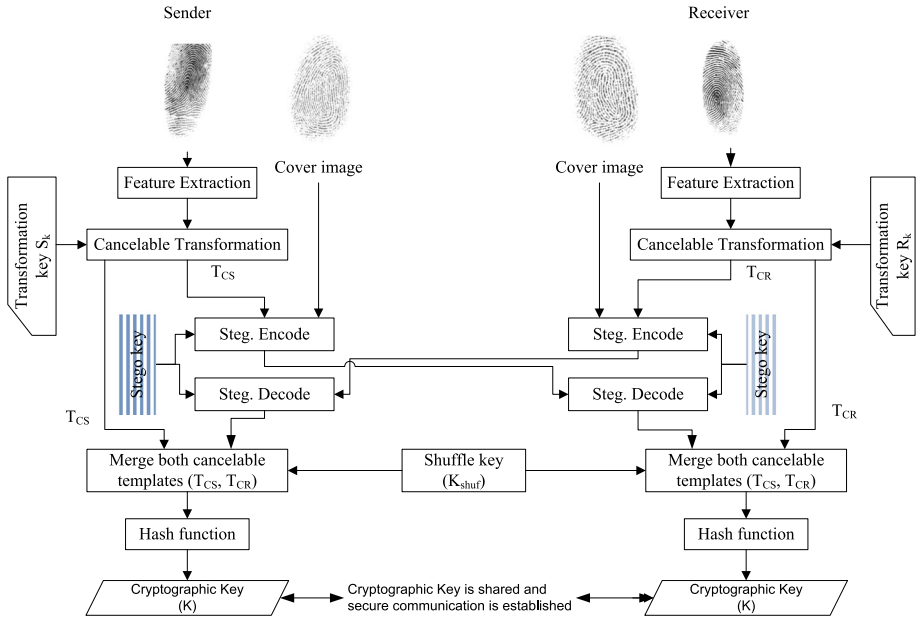



**Figure 1** An overview of our proposed crypto-biometric system.

- $F_S$ = Set of minutiae points extracted from sender's fingerprint.
  $= \left[m_1^s, m_2^s, \ldots, m_{N_s}^s\right]$; where $m_i^s = (x_i, y_i)$, $m_i^s$ is the $i$th minutiae points of sender's fingerprint, $i = 1$ to $N_s$ and $N_s$ is the size of $F_S$
- $F_R$ = Set of minutiae points extracted from receiver's fingerprint.
  $= \left[m_1^r, m_2^r, \ldots, m_{N_r}^r\right]$; where $m_i^r = (x_i, y_i)$, $m_i^r$ is the $i$th minutiae points of receiver's fingerprint, $i = 1$ to $N_r$ and $N_r$ is the size of $F_R$.

Note that, for all fingerprint images, it is a general observation that number of minutiae points for a person lies within 50. However, in case, if there are more than 50 minutiae points (i.e., $N_s, N_r \geq 50$), then first 50 minutiae points according to their quality value would be selected and the rest be discarded.

### 3.2 Cancelable template generation

The fingerprint templates of both sender and receiver are transformed into a non-invertible forms, called cancelable templates, to provide revocability as well as privacy to the fingerprint data of both users before transmitting them to their counter partners. The position of minutiae features are changed using Cartesian transformation with the help of a user-specified transformation key (i.e., $S_k$ for sender and $R_k$ for receiver). The overall process of transformation is discussed below.

1. The coordinate system is divided into $N$ cells of same size $h \times w$, where $h$ is the height and $w$ is the width of the cells. The total number of cells $N$ can be calculated with equation given below
$$N = \frac{(H \times W)}{(h \times w)} \tag{1}$$
where $H$ and $W$ are the height and width of the fingerprint image.
2. Each cell is denoted by $C_{i,j}$ where $i = 1$ to $n$ ($n$ cells in each row) and $j = 1$ to $m$ ($m$ cells in each column). The cells can be represented in a one-dimensional vector, and cell $C_{i,j}$ can be represented by $c_t$ and the value of $t$ can be computed in the following way.
$$t = \{(j-1) \times n\} + i \tag{2}$$
where for any value of $i$ ($1 \leq i \leq n$) and $j$ ($1 \leq j \leq m$), there will be a unique $t$ such as $1 \leq t \leq N_{\text{cell}}$. Each cell contains either no minutiae or a set of minutiae points. It depends on the cell size, distribution of minutiae points in fingerprint image. For sender's fingerprint, if the $t$th cell $c_t$ contains $n_{\text{mt}}$ number of minutiae then the $i$th minutiae point of $t$th cell of senders fingerprint can be represented as $m_i^{\text{st}}$ and the value of $i$ is defined as $1 \leq i \leq n_{\text{mt}}$. For an illustration, we divide the image space into $n$ vertical cells and $m$ horizontal cells, and the cells are shown in Figure 2 where the two-dimensional cells are shown in left and the sequence number is shown in right side table.
3. Generate a user-specific (0,1) matrix ($M$) of size $N \times N$ (where $N = n \times m$) to map the cells with their new positions as per the following equation.

| | | | | | | |
|---|---|---|---|---|---|---|
| m | $C_{1m}$ | $C_{2m}$ | ... | $C_{im}$ | ... | $C_{nm}$ |
| : | : | : | ... | : | ... | : |
| j | $C_{1j}$ | $C_{2j}$ | ... | $C_{ij}$ | ... | $C_{nj}$ |
| : | : | : | ... | : | ... | : |
| 2 | $C_{12}$ | $C_{22}$ | ... | $C_{i2}$ | ... | $C_{n2}$ |
| 1 | $C_{11}$ | $C_{21}$ | ... | $C_{i1}$ | ... | $C_{n1}$ |
| | 1 | 2 | ... | i | ... | n |

| | | | | | |
|---|---|---|---|---|---|
| 31 | 32 | 33 | 34 | 35 | 36 |
| 25 | 26 | 27 | 28 | 29 | 30 |
| 19 | 20 | 21 | 22 | 23 | 24 |
| 13 | 14 | 15 | 16 | 17 | 18 |
| 7 | 8 | 9 | 10 | 11 | 12 |
| 1 | 2 | 3 | 4 | 5 | 6 |

i = j = 4; n = m = 6

**Figure 2** $(x, y)$ coordinate space is divided into cells, and the equivalent cell numbers are shown in right.

$$C'_{i'j'} = C_{i,j}M \tag{3}$$

where $C_{i,j}$ is the original cell which is replaced by cell $C'_{i',j'}$. Here, one cell can replace zero or multiple cells. The number of minutiae points (say $n_{\text{mt}}$) in the new cell can be the same or different.

4. Modify the $(x, y)$ coordinates according to their new cell locations, and cancelable template $T_C$ is generated as follows.

$$T_C = F\left(C_{i,j}, C'_{i',j'}\right) \tag{4}$$

The $(x, y)$ coordinate values of all minutiae points belong to $C'_{i',j'}$ will be placed to the cell $C_{i,j}$ according to the mapping function (F).
For the replacement of cell $C_{i,j}$ by $C'_{i',j'}$ (i.e., $C_{i,j} \leftarrow C'_{i',j'}$), the $(x, y)$ coordinate value of minutiae point of cell $C'_{i',j'}$ can be computed for new location as follows.

(a) If $i' = i$ then $x_i^d = x_i$
(b) If $i' > i$ then $x_i^d = x_i + (i' - i) * w$
(c) If $i' < i$ then $x_i^d = x_i - (i - i') * w$

where $x_i$ is the $x$ coordinate value of a minutiae points and $x_i^d$ is the displaced $x$ coordinate value of the same minutiae points and $w$ is the width of a cell of size $h \times w$. Similarly, depending on the value of $j'$, the $y_j$ will be changed to $y_j^d$.

For example, a simplified coordinate value of minutiae points are divided by four cells and numbered as $C_{1,1} = 1, C_{1,2} = 2, C_{2,1} = 3, C_{2,2} = 4$. The Cartesian transformation is given in the following equation.

$$\begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 3 & 2 & 4 & 3 \end{bmatrix} \tag{5}$$

where, cells 1 and 4 are replaced by cell 3 while cell 3 is replaced by cell 4. The cell-wise coordinate values of minutiae points before and after transformation are shown in Figure 3. A pictorial representation of the cancelability transform, using real minutiae sets is shown in Figure 4. Here, we have divided the height $H$ and width $W$ of the fingerprint image by 8 to make $8 \times 8 = 64$ cells of size $\frac{H}{8} \times \frac{W}{8}$. Also, it may be noted that the height and width of the cells in last row and last column may vary.

