Journal of Ambient Intelligence and Humanized Computing (2020) 11:1495–1509
https://doi.org/10.1007/s12652-019-01437-5

ORIGINAL RESEARCH

A fingerprint based crypto-biometric system for secure communication

Rudresh Dwivedi[1] · Somnath Dey[1] · Mukul Anand Sharma[1] · Apurv Goel[1]

Received: 28 December 2018 / Accepted: 28 August 2019 / Published online: 4 September 2019

Abstract
To maintain secrecy of information during communication, cryptography is considered to be an impressive solution and cryptographic keys play an important role to ensure the security. However, these randomly derived keys (of 256 bits) are hard to memorize. Also, there is a threat of privacy invasion since the storage, protection and transmission of a key over a communication link may lead to information leakage. Therefore, researchers propose to utilize user's biometric trait to generate the cryptographic key in a session-based communication environment. This avoids the storage of cryptographic keys without negotiating on secrecy. The biometric-based key generation encompasses concerns over biometric template protection, biometric data sharing between users and revocable key generation from biometric. To address the aforementioned concerns, we propose a framework for secure communication between two users using a fingerprint-based crypto-biometric system. First, the feature bit-string are computed from the users' fingerprint. Next, revocable transformation is applied to derive the private keys of respective users. Then, the Diffie–Hellman (DH) algorithm is used to generate public keys from private keys of both sender and receiver, which are shared and further used to produce a symmetric cryptographic key at both ends. Here, the biometric data is neither stored nor shared which ensures the security of biometric data. Also, perfect forward secrecy is achieved using session keys. This work also provides the long-term protection of messages communicated between two users. It is evident from the experimental evaluation over four datasets of FVC2002, four datasets of FVC 2004, and NIST special database IV that the proposed framework is privacy-preserving and could be utilized for real access control systems.

Keywords Biometric · template security · Diversity · Fingerprint · Minutiae · Revocability · Biometric cryptosystem

1 Introduction

1.1 Background

The identity of a user is lost if the user's original biometric information is compromised. The authentication systems which integrate biometric traits with cryptography are called crypto-biometric systems (Hao et al. 2006). For better security of a cryptographic system, keys used for encryption and decryption must be long enough to be unbreakable. Knowledge-based (user has to remember the key) and possession-based (key stored in smart card etc.) authentication systems are not secure since long keys cause user inconvenience to remember and smart cards can be stolen or misplaced. Moreover, storing long keys on a system is costly and not secure. Biometrics-based authentication systems can mitigate the limitations of the above-mentioned systems (Uludag et al. 2004). The user's biometric is integrated with cryptography using either key-generation techniques in which cryptographic key is generated from one's biometric or key-binding schemes in which cryptographic key is integrated to the raw biometric data (Rathgeb and Uhl 2011b). In case, user A wants to send a message to user B, A first encrypts the message using a key K and then sends this encrypted message to B. B can decrypt this message using key K only. For this, either the key K or some information to generate the same key K at both ends (A and B) must be shared between two communicating users. In both cases, the sharing of information is required. Therefore, there is a need for secure information sharing over the non-secure communication channel.

✉ Rudresh Dwivedi
rudresh.dwivedi@sot.pdpu.ac.in; phd1301201006@iiti.ac.in

Somnath Dey
somnathd@iiti.ac.in

[1] Discipline of Computer Science and Engineering, Indian Institute of Technology Indore, Simrol, Indore 453552, India

1.2 Existing approaches

In biometric cryptosystems (BCs), biometric data is combined using a cryptographic key either in the key generation or in the key binding based scenario to provide security and privacy in user authentication. As described in Sect. 1, the cryptographic keys are generated from biometric data of end users by utilizing a one way hash function or user-defined algorithm in key generation schemes whereas key-binding systems transform the biometric information using a key.

In literature, a limited number of techniques have been proposed to generate the cryptographic key from biometric traits (Monrose et al. 2001; Feng and Wah 2002; Rathgeb and Uhl 2011a; Chen and Chandran 2007). Monrose et al. (2001) proposed a technique which records a user's voice while uttering a password. Different segments of a password are mapped to a random look-up table to derive the cryptographic key. Feng and Wah (2002) proposed a technique which incorporates the dynamic information such as velocity, pressure, altitude, and azimuth from signature biometric. Feature coding has been employed for quantization of each feature into bits-string. Further, concatenation is performed to form a cryptographic key. Chen and Chandran (2007) utilized Radon transform onto 3D face data to produce 1-D bit string. Further, Advanced Encryption Standard (AES) algorithm of 128-bits is exploited to derive sufficiently long key. Rathgeb and Uhl (2011a) proposed a technique which derives iriscodes using the method implemented by Masek (2003). Next, the most stable bits within iris codes are selected, and their positions are utilized to construct biometric keys.

In key-binding schemes, Soutar et al. (1998) proposed a technique which links the biometric feature string with an N-bit cryptographic key. During linking operation, redundant information is inserted by using a repetitive code structure. Next, the hash of cryptographic key is stored along with the template for secure authentication. Juels and Wattenberg (1999) incorporated the fuzzy commitment scheme to bind a codeword with the witness i.e. biometric data. The hash values are stored as the commitment for authentication. Hao et al. (2006) utilized the fuzzy commitment scheme onto 2048-bit iris-codes. Next, Hadamard and Reed-Solomon error correction codes are utilized to correct bit errors. The fuzzy vault scheme introduced by Juels and Sudan (2006) is the most popular technique for the key-binding schemes. The main idea is to use the biometric information to lock a secret key. Clancy et al. (2003) applied the fuzzy vault scheme onto a set of minutiae points of a fingerprint. The minutiae positions are mapped to a polynomial and chaff points are added to construct a random vault. Reed-Solomon codes are applied to reconstruct the polynomial secure authentication. Kanade et al. (2008) proposed a three-factor key generation scheme for the iris-based authentication system. A user-specific shuffling key has been derived using a password, which is further utilized to randomize the iris code. The shuffled iriscodes reduce the spread out errors. Further, Hadamard codes are used for correcting remaining bit-errors. In another work, Kanade et al. (2010) employed the fuzzy commitment key regeneration mechanism to derive a protected template. Then, they utilized Error Correcting Codes (ECC) to yield random key. Further, a unique encoding is performed to output using the random key. Finally, a locked code is achieved by XORing pseudo code to the protected template. Srinivas et al. (2018) designed a key agreement protocol used for wireless sensor network (WSN). In their method, the Gateway node chooses a non-singular elliptic curve alongwith two hash functions. These hash functions are used to generate secure key which is used to start communication.

Researchers have also worked on key management for biometric-based authentication. Jiang et al. (2018) proposed a three-factor authentication scheme where fuzzy extractor based smart card registration and session key-based authentication is deployed. Akdogan et al. (2018) introduced two novel key agreement protocols namely server-key-agreement pure-biometric and server-key-agreement cancelable-biometric. Hash functions applied over original minutiae are utilized for key agreement. Panchal and Samanta (2018) derived codewords by applying Reed- solomon codes and cryptography key over block-based features. Codewords are encrypted using bio-crypto key and SVM based ranking is applied for verification. Very little work has been proposed about a framework for secure communication on a network using a crypto-biometric system. Barman et al. (2015b) proposed a system in which both sender and receiver exchange their cancelable biometrics using key-based steganography. Kanade et al. (2012) proposed a crypto-biometric system for establishing the secure communication session between two clients. Their method involves CARA (Central Authority for Registration and Authentication) with which the clients are registered. Barman et al. (2017) introduced a key exchange protocol to combine the biometric information of the involved users to bind a secret key which is further deployed for secure message communication. First, the fingerprint data is exploited to cancelable transformation to generate a bit-string. The derived bit-strings are then used for mutual locker and personalized locker generation. Further, the cryptographic keys are secretly exchanged using these lockers. Panchal et al. (2017) proposed a technique in which a unique code is derived from original fingerprint features using the convolution coding principle. Next, the unique code is used to derive a cryptographic key for encryption and decryption of the user's document. Murillo-Escobar et al. (2015)

implemented a fingerprint authentication system on a 32-bit micro-controller where encryption is utilized for template protection. Later, Barman et al. (2015a) proposed a scheme for symmetric cryptographic key generation which is to be used by sender and receiver for authentication and further communication. First, the protected templates are generated from sender and receiver. Then, protected templates are exchanged with each other and a master template is derived. Finally, the cryptographic key is generated from master template. The computation cost, communication complexity, and memory overhead are the limitations associated with the methods proposed in Jiang et al. (2018), Akdogan et al. (2018), and Srinivas et al. (2018).

