Information Sciences 546 (2021) 481–511

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

BIOFUSE: A framework for multi-biometric fusion on biocryptosystem level

Donghoon Chang [a], Surabhi Garg [a,*], Mohona Ghosh [b], Munawar Hasan [a]

[a] IIIT-Delhi, India
[b] Indira Gandhi Delhi Technical University for Women, India

ARTICLE INFO

Article history:
Received 31 August 2019
Received in revised form 17 August 2020
Accepted 18 August 2020
Available online 2 September 2020

Keywords:
Biometric cryptosystem
Biometric template protection
Multi-biometric fusion
Fuzzy commitment
Fuzzy vault
Format-preserving encryption

ABSTRACT

Biometric cryptosystems or biocryptoystems are gaining prominence for cryptographic key generation, encryption and biometric template protection. However, the most popular state-of-the-art biocryptosystems- fuzzy commitment and fuzzy vault are prone to multiple security attacks. Recently proposed multi-biometric cryptosystems improve security and enhance recognition performance. They perform the fusion of multi-biometric characteristics with either a single biocryptosystem or independently accessed, multiple biocryptosystems. An attack on any of the involved biocryptosystems can weaken the security of the whole system. In our paper, we propose a multi-biometric fusion framework- BIOFUSE, that combines fuzzy commitment and fuzzy vault using the format-preserving encryption scheme. BIOFUSE makes it improbable for an attacker to get unauthorized access to the system without impersonation of all the biometric inputs of the genuine user at the same instant. We present 4 most basic ways of constructing BIOFUSE and found only 1 named S-BIOFUSE (S_3) as a secure design. We compare the recognition performance of the proposed scheme with existing multi-biometric cryptosystems on various databases. The results show 0.98 true match rate at 0.01 false match rate on a virtual IITD-DB1 database that indicates that our proposed work achieves significantly good recognition performance while providing high security.

© 2020 Elsevier Inc. All rights reserved.

1. Introduction

The traditional means of user authentication, such as passwords and hardware tokens, have serious security concerns that prompt the widespread implementation of biometric authentication systems. Biometric authentication [21] refers to uniquely recognizing a person based on their physiological or behavioural characteristics. The biometric data, also known as a biometric template, comprises of unique features extracted from the biometric characteristics of an individual using sensors which captures biometric data in the form of a digital image. For example, a fingerprint template consists of ridge endings and ridge bifurcation points that represent minutiae points [29]. Similarly, an iris template is a binary string generated from the patterns found in iris texture [9]. Typically, the biometric templates are stored on servers in their original, unprotected form from where they can be stolen or modified by an attacker [35]. Since biometric data is irreplaceable, therefore once lost, it cannot be changed, unlike passwords. The European Union (EU) General Data Protection Regulation 2016/679 [44] has classified the biometric data as sensitive information, and thus it is contingent to the right to preserve privacy. These

* Corresponding author.
E-mail address: surabhig@iiitd.ac.in (S. Garg).

https://doi.org/10.1016/j.ins.2020.08.065
0020-0255/© 2020 Elsevier Inc. All rights reserved.

concerns prompt the need to protect biometric templates stored on the server. Recently, proposed biometric template protection schemes [35,45,36] transform the original biometric template into the protected template. These are generally classified into three types: homomorphic encryption-based schemes, cancelable biometrics and biocryptosystems. The homomorphic encryption-based schemes [15] encrypt the original template and the comparisons during authentication are performed on the encrypted template. These schemes provide high security but take substantial computational time. The cancelable biometric approaches [41,14] map the original biometric template into a protected, cancelable template using a secret parameter such as a key or a password. The comparisons between the two protected templates take place in the transformed domain. Bit errors are introduced in the protected template due to transformation, which degrades the recognition performance of the system.