After transformation of fingerprint template using this transformation, the transformed template ($T_C$), also known as cancelable template, contains modified minutiae points denoted by $m_i^{s'}$, which represents $i$th modified minutiae point or $i$th elements of the cancelable template. In the cancelable template, we consider that it contains 50 modified minutiae points. If it exceeds 50, then we consider only first 50 elements and if it contains less than 50 elements, then we augment sufficient numbers of zero elements at the end to make it of intended size. In the subsequent discussion, the cancelable templates of sender and receiver are denoted by $T_{\text{CS}}$ and $T_{\text{CR}}$, respectively, where $T_{\text{CS}} = \left\{m_1^{s'}, m_2^{s'}, \ldots, m_{N_{\text{TCS}}}^{s'} | N_{\text{TCS}} = |T_{\text{CS}}|\right\}$ and $T_{\text{CR}} = \left\{m_1^{r'}, m_2^{r'}, \ldots, m_{N_{\text{TCR}}}^{r'} | N_{\text{TCR}} = |T_{\text{CR}}|\right\}$.

### 3.3 Steganographic encoding

In this work, cancelable template of one party (say sender) needs to be sent through a shared communication channel to other party (say receiver) and vice versa. Sender (and receiver) uses steganography-based data hiding technique to hide the cancelable template data into a cover image ($I$) (of size $M_I$ pixels, say). The cancelable template is converted into binary stream $(s_1, s_2, s_3, \ldots s_L)$, where $L$ is the number of bits in cancelable template after the

| Cell No | X | Y |
|---|---|---|
| Cell- 1 | 2 | 2 |
| | 3 | 5 |
| | 4 | 1 |
| Cell- 2 | 12 | 3 |
| | 17 | 7 |
| Cell- 3 | 15 | 12 |
| Cell- 4 | 5 | 18 |
| | 8 | 13 |

Original Template

| Cell No | X | Y |
|---|---|---|
| Cell- 1 | 5 | 2 |
| Cell- 2 | 12 | 3 |
| | 17 | 7 |
| Cell- 3 | 15 | 18 |
| | 18 | 13 |
| Cell- 4 | 5 | 12 |

Cancelable Template

**Figure 3** Original fingerprint template and cancelable template.

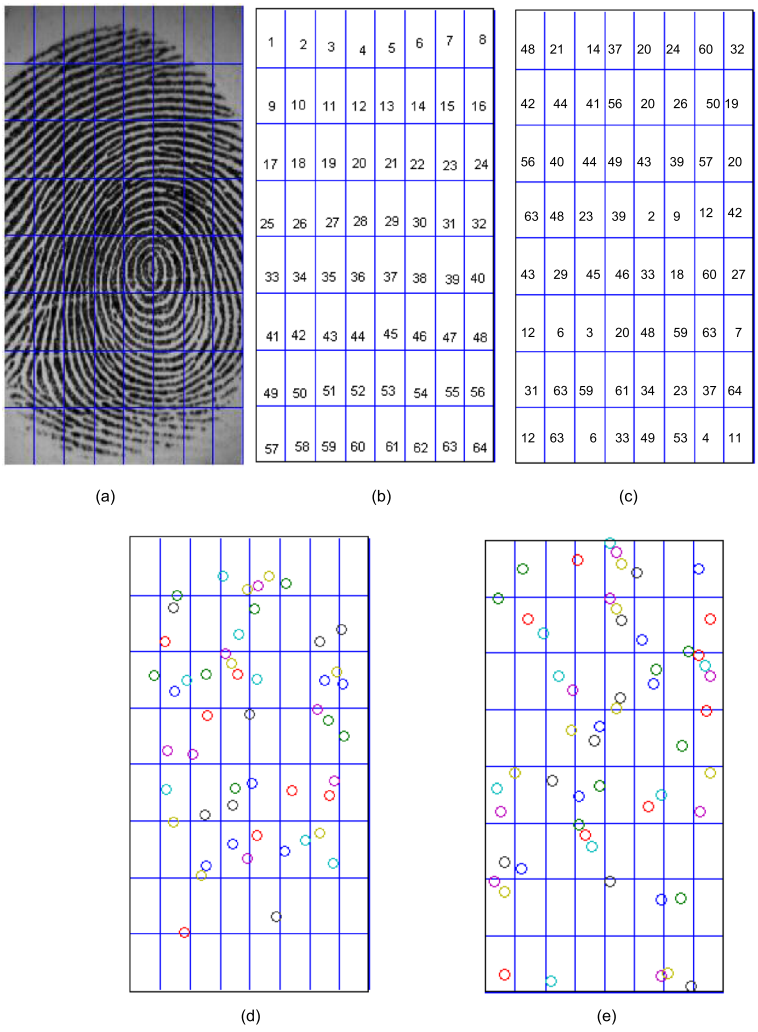



**Figure 4** A pictorial representation of Cartesian transformation. **(a)** Original fingerprint image, **(b)** Cells , **(c)** Cells which replaces the original cells, **(d)** Original minutiae points, **(e)** Minutiae points in cancelable template.

conversion, which is to be hidden into cover image using LSB steganography [35]. A secret key, called stego key $K_g$, is generated from a password (*pwd*) using pseudo random number generator where the password is used as the seed value. The stego key ($K_g = [k_1, k_2, \ldots k_{N_{\text{kg}}}]$; where $L \leq N_{\text{kg}} \leq M_I$] is used to select the pseudo random path of pixel locations in cover image ($I$) to hide the cancelable template bit by bit in cover image. The stego image ($I_{\text{stego}}$) is sent to the recipient from sender and vice versa.

### 3.4 Steganographic decoding

In this phase, hidden data are extracted from the stego image ($I_{\text{stego}}$) using the decoding function. The stego key ($K_g$) is used to locate the pixels where data embedding take place. The extracted binary stream is then used to reconstruct the cancelable template.

### 3.5 Merging cancelable templates $T_{\text{CS}}$ and $T_{\text{CR}}$

After receiving the cancelable template of the counter partner, receiving party merges his own cancelable template with the received cancelable template. Say, sender has its own cancelable template $T_{\text{CS}}$ and received the cancelable template $T_{\text{CR}}$ from receiver. Both $T_{\text{CS}}$ and $T_{\text{CR}}$ consist of modified minutiae points ($m_i^{s'}$, $m_i^{r'}$) of sender and receiver, respectively. These two cancelable templates are fused using the feature level fusion [20,36] of modified minutiae features.

Feature level fusion is achieved with concatenation of two feature sets [20] $T_{\text{CS}} = \left[m_1^{s'}, m_2^{s'}, \ldots, m_{N_{\text{TCS}}}^{s'}\right]$ and $T_{\text{CR}} = \left[m_1^{r'}, m_2^{r'}, \ldots, m_{N_{\text{TCR}}}^{r'}\right]$. ($|T_{\text{CS}}| = N_{\text{TCS}}$ and $|T_{\text{CR}}| = N_{\text{TCR}}$) to generate a combined template $T_f (|T_f| = N_{\text{TCS}} + N_{\text{TCR}})$. For this purpose, all $x$ coordinate values of $m_i^{s'}$ from $T_{CS}$ *are stored in vector X and the x coordinate values*

*of $m_I^{r'}$ from $T_{CR}$ are augmented to the same vector*. Similarly, the $y$ coordinate values of $T_{\text{CS}}$ and $T_{\text{CR}}$ are combined and stored in another vector $Y$. These two vectors $(X, Y)$ generate a new $(x, y)$ coordinates of $T_f$. The size of $T_f$ is the total size of $T_{\text{CS}}$ and $T_{\text{CR}}$. That is

$$T_f = T_{\text{CS}}||T_{\text{CR}};\ |T_f| = |T_{\text{CS}}| + |T_{\text{CR}}|. \tag{6}$$

where $||$ denotes the augmentation operation, $|T_f|$ is the size of $T_f$. In the combined template, redundancy may exist. The redundancy, if it exists, is removed, and only unique modified minutiae points are selected from $T_f$. As an example, sample feature level fusion of two feature sets is shown in Figure 5c.