In recent years, Barman et al. (2018) proposed a multi-user authentication scheme based on fuzzy commitment. The scheme registers each server and users at registration center. Thereafter, both server and user establish a session key for secure communication. Thereafter, Reddy et al. (2019) proposed a three-factor key agreement protocol using elliptic curve cryptography for client-server-based architecture. User registration, login, and mutual key agreement are the three phases in between client and servers. Barman et al. (2019) designed an authentication protocol using fuzzy commitment scheme for healthcare application. This scheme is an improvement over their earlier work (Barman et al. (2018)) which is susceptible by insider attack. The scheme generates a protected template and registers onto a smart card. The revocation has to be performed using an identifier (*id*) at a registration center. However, the scheme in Barman et al. (2019) does not provide smart card revocation, and implementation requires communication and memory overhead.

1.3 Motivation and contributions

In key generation schemes, the following issues may arise. First, deformations (translation, scale and rotation) in the biometric data may derive an erroneous key. Second, generation of a cryptographic key may require the transmission of biometric data over a network and finally there is a need of revocable keys since biometric data is irrevocable and irreplaceable. In key-binding based schemes, errors in the biometric data result to derive erroneous helper data affecting the overall performance of the authentication system. The crypto-biometric system providing secure communication onto a network also has some limitations. Storing of biometric templates is one of such issues which should be avoided. Further, a user has to remember the OTP for the entire session in one-time password (OTP) based communication. Also, compromise of OTP or lockcodes may result in privacy intrusion.

A crypto-biometric system for secure communication among different users requires (1) the generation of unique cryptographic keys from both sender and receiver, (2) secure transmission of keys among users and (3) to be robust enough from possible attacks such as an attack on a host, network and MiM attacks. Moreover, it should also provide privacy to the user's biometric along with generating revocable and non-invertible cryptographic key from the biometric data. This work intends to address the aforementioned concerns. In this work, a complete framework for secure communication among users on a network using crypto-biometric system has been proposed to provide perfect forward secrecy. The sedulous contribution of our method is that we do not store the cryptographic key anywhere. Also, there is no need to store the original biometric template of a user for key generation. Hence, there is no overhead of maintaining the cryptographic key or template information in our approach. Also, it provides revocability to the feature bit string which in turn aid to generate revocable symmetric keys from irrevocable biometric information. The contributions of our work are described as follows:

1. In this method, fingerprint biometric modality has been utilized to generate a symmetric cryptographic key for user authentication and communication. For this, the DH algorithm of public key cryptography has been applied for key generation.
2. The proposed method utilizes the pair-minutiae bit string based feature extraction to deal with translational, rotational and scale deformation in the biometric information thereby deriving error-free private keys.
3. The proposed crypto-biometric framework fulfills the requirement of generating a revocable and non-invertible cryptographic key which in turn provides the secure authentication and communication between sender and receiver.
4. Experimentations have been carried out onto individual datasets (i.e. DB1, DB2, DB3 and DB4) of FVC 2002 database (Maio et al. 2002), FVC 2004 (Maio et al. 2004), complete set (i.e. DB1+DB2+DB3+DB4) of FVC 2002, FVC 2004 databases and NIST special database 4 (Watson and Wilson 1992) to testify the potential robustness of our method. The evaluation confirms that our method outperforms existing state-of-the-art.
5. We have conducted analysis over secret key size, information entropy, randomness for cryptographic key and private keys to test the effectiveness of our findings. Finally, the security for cryptographic keys are analyzed against different attacks.

The rest of this paper organization is as follows. Section 2 describes the proposed approach in detail. Experimental results and security analysis of this method are presented in Sects. 3 and 4, respectively. Section 5 states the computation cost of the proposed method. Finally, the conclusions are drawn in Sect. 6 with a glimpse of future direction.

2 Proposed methodology

The proposed work initially extracts pair-minutiae features from the sender and receiver. A binary string is obtained after quantization and binning. Next, a random key based permutation is applied on feature bit string to obtain a permuted binary string. This binary string is hashed using SHA256 to generate a 256-bit private key which is used as an input to DH algorithm along with two predefined parameters to generate public keys of sender and receiver. These public keys are then shared between sender and receiver. DH algorithm utilizes the user's private key and other user's public key to generate a symmetric key at both users end. This key is termed the intermediate key which is further hashed to generate the final cryptographic key. This key is then used for encryption and decryption of information to be shared between sender and receiver.

This system also involves authentication of users before starting communication among them. For this purpose, a central authority (CA) for enrollment and verification of users has been proposed to verify both communicating entities in the framework. Here, the prevention of Man-in-the-Middle (MiM) attack is the prime motive behind CA-based authentication. To mitigate the MiM attack, a type of trust has to be established between two communicating entities before sharing of any information. The process of enrollment and verification of a user by CA is a standard process for authentication of a user in an enterprise/private network. At the time of registration, a user generates an RSA public-private key pair and shares this public key with CA along with some identification. CA registers the user with all this information and provides a signed certificate to the user. This certificate is used by users to verify each other before setting up the connection as described in Fig. 1. However, well-known Secure Sockets Layer (SSL) (Freier et al. 2011) is not employed since it is primarily incorporated by web browsers to securely establish a connection with certain domains over the inherently insecure Internet. Hence, the CA-based authentication turns out to be the optimal solution.

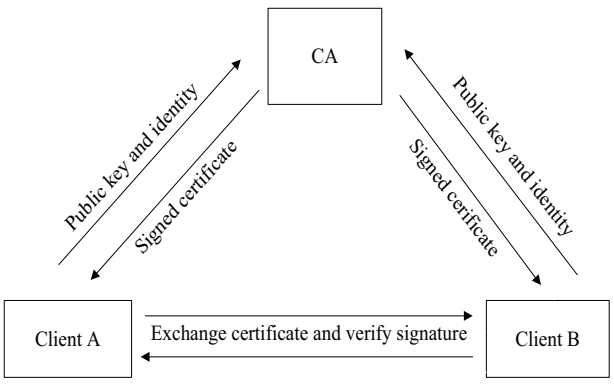

Fig. 1 Enrollment and authentication using CA

A detailed description of the above-mentioned steps of the proposed framework is stated in the following subsections. Figure 2 gives the detailed diagram of the proposed framework.

2.1 Feature extraction

The performance of the fingerprint-based verification system may degrade due to by rotation, translation and scaling deformations caused at the time of image acquisition. Hence, there is a need to evaluate translation and rotation-invariant features from the fingerprint image. For this purpose, we utilize the pair-minutiae feature extraction technique which was originally proposed by Jin et al. (2010). For better understanding, we briefly describe this procedure. Let, set of minutiae points extracted from fingerprint image are denoted as:

$$Ms = \left\{ Ms_k\left(x_k, y_k, \theta_k\right) \right\}_{k=1}^{n} \tag{1}$$

where n is number of minutiae points. (x_k, y_k, θ_k) are (x,y) coordinates and orientation of kth minutiae, respectively. A pair minutiae vector Vp_{ij} can be formed by pairing up two minutiae Ms_i and Ms_j from set Ms. There will be $\frac{n(n-1)}{2}$ pairs constituting the set Vp which can be expressed as:

$$Vp = \{ Vp_{ij} : 1 \le i,j \le n \text{ and } i \ne j \} \tag{2}$$

where each Vp_{ij} is triplet of distance and relative angles of minutiae pair (Ms_i, Ms_j), assuming the reference direction of line segment connecting minutiae pair is from Ms_i to Ms_j. Hence, Vp_{ij} is defined as:

$$Vp_{ij} = \{L, \alpha_i, \beta_j\} \tag{3}$$

where L is the distance between minutiae pairs Ms_i and Ms_j. α_i is the angle between reference direction of line segment joining Ms_i and Ms_j with the orientation of Ms_i in the counter-clockwise direction and β_j is defined analogously. Figure 3 illustrates this triplet formation.