In the biocryptosystems [50], during the enrolment phase, the original biometric template is transformed into biometric-dependent secure information known as helper data. A cryptographic key is bound with the biometric template or is derived from the biometric template. During the authentication phase, the helper data helps to recover the cryptographic key without revealing any information about the original biometric template. The key helps to perform successful user authentication. Fuzzy commitment [23] and fuzzy vault [22] are the two popular examples of biocryptosystems. The overview of these two schemes is shown in Fig. 1. The helper data in both fuzzy commitment and fuzzy vault schemes is public and is stored on database server during the enrolment phase. The detailed working of fuzzy commitment and fuzzy vault scheme is given in Section 2.

Biocryptosystems have been proved to be efficient in protecting biometric templates [35,19,27]. However, the fuzzy commitment and fuzzy vault schemes are prone to multiple security attacks, as discussed in literature [24,16,38]. The fuzzy commitment scheme uses random error correcting codewords. An attack has been proposed [46,48] on these codewords where if a few bits of a codeword or biometric template is revealed, the whole biometric template can be revealed to the attacker. The fuzzy vault scheme is vulnerable to brute force attack on the public vault [31,34]. Additionally, it suffers from other security attacks [38] such as correlation attack, blend substitution attack, key inversion attack etc. These are discussed in detail in Section 6.

Lately, multi-biometric biocryptosystem based authentication systems have been proposed in [42,33,27,32], where multiple biometric templates are given as inputs to the biocryptosystem(s). These schemes require combination or fusion of multiple biometric characteristics, which is generally done either on the feature level, score level or decision level. Multi-biometric biocryptosystems improve the accuracy during authentication [37] as well as they improve resilience to spoof attacks since it is difficult to spoof multiple biometric characteristics of the user simultaneously.

Motivation. The existing multi-biometric biocryptosystems are implemented in two ways. A single biocryptosystem is implemented where multi-biometric characteristics are given as inputs with an existing fusion approach [32]. Such systems are prone to several existing drawbacks of the state-of-the-art biocryptosystems, i.e. fuzzy commitment and fuzzy vault. Another approach is the implementation of multiple biocryptosystems [8], in which one or more biometric characteristic is given as input(s) to each biocryptosystem involved. Each of these biocryptosystems can be accessed independently of each other. The major disadvantage, in the latter case, is that even if one of the underlying biocryptosystems is compromised by an attacker, the security of whole system breaks. In other words, an attacker can attack both (in case if 2 biocryptosystems are involved) the biocryptosystems independent of each other. In the worst case, when the attacker does not have any information about the input biometric templates, the ideal security bound of a system with two biocryptosystems involved is computed in terms of the number of trials needed to perform brute force attack. Given, K_1 and K_2 as the respective security parameters of the two underlying biocryptosystems, the ideal or desired security bound in terms of the number of trials is equal to $2^{|K_1|} \times 2^{|K_2|}$ (or $2^{|K_1|+|K_2|}$). In the latter approach, the exhaustive search attack (in the worst case) on the security parameters- K_1 and K_2, leads to overall security bound equal to $2^{|K_1|} + 2^{|K_2|}$ or $2^{min(|K_1|,|K_2|)}$ which is much lower than the desired security bound. Further limitations are stated in Section 3. The drawbacks of the existing multi-biometric biocryptosystems and the requirements to achieve high-security bounds motivate us to propose a biocryptosystem level fusion framework known as BIOFUSE. It combines multiple biometric characteristics at the biocryptosystem level such that the overall security is equivalent to the combined security of underlying biocryptosystems- fuzzy commitment and fuzzy vault. Even if an attacker attacks one of the underlying biocryptosystem, it is not possible to get unauthorized access to the system without access to all the underlying biocryptosystems in the system.

To design such architecture, we perform a thorough security analysis of the most basic architectures that are possible with the fusion of multi-biometrics at a biocryptosystem level (considering two biocryptosystems in a system). Biocryptosystem level denotes that the fusion of biometric characteristics is done by combining the two biocryptosystems such that accessing any of them independent of others is not a feasible option. We introduce a format-preserving encryption (FPE) scheme [12] for combining the two biocryptosystems that provides an extra layer of security to the proposed system. FPE preserves the format of the plaintext and the ciphertext. We discuss the rationale behind the use of FPE scheme in Section 6. Further, the implementation details are provided in Section 7 with an example shown in B.