The elements of vectors $X$ and $Y$ are shuffled separately using the shuffle key ($K_{\text{shuf}}$). The initial shuffle key (of 200 bits and 100 bits are required for each template) can be generated randomly. Our proposed shuffling method is illustrated in Figure 6. In this shuffling method, the vector elements where corresponding key bits are 1 are sorted starting at the beginning, and the remaining elements where the key bits are 0 are placed starting from the end. In this way, all elements of vector $X$ and $Y$ are shuffled and the shuffled $X$, $Y$ vectors (denoted as $X^S$ and $Y^S$) result a modified $F$. For example, a sample shuffled $F$ is shown in Figure 5d.

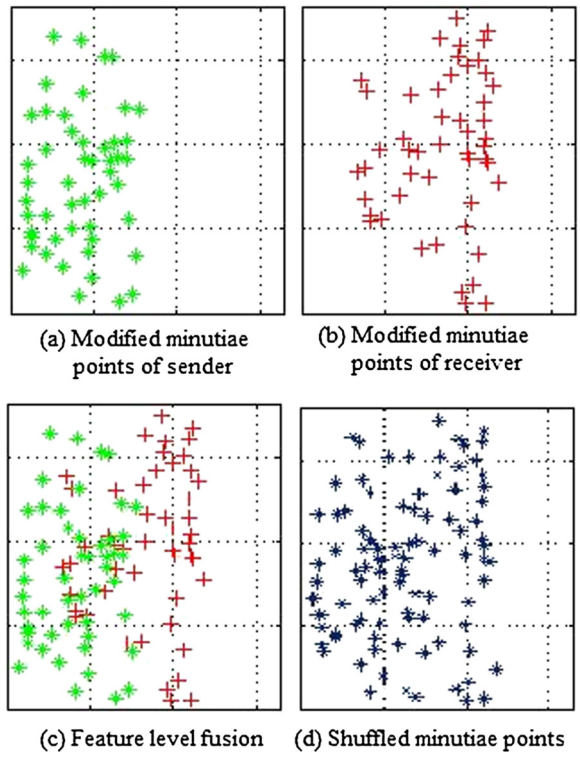



**Figure 5** A snapshot of fused features and shuffled features. **(a)** Modified minutiae points of sender. **(b)** Modified minutiae points of receiver. **(c)** Feature level fusion. **(d)** Shuffled minutiae points.

Each corresponding element of shuffled vectors ($X^S$ and $Y^S$) is merged using XOR operation. For this purpose, $x_i$ and $y_i$ ($x_i \in X^S$ and $y_i \in Y^S$) are converted into binary numbers and bitwise XOR operation is followed for all elements of $X^S$ and $Y^S$. The results of bitwise XOR operation are stored in a vector $F_{\text{code}}$.

$$F_{\text{code}} = \int F_{\text{code}_i} = \int \text{bitwiseXOR}(x_i, y_i) \tag{7}$$

Finally, the cryptographic key is generated from this $F_{\text{code}}$ using a hash function which is as follows. The $F_{\text{code}}$ is divided into blocks (i.e., $F_{\text{code}} = B_1||B_2||\ldots||B_{n_b}$; say total $n_b$ blocks) of size 256 bits each. A vector $(K)$ of size 256 bits of all zeros is generated as the initial hash value. Now, block $B_1$ is XORed with initial $K$ and the output is stored in $K_2$. The $K_2$ is XORed with the next block $B_2$ and the result is stored in $K_3$ and so on. Finally, the hash value $K_{n_{b+1}}$ is the cryptographic key $(K)$. That is

$$K_{i+1} = (K_i \oplus B_i);\ \text{where } i = 1 \text{ to } n_b \text{ and } |K_i| = |B_i| = 256 \text{ bits} \tag{8}$$

This way, sender and receiver both derive the same secret key which establishes a secure communication between the sender and receiver for a session. For a new session, a new session key can be generated from the same fingerprints using an updated shuffle key, which is discussed in the next sub-section.

### 3.6 Shuffle key update

For better security measure, we propose to change the cryptographic key in each session. In other words, if there is a chance to compromise cryptographic key, then it is desirable that the key must be canceled and a new key be used for the next session. Further, we may note that if both cancelable templates become known to a third party then with the knowledge of key generation algorithm, key can be derived by the third party. But, in our approach, cryptographic key generation depends on another factor namely shuffle key ($K_{\text{shuf}}$). The revocability of the cryptographic key is achieved with not only the cancelable fingerprint template but also with shuffled key. Our protocol is also able to update the shuffle key time to time using the fingerprint data of both users. To realize this, we propose to update the shuffle key from one session to another. Session-wise shuffle key update procedure is shown in Figure 7. Initiation to update shuffle key can be taken by sender or receiver.

The steps followed in our shuffle key update process when it is initiated by the sender are given below.

1. Both sender and receiver share their cancelable fingerprint data (Sections 3.3 and 3.4) and generate $F_{\text{code}}$ following the method discussed in Section 3.5.

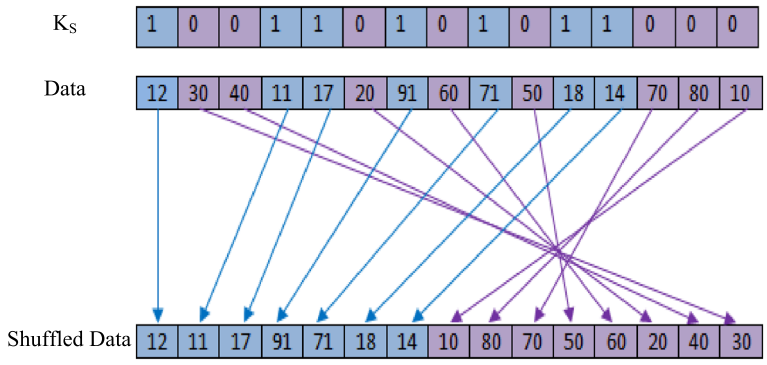



**Figure 6** Shuffling method.

2. Shuffle key ($K_{\text{shuf}}$) is XORed with the first $|K_{\text{shuf}}|$ bits of $F_{\text{code}}$ to obtain a new shuffle key ($K_{\text{SS}}$), that is, $K_{\text{SS}} = K_{\text{shuf}} \oplus F_{\text{code}}$.
3. Sender computes the hash value of the new shuffle key ($h(K_{\text{SS}})$) and sends it along with update request to receiver. We have used XOR-based hash function ($h$), as discussed in Section 3.5.
4. Similarly, receiver also generates a new shuffle key ($K_{\text{SR}} = K_{\text{shuf}} \oplus F_{\text{code}}$) and computes the hash of the new shuffle key ($h(K_{\text{SR}})$) using the same one way hash function ($h$) which is used in sender side.
5. Receiver compares the computed hash ($h(K_{\text{SR}})$) with received hash ($h(K_{\text{SS}})$), if both are same, then receiver replaces old shuffle key ($K_{\text{shuf}}$) with new one ($K_{\text{SR}}$) and sends success message to the sender.
6. On the basis of receiver's report, sender also replaces the old shuffle key ($K_{\text{shuf}}$) with new shuffle key ($K_{\text{SS}}$).

In this way, both sender and receiver are able to update their shuffle key. In every session, a unique shuffle key is generated from the fingerprint data of sender and receiver. After the session is over, old shuffle key is destroyed

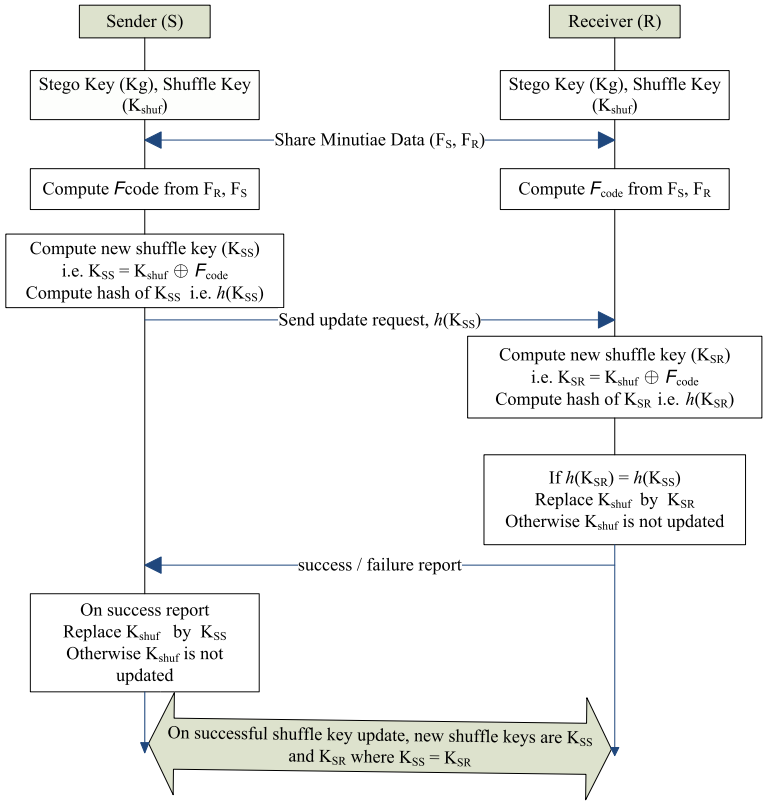



**Figure 7** Shuffle key update.

and modified one is used for next session of communication. In this way, our approach can provide the diversity of cryptographic key of fingerprint-based symmetric cryptography.