To determine Vp_{ij}, the following two quantities X and Y are calculated first:

$$X = (x_j - x_i)cos\theta_i + (y_j - y_i)sin\theta_i$$
$$Y = (x_j - x_i)sin\theta_i - (y_j - y_i)cos\theta_i$$

Next, the triplet contained in $Vp_{ij} = (L, \alpha_i, \beta_j)$ is obtained as follows:

$$L = \sqrt{X^2 + Y^2} \tag{4}$$

$$\alpha_i = \arctan\left(\frac{Y}{X}\right) \tag{5}$$

$$\beta_j = \alpha_i + \theta_j - \theta_i \tag{6}$$

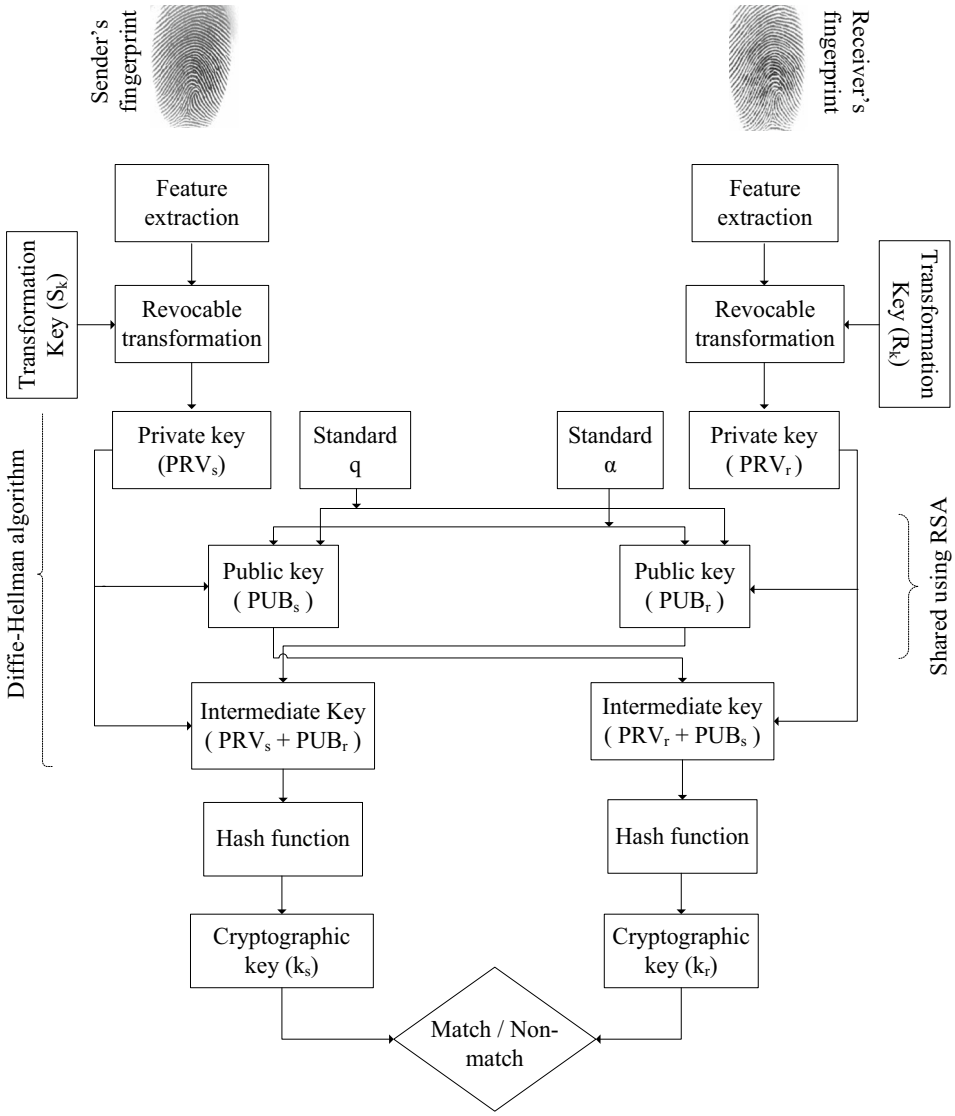

Fig. 2 Proposed crypto-biometric system framework

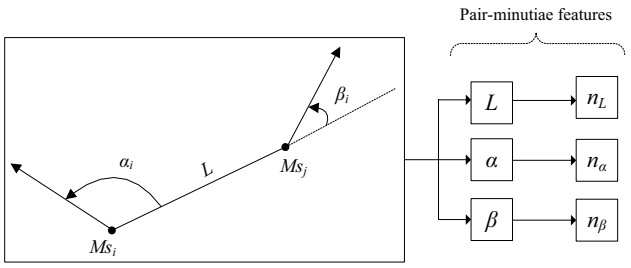

Fig. 3 Pair-minutiae feature extraction

After the evaluation of (L, α_i, β_j), quantization is applied on each Vp_{ij} in Vp. (L, α_i, β_j) are represented in binary notation by choosing a quantization step size. Suppose n_l, n_α and n_β are number of bits required to represent L, α and β in binary notation, respectively. Then the total number of bits to represent each Vp_{ij} in Vp is represented as:

$$n_p = n_l + n_\alpha + n_\beta \tag{7}$$

Thus, for each pair-minutiae vector Vp_{ij} in Vp, a bit-string $Vp_{ij}^{(b)}$ of n_p bits is derived. The set $Vp^{(b)}$ represents the set of $Vp_{ij}^{(b)}$ as follows:

$$Vp^{(b)} = \{Vp_{ij}^{(b)} \ : \ 1 \leq i,j \leq n \text{ and } i \neq j\} \tag{8}$$

Empirical evaluations find that $n_p = 15$ (i.e. 5 bits for each L, α and β provides the optimal equal error rate (EER) and the maximum entropy (please see Sect. 3.6). Further, binning is applied on binarized pair-minutiae vector set. Since there are 2^n possible combinations of n bits, binning starts from $00\cdots0$ to $11\cdots1$. For each $Vp_{ij}^{(b)}$ in $Vp^{(b)}$,we index a bin by 1 if Vp_{ij} falls in it. The bins indexed at least once are assigned 1 and all other bins are assigned 0. At the end of this procedure, a binary string h_k of length 2^{n_p} is obtained in which 1's correspond to the unique occurrence of those

$Vp_{ij}^{(b)}$. In this work, this binary string h_k is considered as the feature vector.

2.2 Authentication system

In crypto-biometric systems, transformation either binds or derives a cryptographic key providing revocability and non-invertibility to the original biometric. The user authentication is performed using this cryptographic key which can easily go into the hands of adversaries. However, the proposed framework does not share or store biometric data. DH algorithm is used for generating symmetric keys at both sender and receiver ends. We describe the procedure in the following subsections.

2.2.1 Revocable transformation

To provide revocability to the bit string, the random permutation based transformation similar to Ratha et al. (2007) is applied on the bit string based on the user-specific key. This key is termed as a transformation key for the user. The transformation key is used as a seed value to generate random numbers equal to the length of feature bit string. Bits corresponding to these random numbers are swapped with bits at positions starting from the start and incrementing with each random number. For example, we have a 9-bit long feature bit-string. The transformation key (T) is used as a seed to generate random numbers from 1 to 9. Say, the first random number generated is 3. We swap bit at 3th position with 1st position bit. Let, the next random number be 6. Now, 2nd bit is swapped with 6th bit. This process continues till last bit is swapped with a bit at the position equal to the new random number. In this way, a bit string is derived which is the permutation of original bit string as shown in Fig. 4.

The method ensures the generation of a revocable template from feature bit-string of the user's biometric since a user can utilize different values of T to generate a different template from same feature bit string.

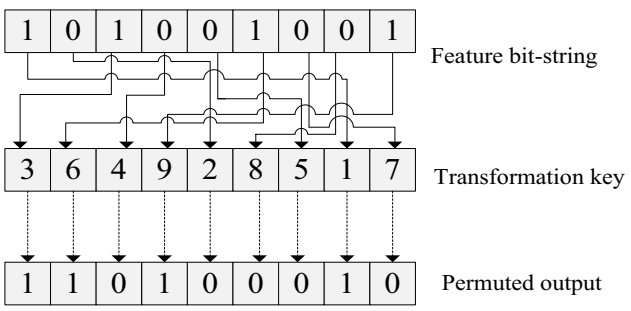

Fig. 4 An example of revocable transformation

2.2.2 Generation of public key and cryptographic key using DH algorithm

DH algorithm (Kivinen 2003) is a key exchange method which uses public key cryptography to generate a symmetric key for two users. First, both users decide on two variables, q and α, which are shared among both. Here, q is a large prime number and α is its primitive root. Then, the private keys of two users i.e. sender and receiver (PRV_s, PRV_r) are fed to the DH algorithm to generate their respective public keys (PUB_s, PUB_r). These public keys are shared with each other. Finally, DH algorithm takes sender's private key (PRV_s) and receiver's public key (PUB_r) to generate a symmetric cryptographic key, K_s. Similarly, the cryptographic key for receiver, K_r is generated using receiver's private key (PRV_r) and sender's public key PUB_s. This way, both users get the same symmetric key for secure communication without sharing their respective private keys. This key is termed as an intermediate key for the communication setup. In our approach, the size of this key is 2048 bits. This intermediate key is hashed using SHA256 to generate a 256-bit key which is the final cryptographic key. This key is then used for encryption and decryption of messages between sender and receiver.

In the DH algorithm, it is easy to calculate exponentials modulo a prime while it is very difficult to calculate discrete logarithms. For large primes, the latter task is considered infeasible. As discussed in Sect. 2.1, n is considered to be 15 in our approach to get a feature bit string of length 2^{15}. Same is the length of the revocable template which is permuted feature bit string. This large binary string needs to be mapped into a smaller one which can be used as key input for the DH algorithm. For this, SHA256 hash has been used here. The permuted binary string is hashed using SHA256 to generate a 256-bit key. This key is termed as the private key of the user. This way, private keys of the sender (PRV_S) and receiver (PRV_R) are generated.

Further, the DH algorithm requires a large prime number q and its primitive root α. These parameters are not required to be generated in each session; we can also use the fixed value of these parameters over a large number of sessions (Appendix A). In this approach, DH parameters of RFC 3526: 2048-bit MODP group (Kivinen 2003) have been used. With private keys PRV_S, PRV_R, q and α, step 4 and 5 of DH algorithm are applied to generate public keys PUB_S and PUB_R of sender and receiver respectively.

2.2.3 Authentication using CA

The proposed crypto-biometric system also involves a central authority (CA) with which all users need to be registered. If a new user joins the system, CA requires to enroll it first. Following steps are performed in the enrollment phase:

1. User generates its own set of RSA public-private key pair and sends its public key along with its identification to the CA after encrypting it with the public key of CA.
2. CA identifies the user using this information and stores its identification in its database.
3. CA computes a hash of public key and identification of the user. Next, it encrypts this hash, public key, and identification of the user using its private key.
4. This encrypted message is termed as a certificate of the user and is sent to the user.

All users enroll with the CA to get their certificates. These certificates are used for verification of other users before setting up a connection with them. Suppose user A wants to communicate with user B. For this purpose, the following steps need to be performed before setting up this connection in the verification phase:

1. A sends its certificate to the user B with a request to initiate the communication.
2. B decrypts A's certificate with CA's public key and computes the hash of A's public key and identification. This hash is matched with hash in the certificate to verify that this certificate is indeed signed by the CA.
3. B then identifies A using its identification and then send its certificate to A.
4. A does the same steps as B to verify B. Once verified by each other, they can start setting up the communication using the proposed approach discussed in the above subsections.

3 Experimental results and analysis

The proposed framework for secure communication is evaluated based on the two criteria i.e. cryptographic key randomness and performance. In the following subsections, we present the experimental results and performance of the proposed method. We use four performance metrics to evaluate the performance of our method:

1. FAR: The probability of mistakenly accepting an imposter as a genuine user
2. FRR: The probability of mistakenly rejecting a genuine user as an imposter
3. GAR: Can be calculated as 1-FRR
4. EER: The error rate where FAR and FRR hold equality

3.1 Database

The proposed method has been evaluated using the four datasets of FVC2002 (Maio et al. 2002), FVC 2004 (Maio et al. 2004) databases (i.e. DB1, DB2, DB3, and DB4) and NIST special database IV. Each dataset of FVC 2002/2004 comprises of 100 subjects with 8 impressions per subject. The performance is evaluated using FVC protocol which states that all possible unique pair of impressions from the same subjects are considered to derive genuine cryptographic keys. As a result, we obtain 2800 (i.e. ${}^{8}C_{2} \times 100$ different possible combinations out of 8 samples for 100 subjects) genuine key comparisons. Next, unique pairs of impression from different subjects are matched to derive 4950 (i.e. ${}^{100}C_{2}$ different possible comparisons) imposer key comparisons.

3.2 Randomness of the cryptographic key for the genuine pair of subjects

To evaluate the randomness in cryptographic keys for the genuine pair of fingerprints, first two fingerprints of each subject from all four datasets DB1, DB2, DB3 and DB4 are considered as a genuine pair. For each subject, the cryptographic key for the first two instances of a fingerprint is generated. Next, we evaluate the number of matching bits between these two keys. Here, we have measured the Hamming distance metric to evaluate the similarity between the two derived cryptographic keys. It computes Hamming distance (HD) i.e. sum of non-equivalent bits (exclusive-OR) between the key pair and subtracts it from total number of bits. Ease and Simplicity in computation over binary string is the rationale behind using HD. For each dataset, we evaluate percentage of matching number of bits out of a total number of bits in the generated feature string. A total of $100 \times 4 = 400$ data points are calculated and are illustrated using histograms in Fig. 5. It can be observed from the histogram that mean value for matching percentage for a genuine pair of fingerprints is 89.99% which means that the average number of matching bits is 3686 bits out of 4096 bits. The percentage of matching bits is spread between the ranges of 81.37–99.83% with a standard deviation of 0.043. Therefore, it is evident that a genuine pair achieves a sufficient number of matchable bits for the pair of the cryptographic key.

3.3 Randomness of the cryptographic key for imposter pair of subjects

To evaluate the randomness in cryptographic keys for imposter pair of users, datasets DB1, DB2, DB3 and DB4 are considered. Each dataset is divided into 50 unique pairs. For each pair, the rest of the combinations are considered imposter pairs of fingerprints. This way, a total of $50 \times 49 = 2450$ comparisons are possible for each genuine pair taking part in communication. Next, we generate the cryptographic keys for each combination of genuine and imposter pair of fingerprints in databases DB1, DB2, DB3 and DB4. Hence, a total of $2450 \times 4 = 9800$ hamming

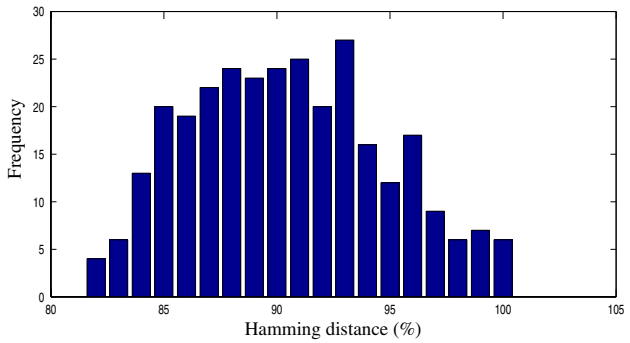

Fig. 5 Hamming distances between genuine pair of cryptographic keys

distances have been calculated which are shown in Fig. 6 using a histogram. It can be observed from the histogram that mean hamming distance is 49.94% which means that the average Hamming distance between genuine and imposter keys is 128 bits. Hamming distances are spread between the range of 37.89–61.72% with a standard deviation of 0.031. Also, it has been observed that 40–60% of the bits of genuine keys are different from 99.89% imposter keys and a small number (0.04%) of imposter keys have unmatched bits under 40%. Hence, an imposter cannot get more than 128 matched bits out of any 256-bit cryptographic key.

3.4 Randomness of the private key for different transformation key

To measure the randomness in private keys, we consider all subjects of DB1, DB2, DB3, and DB4. Next, the private key is evaluated for 30 different randomly generated transformation keys for each subject. Further, we evaluate the Hamming distances between the first private key derived using transformation T and other 30 private keys. This way, a total of 12000 hamming distances are calculated for all subjects in dataset DB1, DB2, DB3 and DB4. Histogram of the Hamming distances is shown in Fig. 7. It can be observed from Fig. 7 that mean hamming distance is 50.03% which means that the average Hamming distance between two private keys generated using different transformation keys is 128 bits. Hamming distances are spread between the range of 37.11–64.06% with a standard deviation of 0.031. For change in transformation keys, 40–60% bits of the private key are different in 99.85% of cases. Therefore, it can be deduced that at least 128 bits of the private key are altered on changing transformation key for a subject.