## 4 Experiment and experimental results

There are mainly two objectives in our experiment. First, we investigate the impact of data encoding into cover images (i.e., synthetic fingerprints from DB4 of each database) and accuracy of data decoding from stego images. In the next part of our experiment, we measure the randomness of cryptographic key generated from fingerprints of genuine users with respect to the key generated from impostor's fingerprints. In this regard, the Hamming distances between the genuine and impostor's keys are measured and corresponding histograms are plotted. Here, we consider all possible cases of attacks when different entities are compromised and the Hamming distances are computed for each case.

### 4.1 Database

We have tested our work using the fingerprint images from publicly available fingerprint databases, FVC2000 [37], FVC2002 [38], and FVC2004 [39]. Each FVC database has four subsets labeled as DB1 to DB4. Each subset has a development set (B) (from fingers 101 to 110) with ten people and a test set (A) with 100 peoples (fingers from 1 to 100). There are eight samples for each person. The details of fingerprints used in our experiment are shown in Table 1. In these databases, DB1, DB2, and DB3 are real fingerprint databases and DB4 is synthetic fingerprint [40] database. In set B of each databases, total number of fingerprint images are $(10 \times 8 \times 4) = 320$; and in set A of each database, there are $100 \times 8 \times 4 = 3,200$ fingerprint instances.

### 4.2 Experimental setup

A unique pair of fingerprints is taken (as fingerprint of sender and receiver) from a subset (i.e., DB1 or DB2 or DB3 or DB4) of a specific FVC to generate a genuine key. All remaining pairs of fingerprints in the same subset of the same FVC database are taken to generate impostor keys with respect to that genuine key. This way, all pairs of genuine fingerprints and corresponding pairs of impostor fingerprints are chosen. Initially, all unique combination of two person's fingerprints from 110 person's fingerprints of each subset of the fingerprint database is computed. Thus, the total number of pairs for genuine users is $\frac{110}{2} = 55$ for each subset (i.e., DB1, DB2, DB3, and DB4 for both sets A and B). For each FVC database (considering all four subsets of set (A+B), the total number of genuine keys is $4 \times 55 = 220$. Similarly, for every genuine key, remaining 54 keys are impostor keys. In our experiment, the duplicate pairs of genuine and impostor fingerprints are avoided carefully. As a result, we get $220 \times 3 = 660$ impostor's keys per FVC database and a total number of $3 \times 4 \times \frac{55 \times 54}{2} = 17,820$ impostor's keys.

Now, the minutiae points from fingerprint images are extracted using NBIS software (MINDTCT) [41]. The MINDTCT tool takes fingerprint image as input and returns minutiae points set in the format of $(x, y, \theta, q)$ where $q$ is the quality of that minuitae point. Average number of minutiae points is found as 50. We consider only $(x, y)$ coordinates of first 50 minutiae points according to the quality of minutiae reported by MINDTCT in our experiment. The minutiae points are transformed into cancelable template with a user-specified transformation key (Section 3.2). We have divided the fingerprint images into 64 cells. The cell size is computed by dividing the height ($H$) and width ($W$) with 8 (i.e., cell size $= h \times w$ where $h = \frac{H}{8}$ and $w = \frac{W}{8}$). The cancelable template converted into binary stream (Section 3.5). Most of the sizes of fingerprint images are within the range of 256 to 511. The maximum values of $x$ and $y$ coordinate points thus can be represented with 9 bits binary numbers. However, as exceptions, the $y$ coordinate value of the fingerprint images in some FVC2002 database (it is A and B sets in DB2), and the $x$ coordinate values of fingerprint images in FVC2004 database (e.g. A and B sets in DB1) exceed the range of 511. As the $x$ and $y$ coordinate values of minutiae points in almost all fingerprint images are less than 512, we represent them by 9 bits (maximum decimal value with 9 bits is $2^9 - 1 = 511$). Of course, the coordinates exceeding 511 are with a less approximation, which do not affect the results adversely. The values of $x$ and $y$ coordinate of modified minutiae points (i.e., the elements of cancelable

**Table 1 Fingerprints used in our experiments**

| | FVC2000 (sensors, image size) | FVC2002 (sensors, image size) | FVC2004 (sensors, image size) |
|---|---|---|---|
| DB1 | Optical (KeyTronic), 300 × 300 | Optical (Identix) (CrossMatch V300), 388 × 374 | Optical sensor, 640 × 480 |
| DB2 | Capacitive (ST Microelectronics), 256 × 364 | Optical (Biometrika), 296 × 560 | Optical Sensor (Digital Persona U.are.U 4000), 328 × 364 |
| DB3 | Optical (Identicator Technology), 448 × 478 | Capacitive (Precise Biometrics), 300 × 300 | Thermal sweeping sensor (Atmel FingerChip), 300 × 480 |
| DB4 | SFinGe v2.0, 240 × 320 | SFinGe v2.51, 288 × 384 | SFinGe v3.0, 288 × 384 |

templates) also lie within this range. Therefore, each element of the cancelable template is converted into 18 bits binary number (9 bits for $x$ coordinate and 9 bits for $y$ coordinate) and binary conversion of all elements of cancelable template produces a binary bit stream of $18 \times 50 = 900$ bits.

We propose to generate the transformation keys ($S_k$, $R_k$) randomly. In our experiment, these keys are generated using pseudo random number generator (PRNG) available in MATLAB. We have used a unique transformation key to generate a cancelable template from fingerprint of a specific person. In our approach, stego key ($K_g$) is also generated randomly (using PRNG in MATLAB) from a password (*pwd*) and a unique stego key is assigned to each cover image of the synthetic fingerprints (i.e., the fingerprints in DB4). A unique shuffle key (pseudo random number) is used for a unique pair of fingerprint images.

We use synthetic fingerprint [40] from DB4 database (of FVC2000, FVC2002, FVC2004) as the cover image to hide cancelable fingerprint template of genuine users using LSB steganography. The cover image is picked up at random by both sender and receiver. A stego key ($K_g$) is used to locate the pixels of cover image where the cancelable template ($C_{\mathrm{TS}}$, $C_{\mathrm{TR}}$) bits are hidden during steganographic encoding (Section 3.3). Similarly, the same stego key is used to decode the cancelable template bits from stego image during steganographic decoding (Section 3.4).

Note that in our experiment, both sender and receiver exchange their own cancelable template between them using the same stego key ($K_g$) but different cover images (say $I_S$, $I_R$).

### 4.3 Experimental results

The results of our experiment are stated below with respect to different scenarios.

### 4.4 Case 1: Impact of steganography

In our experiment, effect of data encoding over cover image is investigated and we follow the evaluation method as given in [26,27]. The stego key $K_g$ is fixed for each unique pair of users and it differs when the pair of sender and receiver is changed.

Few observations are summarized in Table 2. The first column is the average pixel value for cover images. The second column is the average pixel value for the stego images. The third column is the pixel change with respect to cover image. The last column represents the absolute pixel change of the total encoded pixels. It is also observed that 100% of the encoded message is extracted using the same stego key from stego image and nearly 0.47% pixels of the cover image is changed due to data hiding.