3.5 Key size analysis

In literature, it has been analyzed that an authentication system requires more than 2^{100} secret keys according to Alvarez and Li (2006). In his case, each key must be strong enough and should generate random data to be resilient against an exhaustive attack. In our method, we use 256-bit cryptographic key whose randomness has been tested in Sects. 3.2–3.4. Hence, 2^{256} number of different keys can be derived. The strength of derived cryptographic keys is based upon maximum Lyapunov exponent analysis as stated in Murillo-Escobar et al. (2014). The positive value along with a uniform distribution confirms the strength of secret keys. In our method, 256-bit cryptographic key is utilized with all positive bits and uniform binary distribution which confirms the secrecy strength of the key.

3.6 Information entropy analysis

Information entropy is measured by the randomness present in a cryptographic key, i.e. greater the unpredictability of the key, greater is the entropy. Otherwise, the authentication system is susceptible to an entropy attack since there exists a certain degree of predictability in the key generation. The entropy $H(PUB)$ of a key PUB can be calculated as follows:

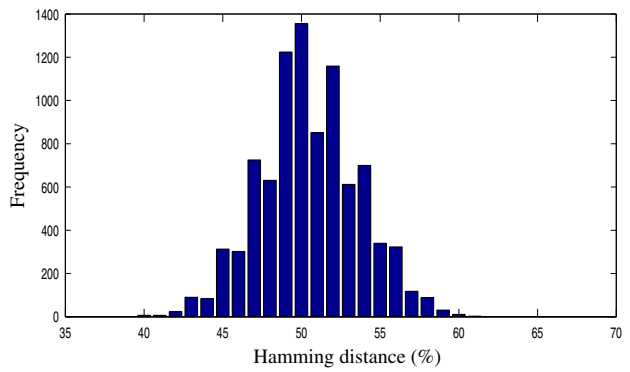

Fig. 6 Hamming distances between imposter pair of cryptographic keys

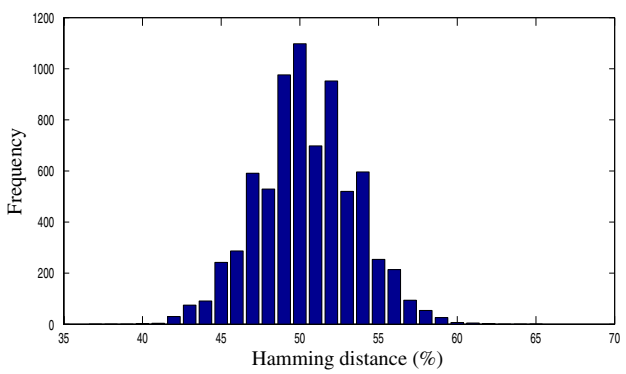

Fig. 7 Hamming distances among private keys for different transformation keys

$$H(PUB) = \sum_{i=0}^{2^N-1} p(PUB_i)\, log_2\left(\frac{1}{PUB_i}\right) \quad (9)$$

where N is the number of bits in the key PUB; 2N is all possible bits in the key, $p(PUB_i)$ represents a probability of any bit present in PUB_i. If there is a key PUB containing 2^N possible bits, the entropy should be $H(PUB)$=N ideally. The cryptographic key has 256 bits, and its maximum entropy is H = 8. This confirms that all bits appear with the same probability. The entropy of the original template is H = 5.14, whereas the entropy of the generated cryptographic key is H = 7.28 for the parameter n_p = 15. Therefore, it is ascertained that the key generation is highly pseudorandom in our method.

3.7 Performance

The proposed method has been evaluated using all datasets of (i.e., DB1, DB2, DB3 and DB4) of FVC2002 and FVC 2004. In this work, minutiae points are extracted using the commercial software VeriFinger SDK (Verifinger 2010). For each subject, hamming distances between the cryptographic keys generated from a genuine pair of fingerprints and hamming distances between the cryptographic keys generated from imposter pair of fingerprints are calculated. For a genuine pair of fingerprints, hamming distances must be minimal while for a pair of genuine and imposter fingerprint, the Hamming distance must be higher. The experimental results are obtained while maintaining same transformation key for all subjects.

FVC 2002 The datasets cover a wide range of fingerprint images in terms of quality. Among these four datasets, dataset DB3 and DB4 contains the low-quality images. We have evaluated the genuine and imposter scores using the same transformation key for each user present in the database. The ROC curves for DB1, DB2, DB3 and DB4 of FVC2002 are shown in Fig. 8.

It has been observed from Fig. 8, that we achieve GAR of 98.29% and 99.03% for the datasets DB1 and DB2, respectively due to the presence of more number of good quality images. In comparison to DB1 and DB2, relatively less number of minutiae points are extracted from fingerprints in DB3 due to the poor-quality images with spurious and missing minutiae. As a result, we achieve 95.56% and 86.4% of GAR for DB3 and DB4, respectively. The lack of reliable minutiae causes the degradation in performance. Further, we also evaluate the performance after combining all the four datasets of FVC2002 resulting in $2800 \times 4 = 11200$ genuine key comparison and $4950 \times 4 = 19800$ imposter key comparisons. The ROC curve for the whole FVC2002 database is also shown in Fig. 8. It has been observed that the GAR of the proposed method is slightly lower than the

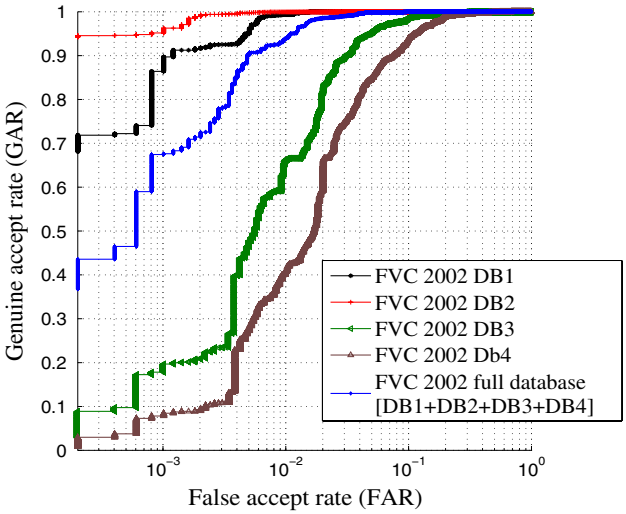

Fig. 8 ROC curves for FVC2002 datasets

GAR of the method proposed in Barman et al. (2017) since error-correction has not been applied for the derived cryptographic keys. However, the method outperforms Srinivas et al. (2018). Clancy et al. (2003) and Kanade et al. (2010) have not provided any empirical evaluation onto publicly available databases. Therefore, we have not compared our method with these methods.

FVC 2004 For all four sets of FVC 2004 database, the ROC curves are shown in Fig. 9. The database consists of images where no efforts were made to control image quality. Also, dried, moistened, and images acquired on uncleaned sensor plates are present in the database. Therefore, we achieve relatively low GAR for all datasets i.e. 97.28%, 98.02%, 96.98%, and 89.90% for DB1, DB2, DB3, and DB4, respectively. From the experimental evaluations, it is observed that the dataset DB2 outperform other three datasets since the quality of images are not much deteriorated. For DB4, we achieve a very low GAR due to the small overlap area present in the compared images. Further, we also perform experiments after combining all the four datasets. Next, we compare our results with Panchal et al. (2017) since no other researcher have utilized FVC 2004 for experimentation. We observe that our method produces better GAR than the existing method.

NIST special database In addition to the datasets of FVC2002, we also tested our method with NIST special database 4 (Watson and Wilson 1992). The ROC curve for the NIST special database 4 is illustrated in Fig. 10. It is worth mentioning that the proposed method achieves a GAR of 96.73% for partial NIST special database 4, which is better than the reported 95.12% GAR in Panchal et al. (2017). However, we obtain high FAR for NIST special database due to the absence of error-correction codes. The performance comparison of the proposed method with some existing

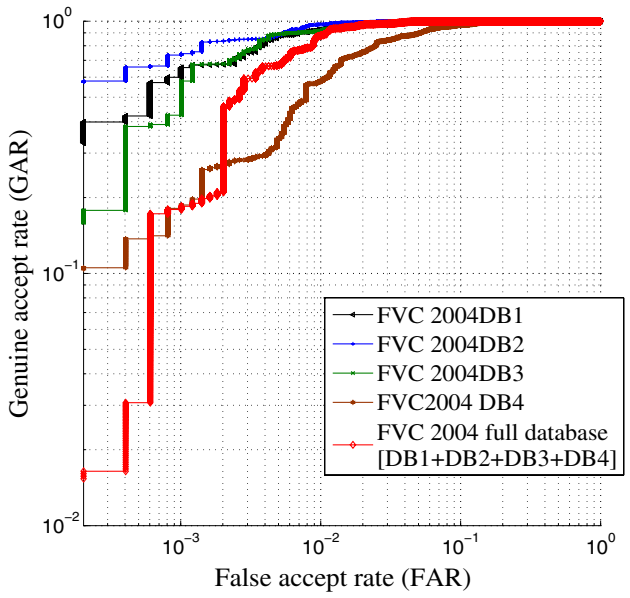

Fig. 9 ROC curves for FVC2004 datasets

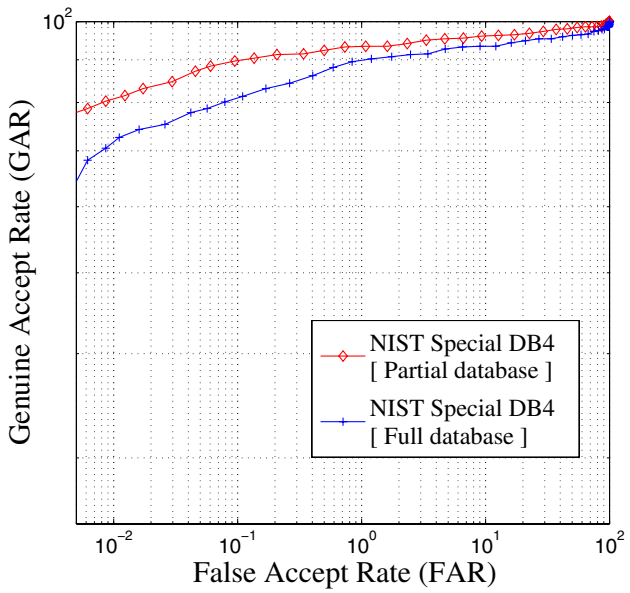

Fig. 10 ROC curves for NIST special database 4

cryptosystem design framework is reported in Table 1. We have compared our method with the approaches proposed by Barman et al. (2017) and Panchal et al. (2017) due to the scarcity of the research work carried out in this direction and the proposed work achieves the best performance over the current state-of-the-art. From the reported results in Table 1, we can observe that the performance of the proposed method for the whole FVC2002 database is slightly lower than Barman et al. (2017), yet comparable.

4 Security analysis

In our approach, the biometric templates of users are neither stored nor shared. Two users can communicate with each other without worrying about storing or sharing their biometric data. In this section, we focus on the security of the proposed framework against various possible attacks along with a formal security verification of the proposed scheme.

4.1 Security of DH algorithm

In the DH key exchange, it is relatively easy to calculate exponential modulo a prime while it is very tough to calculate discrete logarithms. For larger primes, the latter task is considered infeasible Stallings (2006). DH algorithm requires two parameters q and α. For example, prime number q be 353 and its primitive root α be 3. A and B select private keys $PRV_A = 97$ and $PRV_B = 233$ respectively. Now, the public keys (i.e. PUB_A, PUB_B) become:

$$PUB_A = 3^{97} \bmod 353 = 40$$
$$PUB_B = 3^{233} \bmod 353 = 248$$

After exchanging public keys, common secret key is:

$$K_r = (PUB_A)^{PRV_B} \bmod 353 = 248^{97} \bmod 353 = 160$$
$$K_s = (PUB_B)^{PRV_A} \bmod 353 = 40^{233} \bmod 353 = 160$$

Assume that, an attacker gets q, α, PUB_A and PUB_B. To evaluate the secret key K_r or K_s, an attacker needs to solve the expression $3^{97} \bmod 353 = 40$ or $3^{233} \bmod 353 = 248$. However, the evaluation becomes impractical for larger primes. Hence, even if an attacker gets access to the public keys, private keys cannot be generated. This ensures that an attacker would not be able to unveil original biometric template of a user in any circumstances.

4.2 Security of cryptographic key

In this approach, the cryptographic key is generated from biometric of sender and receiver using DH algorithm. This key is valid for only for a session and is destroyed as soon as the session gets over. This key is never shared or stored. Hence, there is no way possible to reveal this key. Further, we analyze the security of this system against different possible attacks.

4.2.1 Network attack

Assume that, an attacker invades the security of the network and takes control of all the information shared over

Table 1 Comparison with existing crypto-biometric systems GAR (FAR / FRR)

Databases	Barman et al. (2017)	Barman et al. (2015a)	Srinivas et al. (2018)	Panchal et al. (2017)	Proposed method
FVC2002					
DB1	–	–	–	–	98.29 (0.20/1.45)
DB2	–	–	–	–	99.03 (0.11/0.72)
DB3	–	–	–	–	95.56 (1.80/7.73)
DB4	–	–	–	–	96.4 (3.08/19.45)
DB1 + DB2 + DB3 + DB4	97 (0.562/–)	-	95.59 (0.2/0.2)	–	96.49 (0.61/2.81)
FVC 2004					
DB1	–	–	–	95 (–/–)	97.28 (1.55/2.73)
DB2	–	–	–	96.25 (–/–)	98.02 (0.73/1.81)
DB3	–	–	–	95 (–/–)	96.98 (1.84/3.02)
DB4	–	–	–	86.25 (–/–)	89.9 (4.37/10.08)
DB1 + DB2 + DB3 + DB4	–	–	–	–	96.82 (1.2/3.19)
NIST special database 4					
Partial database	–	–	–	95.12 (7.6/4.72)	96.73 (0.83/6.30)
Full database	–	–	–	–	95.89 (0.762/8.1)

the network. In our method, public keys of the sender and receiver are the only information shared over the network before generating a secure session key. Even if an attacker gets able to unveil the public keys of sender and receiver, no information can be reverse-engineered using these keys as discussed in Sect. 4.1.

4.2.2 Attack on a host

In this attack, an attacker takes control over a user/host in the network and gets all the information available at the user end. In this system, a user stores transformation key, cryptographic session key, public-private key pair and authentication certificate. With all this information, the attacker gets access only for that session. An attacker can log all the encrypted transmissions and can get stored messages of that user, but the attacker still can't decrypt these messages. As cryptographic keys are changed in each session and are not related in any way except that they are generated from the original biometric of the user, an attacker can access messages of that session only. For the decryption of messages in previous communication and encryption of messages in future conversations, an attacker still needs original biometric of the user. Nevertheless, it would be very tough for an attacker to unveil the original template of a user. This way, access to the cryptographic key of a session gives access to messages of that session only, neither the previous nor the future communications. This property is called perfect forward secrecy which the proposed system achieves.

4.2.3 Replay attack

In this attack, a falsified data is injected between the sensor and feature extractor. To avoid this, we utilize the session key between two users. For each session, a different cryptographic key is generated and destroyed after the session gets over. If an attacker eavesdrops a message previously transmitted by genuine users, it will fail since the cryptographic key is altered. Even if an attacker eavesdrops one of the public keys shared between two users to launch replay attack, it would not be possible to derive cryptographic keys as it requires user's private key along with the public key. This way, the proposed system found to be secure against replay attacks.

4.2.4 Man in the middle (MiM) attack

In the MiM attack, the attacker inserts him/herself into communication between two users, impersonates both users and gains access to information that the two users are trying to send to each other. To avoid the MiM attack, two users need to verify each other before starting communication. In this method, this verification takes place using certificates provided by a trusted certification authority (CA). This verification before communication setup makes sure that a user is communicating with the genuine user at the other end avoiding the MiM. If a MiM eavesdrops two users certificate at the time of verification and sends his certificate to both of them to setup two-way communications simultaneously, he will be verified as himself, not the genuine user with which a user wishes to setup the communication. Even for this to

happen, a MiM needs to have an authentication certificate provided by the CA which can only be provided to him after verifying his identity. Therefore, an attacker could not be able to get the certificate from CA without identifying himself. Once identified, it would not be possible to launch the MiM attack since the attacker's identity will be revealed. User's verification from CA prevents the authentication system from MiM attacks in the proposed method.