**Table 2 Effect of steganography**

| Cover pixel average | Stego pixel average | Overall pixel change | Absolute pixel change |
|---|---|---|---|
| 173.26 | 173.48 | 0.47% | 50.45% |

### 4.5 Case 2: Both fingerprints and shuffle keys are unknown

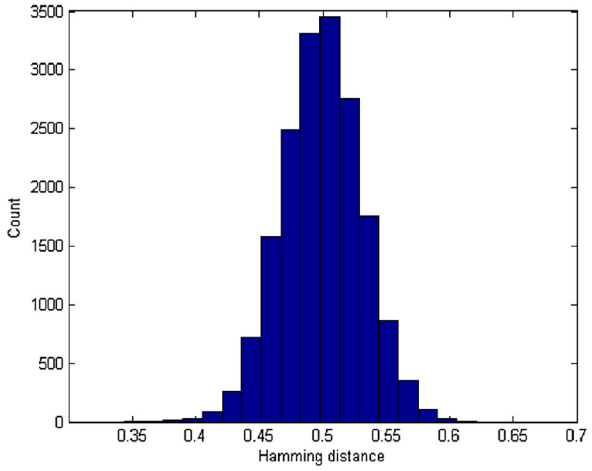
For this purpose, we consider that the attacker has no knowledge about either the genuine fingerprints or shuffle key ($K_{\mathrm{shuf}}$) to compromise the cryptographic key. In this case, unique $S_k$ and $R_k$ are used to transform each fingerprint and unique $K_{\mathrm{shuf}}$ is used for generation of each key. In this condition, we compute the Hamming distance between genuine and impostors' keys and the Hamming distances are plotted using histogram. The histogram in Figure 8a shows the distribution of Hamming distances of 17,820 comparisons between genuine and impostors' keys. It is observed from the histogram that mean Hamming distance is 49.95% which means that the average hamming distance between the genuine and impostors' key is 128 bits. In this case, the Hamming distances are spreaded between the range of 34.38% to 62.11% with a standard deviation of 0.032. According to the quantity of impostor's key, it is observed that 40% to 60% bits of the genuine keys are different from 99.89% impostor's keys. There is a small number (0.04%) of impostor's key whose unmatched bits are below 40%.

### 4.6 Case 3: Shuffle key is known

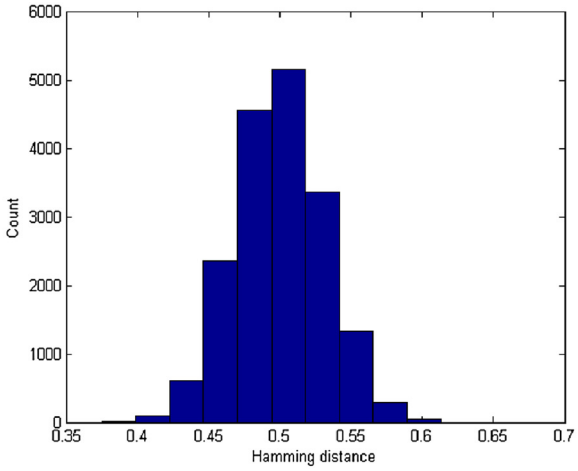
Now, we consider that the shuffle key ($K_{\mathrm{shuf}}$) is compromised by the attacker and an attacker tries to generate the same cryptographic key using this shuffle key from the fingerprints other than genuine fingerprints. In this case, same shuffle key ($K_{\mathrm{shuf}}$) is used to generate genuine and impostor keys. The transformation keys ($S_k$ and $R_k$) are distinct for each fingerprint. In our experiment, Hamming distances are computed to measure the similarity or dissimilarity of the genuine keys and impostor's keys. The observation is shown using histogram in Figure 8b. In this case, the Hamming distances are distributed from 37.50% (minimum) to 61.33% (maximum) with a mean of 50% and standard division of 0.0309. Maximum impostors' keys (i.e., 99.85% impostors' keys) differ from genuine key with the range of 40% to 60% Hamming distances. Even when the similarity of the impostor's key is maximum (i.e., 62.5% bits are similar), the attacker needs to guess 96 bits (i.e., $2^{96}$ trials in brute force attacks) to crack the genuine key.

### 4.7 Case 4: Only one genuine fingerprint/cancelable template is known

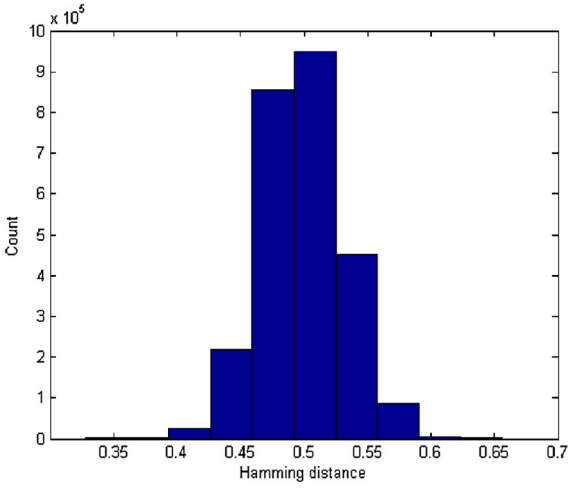
In the proposed approach, we also consider that one of the genuine fingerprints (either the fingerprint of sender or receiver) along with transformation key is compromised by the impostor and cryptographic key is generated using one genuine cancelable template and one impostor's



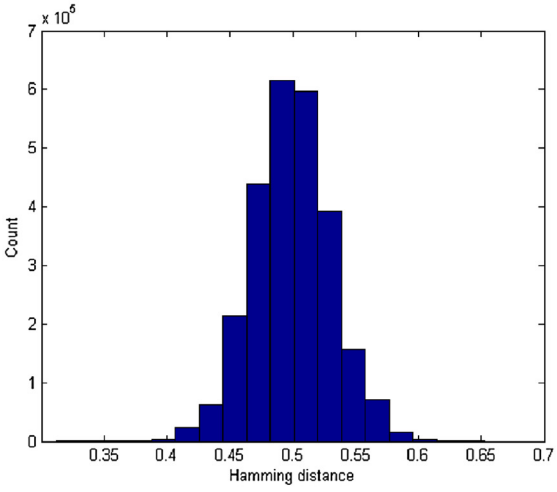

(a) $K_{shuf}$ and fingerprints are unknown



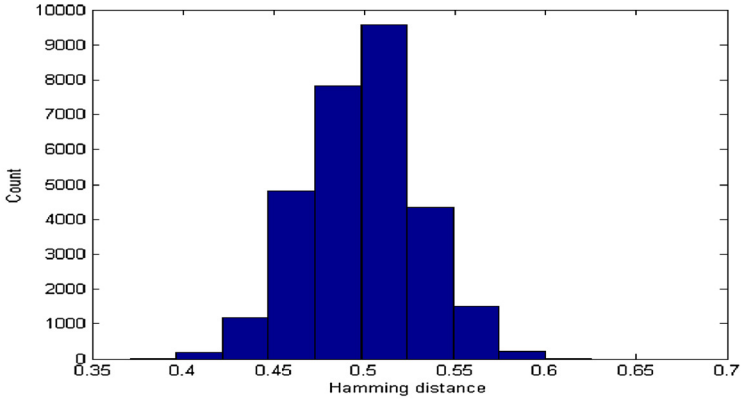

(b) Only $K_{shuf}$ is known




(c) One fingerprint is known but $K_{shuf}$ is unknown




(d) One fingerprint and $K_{shuf}$ are unknown




(e) Known fingerprints but unknown $K_{shuf}$

**Figure 8** Hamming distances between genuine and impostor's keys. **(a)** $K_{\text{shuf}}$ and fingerprints are unknown. **(b)** Only $K_{\text{shuf}}$ is known. **(c)** One fingerprint is known but $K_{\text{shuf}}$ is unknown. **(d)** One fingerprint and $K_{\text{shuf}}$ are unknown. **(e)** Known fingerprints but unknown $K_{\text{shuf}}$.

cancelable fingerprint template. In this case, shuffle key $K_{\text{shuf}}$ is unknown to the attacker and a different $K_{\text{shuf}}$ is used to generate impostor keys. Whereas, we have used one fixed transformation key ($S_k$) to generate genuine and impostor templates. The dissimilarity of the trial key with respect to genuine key is shown in Figure 8c. In this case, maximum 67.19% bits of the genuine key is unchanged for impostor's key and up to 65.63% bits of the genuine key is changed in impostor's key. Whereas, mean value of the Hamming distances and standard deviation of the distributions are almost similar as in other cases.