4.2.5 Privileged insider attack

To start communication, an attacker may send PRV_s to central authority. An insider user of the trusted CA may act as an attacker and a reply from CA can be recorded by him. In the proposed method, the PUB_s cannot be from PRV_s without the actual knowledge of q and α. Hence, a privileged insider cannot allow anyone to log into CA.

4.2.6 Guessing and tracking attack

In the proposed method, an attacker cannot employ guessing or tracking attack as the attacker needs to know S_k, Cancelable template, and α to know PUB_s. In addition, it is very hard to guess the biometric template since as the attacker has no information about receiver.

4.2.7 Man-at-the-end attacks

This attack may be launched in numerous ways if the attacker has physical and authorized access. At the CA side, the biometric information is assumed to be protected since it is required to have the genuine user and other credentials. Only the presence of protected template alongwith transformation keys may help the attacker to attain intermediate key. Hence, the proposed method is secured against man-at-the-end attacks.

4.2.8 Impersonation attack

In a practical crypto-biometric system, there can be two kinds of impersonation attacks are possible:

- User impersonation attack: This kind of attack is not applicable to our proposed system since it does not store any cryptographic key and biometric template anywhere.
- Server impersonation attack: In the proposed system, the mutual authentication is performed by CA. In this attack, the attacker attempts to persuade the user with a reply sent on behalf of CA. For this to happen, the attacker has to generate α and a timestamp to evaluate PUB_s. Further, intermediate key generation requires the computation of PUB_r, which is infeasible for an attacker since he does not have the knowledge of other credentials owned by receiver.

4.2.9 Ephemeral secret leakage (ESL) attack

In the proposed method, both of the communicating parties agree upon a common secret session/cryptographic key after the mutual authentication. Hence, the session/cryptographic key should be secured by a protection mechanism to resist against the attacker. In our proposed scheme, session key is computed after the mutual authentication. Session key and the credentials through which it has been computed, are derived through collision resistance one-way hash function. Due to the non-invertible characteristics of the cryptographic one-way hash function, the attacker cannot gain control over the session/cryptographic key without the knowledge of secret parameters.

4.3 Formal security verification: AVISPA simulation

At present, the formal security verification of various existing schemes (Barman et al. 2018, 2019; Reddy et al. 2019) is performed by AVISPA tool. Primarily, AVISPA utilizes High Level Protocols Specification Language (HLPSL) to validate On-the-fly Model-checker (OFMC) and Constraint-Logic-based Attack Searcher (CL-AtSe) back ends. These two back ends are tested to verify the security strength of any authentication scheme. In the protocol specification, we define three basic roles for the participants (i.e. the user role A_i, the central authority C_i and, another (communicating) user B_i. Corresponding session role, environment role and other goals are specified in HLPSL.

- Role (A_i)/role (B_i): First, after receiving the start signal, user A_i sends the request to the C_i for registration purpose, and changes its state (maintained by the variable state) from 0 to 1. Once A_i receives the authentication reply from C_i, it also changes its state from 1 to 3. During the mutual authentication and key agreement phase, Ui again sends the request to C_i.
- Role (C_i): After receiving the registration request from user A_i and updates its state (maintained by the variable state) from 1 to 2. Thereafter, it sends the authentication reply to user A_i and changes its state from 2 to 4. During the mutual authentication and key agreement phase, it sends authentication message to A_i, receives reply from A_i.

In a similar way, the mandatory roles of session, goal and environment are also defined in AVISPA. The simulation verifies the security strength over OFMC and CL-AtSe back-ends models. The simulated results are illustrated in Fig. 11. Here, the depth of search is 7 plies and the number

of total traversed nodes is 128 in OFMC model. Further, 0.27 s and 0.06 s are required to complete the search attack for OFMC and CL-AtSe backend, respectively. From the reported results, it is evident that the proposed method is secure enough against replay and MiM attacks.

5 Computational time

We also analyze the computation cost of our approach and compare with the state-of-the-art. It is noted that there are three tasks involved in the approach namely cancelable template generation, intermediate key generation, and cryptographic key generation. The consumed time in each of the above-mentioned tasks are reported in Table 2. The execution time are provided with reference to our implementation with Intel® Core TM i5 processor with 2.3 GHz clock speed in MATLAB 2014a running on Windows 10 OS. In our method, the maximum time is needed for intermediate key generation and minimum time is needed for cryptographic key generation.

In comparison, the computation cost of our method is lower than the key generation methods proposed in Barman et al. (2015a), Srinivas et al. (2018), and Panchal et al. (2017). However, the communication cost (i.e. bits required) of our method is lower than Barman et al. (2015a) and Srinivas et al. (2018) yet comparable. Therefore, our proposed method is efficient than the existing key generation scheme in terms of performance, simplicity, and computation cost.

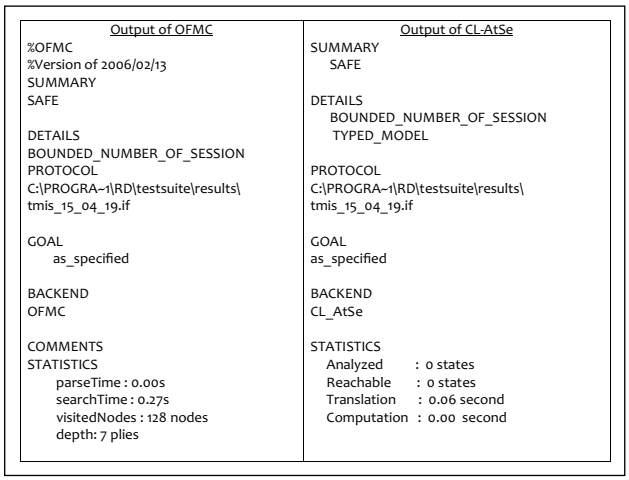

Fig. 11 Simulation results of security analysis with the AVISPA tool

6 Conclusion and future scope

Cryptographic key generation and subsequently its security are the two major issues in traditional cryptography. To address these issues, we have introduced a novel crypto-biometric framework for secure communication which incorporates the DH algorithm to generate symmetric cryptographic keys from the user's fingerprint biometric modality. Here, CA-based authentication is incorporated to prevent MiM attack whereas DH algorithm is utilized for key exchange. These existing algorithms have been deployed optimally to establish a secure transmission between two users avoiding any possibility of impersonation. To derive the private keys, we utilize invariant pair minutiae bit-string to mitigate the adverse effect over performance due to non-linear deformations at the time of acquisition. To provide revocability to the bit string, a random permutation has been applied on the bit string based on the user-specific key. Next, the revocable bit strings are fed to DH algorithm along with two predefined parameters to generate public keys of sender and receiver. DH algorithm utilizes user's private key and other user's public key to generate a session key for both user's end. This key is further hashed to generate the final cryptographic key. The private keys and cryptographic keys are assessed over different perspectives in terms of randomness, key size and information entropy to ensure the pragmatic real-world implications. Finally, we have performed an exhaustive evaluation of our method onto all four datasets i.e.DB1-DB4 of FVC2002, FVC2004, and NIST special database 4. The evaluations demonstrate that a GAR of 96.49%, 96.82% and 95.89% are obtained for complete FVC2002 database and NIST special database 4, respectively which indicates that our approach outperforms existing crypto-biometric systems. Also, the analysis of different attacks such as network attack, attack on a host, replay attack, MiM attack,

Table 2 Computation time comparisons with state-of-the-art

Methods	Computation time (in ms)				Computation cost (in bits)
	Template generation	Intermediate key generation/encoding-decoding	Cryptographic key generation	Total time	
Barman et al. (2015a)	0.05	60	0.002 ms	60.052	192
Srinivas et al. (2018)	–	–	–	29.8085	128
Panchal et al. (2017)	–	–	–	1040	1274
Proposed method	0.07	10	0.04	10.11	256

ESL attack, man-at-the-end attack, impersonation attack, and privileged insider attacks confirms the potential robustness of the proposed work. Thus, it provides an effective solution to the need of session-based secure communication setup for transmitting messages over an insecure communication channel. In future, there is a scope of optimizing this approach in terms of performance by applying error-correction codes over the cryptographic keys. Additionally, the secure crypto-biometric system design for multimodal biometric systems can be looked into the future.

Acknowledgements We would like to acknowledge the Indian Institute of Technology Indore for providing the laboratory support and research facilities to carry out this research. The authors are thankful to SERB (ECR/2017/000027), Department of science and Technology, Govt. of India for providing financial support to carry out this research work.

Appendix A

DH parameters

The following parameters (RFC 3526 : 2048-bit MODP group) were used for implementing DH algorithm.

1:] Prime number (for modulo) q:

FFFFFFFFFFFFFFFFC90FDAA22168C234C4C6628B-80DC1CD1

2 9 0 2 4 E 0 8 8 A 6 7 C C 7 4 0 2 0 B B E A 6 3 B - 139B22514A08798E3404DD

EF9519B3CD3A431B302B0A6DF25F14374FE1356D 6D51C245

E485B576625E7EC6F44C42E9A637ED6B0BFF5CB-6F406B7ED

E E 3 8 6 B F B 5 A 8 9 9 F A 5 A E 9 F 2 4 1 1 7 C 4 B - 1FE649286651ECE45B3D

C2007CB8A163BF0598DA48361C55D39A-69163FA8FD24CF5F

83655D23DCA3AD961C62F356208552BB9ED52907 7096966D

6 7 0 C 3 5 4 E 4 A B C 9 8 0 4 F 1 7 4 6 C 0 8 C A - 18217C32905E462E36CE3B

E39E772C180E86039B2783A2EC07A28FB-5C55DF06F4C52C9

D E 2 B C B F 6 9 5 5 8 1 7 1 8 3 9 9 5 4 9 7 C E A 9 5 6 A E - 515D2261898FA0510

15728E5A8AACAA68FFFFFFFFFFFFFFFF

Primitive root (generator) α: 2

References

Akdogan D, Altop DK, Eskandarian L, Levi A (2018) Secure key agreement protocols: Pure biometrics and cancelable biometrics. Comput Netw 142:33–48

Alvarez G, Li S (2006) Some basic cryptographic requirements for chaos-based cryptosystems. Int J Bifurc Chaos 16(08):2129–2151

Barman S, Samanta D, Chattopadhyay S (2015a) Approach to cryptographic key generation from fingerprint biometrics. Int J Biom 7(3):226–248

Barman S, Samanta D, Chattopadhyay S (2015b) Fingerprint-based crypto-biometric system for network security. EURASIP J Inf Secur 2015(1):3

Barman S, Chattopadhyay S, Samanta D, Panchal G (2017) A novel secure key-exchange protocol using biometrics of the sender and receiver. Comput Electr Eng 64:65–82

Barman S, Das AK, Samanta D, Chattopadhyay S, Rodrigues JJPC, Park Y (2018) Provably secure multi-server authentication protocol using fuzzy commitment. IEEE Access 6:38578–38594

Barman S, Shum HPH, Chattopadhyay S, Samanta D (2019) A secure authentication protocol for multi-server-based e-healthcare using a fuzzy commitment scheme. IEEE Access 7:12557–12574

Chen B, Chandran V (2007) Biometric based cryptographic key generation from faces. In: Proceedings of the 9th IEEE biennial conference of the Australian pattern recognition society on digital image computing techniques and applications, Washington, DC, USA, pp 394–401

Clancy TC, Kiyavash N, Lin DJ (2003) Secure smartcard based fingerprint authentication. In: Proceedings of the ACM SIGMM workshop on biometrics methods and applications, Berkeley, California, USA, pp 45–52

Feng H, Wah CC (2002) Private key generation from on-line handwritten signatures. Inf Manag Comput Secur 10(4):159–164

Freier A, Karlton P, Kocher P (2011) The secure sockets layer (SSL) protocol version 3.0. Request for comments: 6101

Hao F, Anderson R, Daugman J (2006) Combining crypto with biometrics effectively. IEEE Trans Comput 55(9):1081–1088

Jiang Q, Chen Z, Li B, Shen J, Yang L, Ma J (2018) Security analysis and improvement of bio-hashing based three-factor authentication scheme for telecare medical information systems. J Ambient Intell Humaniz Comput 9(4):1061–1073

Jin Z, Teoh ABJ, Ong TS, Tee C (2010) A revocable fingerprint template for security and privacy preserving. KSII Trans Int Inf Syst 4:1327–1342

Juels A, Sudan M (2006) A fuzzy vault scheme. Des Code Cryptogr 38(2):237–257

Juels A, Wattenberg M (1999) A fuzzy commitment scheme. In: Proceedings of the 6th ACM conference on computer and communications security, New York, NY, USA, pp 28–36

Kanade S, Camara D, Krichen E, Petrovska-Delacretaz D, Dorizzi B (2008) Three factor scheme for biometric-based cryptographic key regeneration using iris. In: 6th Biometrics Symposium. pp 59–64

Kanade S, Petrovska-Delacretaz D, Dorizzi B (2010) Generating and sharing biometrics based session keys for secure cryptographic applications. In: Fourth IEEE international conference on biometrics: theory, applications and systems (BTAS). pp 1–7

Kanade SG, Petrovska-Delacretaz D, Dorizzi B (2012) A novel crypto-biometric scheme for establishing secure communication sessions between two clients. In: Proceedings of the IEEE international conference of biometrics special interest group (BIOSIG), Darmstadt, Germany, pp 1–6

Kivinen T (2003) More modular exponential (MODP) Diffie–Hellman groups for internet key exchange (IKE). Request for Comments (RFC) 3526

Maio D, Maltoni D, Cappelli R, Wayman JL, Jain AK (2002) FVC2002: second fingerprint verification competition. In: 16th IEEE international conference on pattern recognition, vol 3, Quebec, Canada, pp 811–814

Maio D, Maltoni D, Cappelli R, Wayman JL, Jain AK (2004) FVC2004: third fingerprint verification competition. In: Zhang D, Jain AK (eds) International conference on biometric authentication. Lecture notes in computer science, vol 3072. Springer, Berlin, Heidelberg, pp 1–7

Masek L (2003) Recognition of human iris patterns for biometric identification. Tech. rep., Univ. of Western Australia

Monrose F, Reiter MK, Li Q, Wetzel S (2001) Cryptographic key generation from voice. In: Proceedings of IEEE symposium on security and privacy, Oakland, CA, USA, pp 202–213

Murillo-Escobar MA, Abundiz-Pérez F, Cruz-Hernández C, López-Gutiérrez RM (2014) A novel symmetric text encryption algorithm based on logistic map. In: Proceedings of the international conference on communications, signal processing and computers, pp 49–53

Murillo-Escobar M, Cruz-Hernández C, Abundiz-Pérez F, López-Gutiérrez R (2015) A robust embedded biometric authentication system based on fingerprint and chaotic encryption. Expert Syst Appl 42(21):8198–8211

Panchal G, Samanta D (2018) A novel approach to fingerprint biometric-based cryptographic key generation and its applications to storage security. Comput Electr Eng 69:461–478

Panchal G, Samanta D, Barman S (2017) Biometric-based cryptography for digital content protection without any key storage. Multimed Tools Appl. https://doi.org/10.1007/s11042-017-4528-x

Ratha NK, Chikkerur S, Connell JH, Bolle RM (2007) Generating cancelable fingerprint templates. IEEE Trans Pattern Anal Mach Intell 29(4):561–572

Rathgeb C, Uhl A (2011a) Context-based biometric key generation for iris. IET Comput Vis 5(6):389–397

Rathgeb C, Uhl A (2011b) A survey on biometric cryptosystems and cancelable biometrics. EURASIP J Inf Secur 2011(1):3

Reddy AG, Das AK, Odelu V, Ahmad A, Shin JS (2019) A privacy preserving three-factor authenticated key agreement protocol for client-server environment. J Ambient Intell Humaniz Comput 10(2):661–680

Soutar C, Roberge D, Stoianov A, Gilroy R, Bhagavatula V (1998) Method for secure key management using a biometric. WO Patent App. PCT/CA1998/000,362

Srinivas J, Mishra D, Mukhopadhyay S, Kumari S (2018) Provably secure biometric based authentication and key agreement protocol for wireless sensor networks. J Ambient Intell Humaniz Comput 9(4):875–895

Stallings W (2006) Cryptography and network security: principles and practices. Pearson Education, Chennai

Uludag U, Pankanti S, Prabhakar S, Jain AK (2004) Biometric cryptosystems: issues and challenges. Proc IEEE 92(6):948–960

Verifinger S (2010) VeriFinger SDK, fingerprint recognition technology, Neuro technology

Watson CI, Wilson C (1992) NIST special database 4. In: Fingerprint database. National Institute of Standards and Technology, pp 1–14