### 4.8 Case 5: One genuine fingerprint and shuffle key are known

In this case, we consider that the impostor knows the cancelable template of either sender or receiver and the shuffle key ($K_{\text{shuf}}$) to be used in the key generation. In this case, one fingerprint along with transformation key (either $S_k$ or $R_k$) and $K_{\text{shuf}}$ are common in genuine and impostor keys. Hamming distances are computed and plotted using histogram. The histogram in Figure 8d shows that the knowledge of fingerprint and shuffle key helps the attacker slightly. In this case, minimum 31.25% bits of the genuine key is changed with respect to impostor's key and maximum 65.23% bits of the impostor's key are different from genuine key.

### 4.9 Case 6: Both genuine fingerprints are known but shuffle key is unknown

Let us consider that attacker has complete knowledge about both genuine fingerprints or cancelable templates but no knowledge about shuffle key $K_{\text{shuf}}$. In this case, we have used same cancelable templates (i.e., same fingerprints with same $S_k$ and $R_k$) with ten different shuffle keys to generate ten impostors keys and compare the similarity between genuine key and these impostors keys with respect to Hamming distance. The histogram in Figure 8e shows that an attacker is not able to compromise the cryptographic key even when both fingerprints/templates are known but the shuffle key is unknown. The average Hamming distance between a geneuine and impostor key is 49.99% of the length of cryptographic key. The most of the impostor's keys (99.86% of impostor's keys) are reported a dissimilarity between the range of 40% to 60% with respect to genuine key.

From our experimental results, we may conclude that an attacker is not able to generate the cryptographic key without the complete knowledge about both shuffle key and fingerprints of sender and receiver. According to our experiment, for all cases, minimum Hamming distance is 31.25% (case 5), that is, 80 bits of the impostor's key are mismatched and maximum Hamming distance is about 65.63% (case 4) that is, 168 bits of the impostor's key are needed to correct to break the genuine key. It is also observed that average Hamming distance is about 50% which means 128 bits among 256 bits of the genuine key are dissimilar on an average case. If we consider the average case, then at least $2^{128}$ trials are required in brute force attack to break the cryptographic key.

In order to evaluate the proposed method on the basis of execution time, we have consider the time to extract features from fingerprint, steganographic encoding, and decoding time and time for cryptographic key generation from two minutiae data sets. The computation time of our approach is given in Table 3, and it is observed that maximum time is required for steganographic encoding and decoding and minimum time is required for key generation from cancelable template.

## 5 Security analysis

In this section, security efficacy of the proposed approach is analyzed under different conditions. In our work, both communicating parties exchange their fingerprint data after transformation and mutually agree on two secrets like stego key ($K_g$) and shuffle key ($K_{\text{shuf}}$).

### 5.1 Privacy of fingerprints

We use least significant bit (LSB)-based steganography to exchange fingerprint data between sender and receiver. In our proposed approach, stego key ($K_g$) is used to hide fingerprint data in the pixels which are randomly chosen from cover image. For each communication, a unique synthetic fingerprint image ($I$) is used as cover image. An eavesdropper does not suspect the existence of the genuine fingerprint data (i.e., cancelable template) due to high imperceptible stego image ($I_{\text{stego}}$). The cover image is not required to decode the hidden data from stego image. An adversary, with sufficient knowledge of decoding methods, is not able to extract the correct fingerprint data from the stego images $I^S_{\text{stego}}$ and $I^R_{\text{stego}}$ of sender and receiver, without possessing the $K_g$. Our experimental result shows that only 0.47% pixels of cover image is modified, which assures that the conventional *Targeted Steganalysis* is not able to detect the existence and meaning of the hidden data [42]. The statistical steganalysis-like *histogram attack* [43] detects only sequential embeddings but it does not detect the random embedding of small size message ( i.e.,

**Table 3 Computation time of our approach**

| Operations | Time (in sec)[a] |
|---|---|
| Feature extraction | 0.05 |
| Cancelable template generation | 0.002 |
| Steganographic encoding and decoding | 0.15 |
| Cryptographic key generation | 0.002 |
| Total time | 0.204 |

[a]The experiment is conducted with intel® Core™2 Duo processor with 2.4 GHz clock speed running with Windows 7 OS.

size <50% of available LSB of $I$). Another steganalysis-like *sample pair analysis* [44] is able to detect message embedding of size up to 5% of the available embedding space (LSB) of cover image while *RS analysis* [45] can detect embedding of message of size 2% to 5%.

Both sender and receiver share their fingerprint data in a transformed format, that is, cancelable template. Receiver is not able to derive the minutiae points of sender's fingerprint from cancelable template of sender and vice versa. It assures that fingerprint identity of one user is not disclosed to other user. Even if the transformation process and transformation key are disclosed, the attacker will not be able to compute the entire set of original minutiae points from the cancelable template. This can be argued as follows.

- According to the nature of transformation key, some cells may replace multiple cells and some cells may not replace any cell but they may be replaced by other cells. For example, say, there are four cells (1, 2, 3, 4) and they are replaced by (3, 2, 4, 3). Now it is found that cell 3 replaces cells 1 and 4, and cell 1 does not replace any cell. If transformation key and cancelable template both are compromised, then attacker may know the minutiae points belongs to cell 2, 3, and 4 but not of the cell 1.
- In fact, an attacker needs to know the complete information about the fingerprint image size, that is, height and width of the image to compute the minutiae points accurately.

The cancelable templates ($T_{\mathrm{CS}}$, $T_{\mathrm{CR}}$) are fused, and the resultant template ($T_f$) of fusion is shuffled and hashed to generate cryptographic key ($K$). Therefore, the cryptographic key is non-invertible, which confirms that the cryptographic key does not leak any information about the fingerprints of users.

### 5.2 Security of cryptographic key

In our approach, the cryptographic key is generated from the combination of two fingerprints of sender and receiver. The key is not shared by them but generated at their end separately. There is no need to store the key for the use in decryption. So the key is secured from any attack. An attacker needs to compromise either (stego key $K_g$, sender's stego image $I^S_{\mathrm{stego}}$, receiver's stego image $I^R_{\mathrm{stego}}$, shuffle key $K_{\mathrm{shuf}}$) or (sender's fingerprint $F_S$, receiver's fingerprint $F_R$, transformation key of sender $S_K$, transformation key of receiver $R_K$, $K_{\mathrm{shuf}}$) to generate the genuine cryptographic key ($K$). Otherwise, the following conditions arise for different attacks.

#### 5.2.1 Known stego key attack

It means that the stego key $K_g$ is compromised by eavesdropper who eavesdrops both stego images $I^S_{\mathrm{stego}}$ and $I^R_{\mathrm{stego}}$. Next, as the shuffle key $K_{\mathrm{shuf}}$ is unknown, he has to guess the $K_{\mathrm{shuf}}$ which is of 200 bits. It requires a trial of $2^{200}$ to break the $K_{\mathrm{shuf}}$ using brute force attacks. Indeed, without $K_{\mathrm{shuf}}$, it is almost impossible to compute the original cryptographic key even if both fingerprints data are known.

#### 5.2.2 Known shuffle key attack

It happens when the token is stolen or lost and the attacker gets access to the shuffle key ($K_{\mathrm{shuf}}$) but he has no knowledge about the fingerprint data of genuine sender and receiver. In this case, the attacker has to guess $50 + 50 = 100$ minutiae points for two fingerprints. Otherwise, he has to compute the key using impostor's fingerprints which may be available to him. As the fingerprint is unique for a person, there is a rare chance to get the same fingerprint from impostor's fingerprint database. Even in our experiment, there is no collision between genuine cryptographic key and impostor's cryptographic key when the $K_{\mathrm{shuf}}$ is known. The shuffle key is updated in every session, which guarantees that it is impractical to compute the $K_{\mathrm{shuf}}$ of previous session or the same of future sessions.

#### 5.2.3 Known fingerprint attack

In this attack, there are two scenarios that may occur. In the first scenario, only the fingerprints of both parties are compromised but transformation key is not known by attacker. In the second scenario, the attacker has complete knowledge about the fingerprints along with transformation key, that is, the attacker is able to generate cancelable templates from fingerprints (of both sender and receiver) using the transformation keys and algorithm of transformation.

In the first case, attacker is not able to generate even cancelable template from the knowledge of genuine fingerprint. An adversary should know two parameters (i.e., fingerprint and transformation key) along with transformation function ($f_c$), to generate cancelable template from fingerprint template. Now according to the condition, adversary knows $F_S$, $F_R$, and $f_c$ but does not have any idea about transformation keys ($S_K$, $R_K$). Similarly, to generate the cryptographic key, an adversary should know another parameter which is shuffle key ($K_{\mathrm{shuf}}$) also. The key generation function $f_k$ is known as it is public, but all other three parameters are to be compromised by the attacker.

$$\begin{aligned} K &= f_k\{f_c(F_S, S_K), f_c(F_R, R_K), K_{\mathrm{shuf}}\} \\ &= f_k\{f_c(F_S, ?), f_c(F_R, ?), ?\} \end{aligned}$$

where $K$ is the cryptographic key and '?' marks represent those parameters which are unknown to the adversary.

In the second case,

$$\begin{aligned} K &= f_k\{f_c(F_S, S_K), f_c(F_R, R_K), K_{\text{shuf}}\} \\ &= f_k\{T_{CS}, T_{CR}, ?\} \end{aligned}$$

only one parameter ($K_{\text{shuf}}$) which is unknown to the adversary. Then, the attacker has to break the $K_{\text{shuf}}$ which needs $2^{200}$ trials in *brute force* attack.

#### 5.2.4 Known key attack

In this case, we consider that the shuffle key $K_{\text{shuf}}$ and or cryptographic key ($K$) are compromised by the attacker. Our proposed approach assures that compromise of ($K_{\text{shuf}_i}$ or $K_i$) of $i$th session does not affect the previous or future session as $K_{\text{shuf}_i} \neq K_{\text{shuf}_{i+1}} \neq K_{\text{shuf}_{i-1}}$ or $K_i \neq K_{i+1} \neq K_{i-1}$ . In our approach, both $K_{\text{shuf}}$ and $K$ are updated session wise. Initially, $K_{\text{shuf}}$ is randomly generated and time to time it is updated with the help of $F_{\text{code}}$. Cryptographic key ($K$) is revocable and is used as a session key for symmetric cryptography. When the session is over, the current session key is destroyed.

#### 5.2.5 Resists replay attack

Our approach can prevent replay attack using session key. In every session of communication, a unique session key is used to establish a secure communication between sender and receiver and the session key is destroyed after the session. Our proposed approach is able to generate $100! \times 100!$ different cryptographic keys from two fingerprint biometric traits. If an eavesdropper wants to make replay attack using a message previously transmitted by legal users, then it will make no sense to the legal user as the cryptographic key is changed. Even when stego image is used for replay attack, eavesdropper is not able to decode the stego image without the stego key.

#### 5.2.6 Resists man-in-middle attack

In our approach, fingerprints of communicating parties are transmitted over communication channel using data hiding scheme. If the man-in-middle (*MiM*) eavesdrops the stego image and is able to decode the hidden data by any means, then the *MiM* requires the perfect knowledge of secret shuffle key ($K_{\text{shuf}}$). Otherwise, he is not able to generate the genuine key ($K$) and, as a result, is not able to decrypt the ciphertext sent by genuine sender.

Case 1: The *MiM* is able to receive messages exchanged between sender and receiver. In the worst case, we also consider that the *MiM* also knows the stego key $K_g$.

$$\begin{aligned} T'_{\text{CS}} &= D_s\left(I^S_{\text{stego}}, K_g\right) \\ T'_{\text{CR}} &= D_s\left(I^R_{\text{stego}}, K_g\right) \\ K_{\text{est}} &= f_k\left(T'_{\text{CS}}, T'_{\text{CR}}, K'_{\text{shuf}}\right) \end{aligned}$$

where $D_s$ is the data unhiding function, $T'_{\text{CS}}$ and $T'_{\text{CR}}$ are the cancelable templates of sender and receiver, respectively, extracted by *MiM*, whereas $K'_{\text{shuf}}$ is the estimated shuffle key by the *MiM*. If $K_{\text{shuf}} \neq K'_{\text{shuf}}$, then the estimated cryptographic key $K_{\text{est}}$ does not match with the genuine key $K$.

Case 2: The *MiM* receives the messages $I^S_{\text{stego}}$ and $I^S_{\text{stego}}$ sent by sender and receiver, respectively, but sends a stego image, where some random data is encoded by the *MiM* to both parties. The *MiM* acts like receiver to genuine sender and behaves like sender to genuine receiver. The *MiM* is able to receive the encrypted message sent by any genuine party.

$$\begin{aligned} K^S &= f_k(T_{\text{CS}}, T_{\text{mim}}, K_{\text{shuf}}) \\ K^R &= f_k\left(T_{\text{CR}}, T_{\text{mim}}, K_{\text{shuf}}\right) \\ K^S_{\text{est}} &= f_k\left(T'_{\text{CS}}, T_{\text{mim}}, K'_{\text{shuf}}\right) \\ K^R_{\text{est}} &= f_k\left(T'_{\text{CR}}, T_{\text{mim}}, K'_{\text{shuf}}\right) \end{aligned}$$

where the cancelable template of *MiM* is $T_{\text{mim}}$ and $K^S$, $K^R$ are the keys generated by sender and receiver, respectively, and used in encryption. $K^S_{\text{est}}$ and $K^R_{\text{est}}$ are the estimated key by the *MiM* and used for decryption.

Now, the *MiM* tries to decrypt the encrypted messages (i.e., $\text{Cipher}_S$ of sender, $\text{Cipher}_R$ of receiver) received from sender and or receiver in the following ways,

$$\begin{aligned} \text{Cipher}_S &= E_{K^S}(\text{message}_S) \\ \text{Cipher}_R &= E_{K^R}(\text{message}_R) \\ \text{message}^m_S &= D_{K^S_{\text{est}}}(\text{Cipher}_S) \\ \text{message}^m_R &= D_{K^R_{\text{est}}}(\text{Cipher}_R) \end{aligned}$$

In the estimation of cryptographic keys $\left(K^S_{\text{est}}, K^R_{\text{est}}\right)$, *MiM* trials the $K_{\text{shuf}}$ which produces different keys, (i.e., $\left(K^S_{\text{est}} \neq K^S\right)$, and $\left(K^S_{\text{est}} \neq K^R\right)$), which do not decrypt the ciphertexts correctly (i.e., $\text{message}_S \neq \text{message}^m_S$ and $\text{message}_R \neq \text{message}^m_R$). In this way, our approach resists man-in-middle attack.

## 6 Conclusions

Cryptographic key generation and subsequently its maintenance are the two important issues in traditional cryptography. A cryptographic key should be generated in such a way that it is hard enough to guess and then it should be managed without any overhead of users. This work addresses these issues and propose a novel approach to generate random cryptographic key using fingerprint biometric of sender and receiver.

In our works, the privacy and security of fingerprint data are provided with cancelable template. Also, we propose a protocol with which key can be revoked thus addressing the limitation of irrevocability property of biometric

trait. More significantly, there is no need to store the key, prior to communication. In fact, our protocol adds more security allowing to generate different keys in different sessions. The proposed crypto-biometric system is resilient to many attacks such as known key attacks, replay attack, man-in-middle attacks, etc. Our proposed approach thus provides an effective solutions where we need a session-based cryptographic key during message transmission over an insecure network channel.

**Competing interests**
The authors declare that they have no competing interests.

**Author details**
[1]Department of Computer Science and Engineering, Govt. College of Engineering and Textile Technology, 4, Cantonment Road, Berhampore-742101, West Bengal, India. [2]School of Information Technology, Indian Institute of Technology Kharagpur, Kharagpur-721302, West Bengal, India. [3]Department of Information Technology, Jadavpur University, Kolkata-700098, India.

Received: 15 May 2014 Accepted: 11 March 2015
Published online: 03 April 2015

**References**

1. W Stallings, *Cryptography and Network Security: Principles and Practice, 5e*. (Prentice Hall, 2010)
2. Advance Encryption Standard (AES), *Federal Information Processing Standards Publication 197*. (United States National Institute of Standards and Technology (NIST), November 26, 2001)
3. K Mitnick, W Simon, S Wozniak, *The Art of Deception: Controlling the Human Element of Security*. (Wiley, New York, 2002)
4. DV Klein, in *Proceedings of the 2nd USENIX Security Workshop (Portland)*. Foiling the cracker: A survey of, and improvements to, password security, (1990), pp. 5–14
5. F Hao, R Anderson, J Daugman, Combining Crypto with Biometrics Effectively. IEEE Trans. Comput. **55**(9), 1081–1088 (2006)
6. U Uludag, S Pankanti, S Prabhakar, AK Jain, Biometric Cryptosystems: Issues and Challenges. Proc. IEEE. **92**(6), 948–960 (2004)
7. D Maltoni, D Maio, AK Jain, S Prabhakar, *Handbook of Fingerprint Recognition*. (Springer- Verlag, New York, 2003)
8. K Nandakumar, A Jain, S Pankanti, Fingerprint-based fuzzy vault: Implementation and performance. IEEE Trans. Inf. Forensics Secur. **2**(4), 744–757 (2007)
9. N CT Charles, DJ Kiyavash, in *Proceedings of the 2003 ACM SIGMM workshop on Biometrics methods and applications*. Lin, Secure smartcardbased fingerprint authentication (ACM New York, NY, USA, 2003), pp. 45–52
10. S Yang, I Verbauwhede, in *Proceedings (ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing,* Vol. 5. Automatic secure fingerprint verification system based on fuzzy vault scheme (IEEE Philadelphia, Pennsylvania, USA, 2005), pp. v/609–v/612
11. S Dutta, A Kar, BN Chatterji, NC Mahanti, in *Proc. Adv. Concepts Intell. Vis. Syst.*, LNCS 5259. Network Security Using Biometric And Cryptography (Springer Berlin Heidelberg, 2008), pp. 38–44
12. A Jagadeesan, K Duraiswamy, Secured Cryptographic Key Generation from Multimodal Biometrics: Feature Level Fusion of Fingerprint and Iris. Int. J. Comput. Sci. Inform. Secur. **7**(2), 28–37 (2010)
13. A Jagadeesan, T Thillaikkarasi, K Duraiswamy, Cryptographic Key Generation from Multiple Biometrics Modalities: Fusing Minutiae with Iris Feature. Int. J. Comput. Appl. **2**(6), 16–26 (2010)
14. F Monrose, MK Reiter, Q Li, S Wetzel, in *Proceedings of IEEE Symposium on Security and Privacy*. Cryptographic key generation from voice (IEEE Computer Society Washington, DC USA, 2001), pp. 202–213
15. H Feng, CC Wah, Private key generation from on-line handwritten signatures. Inform Manag. Comput. Secur. **10**(4), 159–164 (2002)
16. B Chen, V Chandran, in *Proceedings of 9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications*. Biometric Based Cryptographic Key Generation from Faces (Glenelg Australia, 2007), pp. 394–401
17. SVK Gaddam, M Lal, Efficient Cancellable Biometric Key Generation Scheme for Cryptography. Int. J. Netw. Secur. **11**(2), 57–65 (2010)
18. AK Jain, K Nandakumar, A Nagar, in *Security and privacy in biometrics*. Fingerprint Template Protection: From Theory to Practice (Springer London, 2013), pp. 187–214
19. NK Ratha, S Chikkerur, JH Connell, RM Bolle, Generating Cancellable Fingerprint Templates. IEEE Trans. Pattern Anal. Mach. Intell. **29**(4), 561–572 (2007)
20. A Ross, AK Jain, Information fusion in biometrics. Pattern Recognit. Lett. **24**, 2115–2125 (2003)
21. A Bodo, Method for Producing a Digital Signature with Aid of Biometric Feature. German Patent DE 4243908A1 (1994)
22. JH RM Bolle, S Connell, NK Pankanti, AW Ratha, *Senior, Guide to Biometrics*. (Springer-Verlag, New York, 2003)
23. NK Ratha, JH Connell, R Bolle, Enhancing Security and Privacy in Biometric-Based Authentication System. IBM Syst. J. **40**(3), 614–634 (2001)
24. S Kanade, D Camara, E Krichen, D Petrovska-Delacrētaz, B Dorizzi, Evry F, in *Proceedings of 6th Biometrics Symposium (BSYM 2008)*. Three Factor Scheme for Biometric-Based Cryptographic Key Regeneration Using Iris (Tampa, Florida, USA, 2008), pp. 59–64
25. S Kanade, D Petrovska-Delacrētaz, B Dorizzi, in *Proceedings of Fourth IEEE International Conference on Biometrics: Theory Applications and Systems (BTAS), Washington, DC, USA, 2010*. Generating and sharing biometrics based session keys for secure cryptographic applications, (2010), pp. 1–7
26. A Jain, U Uludag, Hiding Fingerprint Minutiae in Images. in Proceedings of Third Workshop on Automatic Identification Advanced Technologies (AutoID), (Tarrytown, New York. USA, 97–102 (2002)
27. A Jain, U Uludag, in *Proceedings of 16th International Conference on Pattern Recognition,* vol. 3. Hiding a Face in a Fingerprint Image (Canada, 2002), pp. 756–759
28. A Jain, U Uludag, Hiding Biometric Data. IEEE Trans. Pattern Anal. Mach. Intell. **25**, 1494–1498 (2003)
29. N Agrawal, M Savvides, in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Biometric data hiding: A 3 factor authentication approach to verify identity with a single image using steganography, encryption and matching (Miami Beach Florida, 2009), pp. 85–92
30. A Juels, M Wattenberg, in *Proc. 6th ACM Conf. Computer and Communications Security*, ed. by G Tsudik. A fuzzy commitment scheme (ACM New York, NY, USA, 1999), pp. 28–36
31. C Rathgeb, A Uhl.Context-based biometric key generation for Iris. IET Comput. Vis. **5**(6), 389–397 (2011)
32. C Yao-Jen, W Zhang, T Chen, in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME'04), Taipei, 2004,* Vol. 3. Biometrics-based cryptographic key generation (IEEE, 2004), pp. 2203– 2206
33. N Lalithamani, KP Soman, in *The 2nd International Conference on Computer Science and Information Technology*. Towards Generating Irrevocable Key for Cryptography from Cancelable Fingerprints (Beijing China, 2009), pp. 563–568
34. N Lalithamani, KP Soman, Irrevocable Cryptographic Key Generation from Cancelable Fingerprint Templates: An Enhanced and Effective Scheme. Eur. J. Sci. Res. **31**(3), 372–387 (2009)
35. V Lokeswara Reddy, A Subramanyam, P Chenna Reddy, Implementation of LSB Steganography and its Evaluation for Various File Formats. Int. J. Adv. Netw. Appl. **02**(05), 868–872 (2011)
36. A Ross, K Nandakumar, AK Jain, *Handbook of Multibiometrics*. (Springer-Verlag, Berlin, Germany, 2006)
37. Fingerprint Verification Competition FVC2000, [Online]. Available: http://bias.csr.unibo.it/fvc2000
38. Fingerprint Verification Competition FVC2002, [Online]. Available: http://bias.csr.unibo.it/fvc2002
39. Fingerprint Verification Competition FVC2004, [Online]. Available: http://biometrics.cse.msu.edu/fvc04db/index.html
40. R Cappelli, A Erol, D Maio, D Maltoni, in *Proceedings of 15th International Conference on Pattern Recognition, 2000,* vol. 3. Synthetic Fingerprint Image Generation, (2000), pp. 475–478
41. C Watson, M Garris, E Tabassi, C Wilson, M McCabe, S Janet, Ko K, *User's Guide to NIST Biometric Image Software (NBIS)*. (National Institute of Standards and Technology, Gaithersburg, MD, 2007)

42. A Rocha, W Scheirer, T Boult, S Goldenstein, Vision of the unseen: Current trends and challenges in digital image and video forensics. ACM Comput. Surv. **43**(4), 26:1–26:42 (2011)
43. A Westfeld, A Pfitzmann, in *Proc. Information Hiding, 3rd Int'l Workshop*. Attacks on Steganographic Systems (Springer Verlag, 1999), pp. 61–76
44. S Dumitrescu, Wu Xiaolin, N Memon, in *International Conference on Image Processing, (ICIP 2002)*, vol.3, no., 24-28. On steganalysis of random LSB embedding in continuous-tone images (Rochester, New York, USA, 2002), pp. 641–644
45. J Fridrich, M Goljan, Du Rui, Detecting LSB steganography in color, and gray-scale images. MultiMedia IEEE. **8**(4), 22–28 (2001)