Our paper makes the following contributions.

1. A multi-biometric fusion framework known as BIOFUSE is proposed where the multi-biometric characteristics are combined by fusion of two popular biocryptosystems at the biocryptosystem level. We propose, best to the authors' knowledge, a novel integration of two biocryptosystems using a format-preserving encryption scheme.

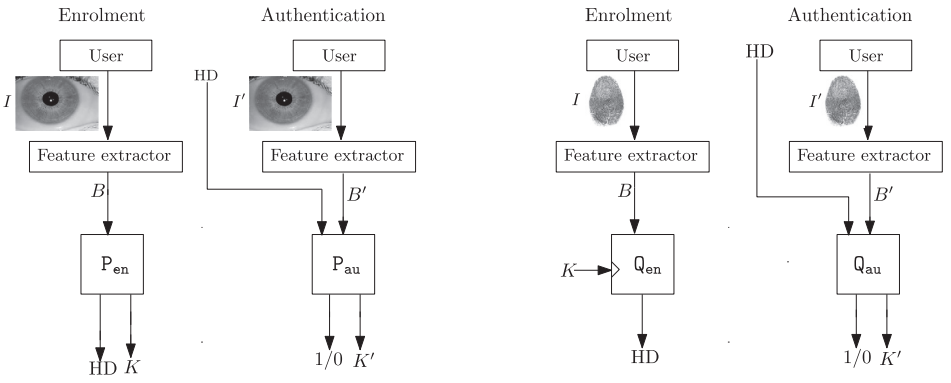


Fig. 1. General overview of enrolment and authentication phases in biocryptosystems - (a) A module $\mathtt{P}_{\mathtt{en}}$ of FC scheme takes biometric template B extracted from biometric characteristics I. It generates a secret, cryptographic key K and a helper data HD. During authentication using another sample B' and HD, key K' is generated from module $\mathtt{P}_{\mathtt{au}}$. (b) Module $\mathtt{Q}_{\mathtt{en}}$ of FV scheme takes biometric template B and a system's generated key K to give helper data HD. The helper data helps to recover K' from B' using module $\mathtt{Q}_{\mathtt{au}}$. The internal workings of these modules are shown in Figs. 2 and 3.

2. We perform a thorough security analysis of all the possible combinations in which biometric fusion is performed, taking fuzzy commitment and fuzzy vault schemes into consideration. Table 1 summarizes the security of all the 4 possible constructions in both offline and online attack modes. We denote the most secure design among these as S-BIOFUSE (S_3).
3. As shown in Table 1, the best attack possible on the proposed framework (particularly, S_3) is the time-memory tradeoff attack in the offline mode with security bound equals to $2^{min(|K_1|,p)+|K_1|} + 2^{min(|K_2|,q)}$ which approximates to $2^{2\times|K_1|}$ or $2^{|K_1|} \times 2^{|K_1|}$, assuming the underlying biocryptosystems are secure.
4. No additional security parameter- a key or a password is used in our work except those required for the state-of-the-art biocryptosystem shown in Fig. 1.
5. Further, our proposed framework is secure against various attacks discussed in Section 6 proposed in the literature on fuzzy vault, including the brute force attack, key inversion attack, correlation attack, and blend substitution attack.
6. We provide a detailed analysis on recognition performance of the proposed system along with the comparisons with existing multi-biometric cryptosystems. The results are shown in Table 5, which emphasize the scope of deployment of our proposed architecture in real-time scenarios.

Organization. The rest of the paper is organized as follows. Section 2 provides background on various definitions. In Section 3, we provide a literature review of the existing works related to multi-biometric biocryptosystems. Section 4 describes the attack model and security notion. It is followed by Section 5 where we describe details and security of all possible combinations in which BIOFUSE can be constructed. Section 6 provides the detailed construction of S-BIOFUSE along with its security analysis. The experiments and results are given in Section 7. Finally, the conclusion and future work are summarized in Section 8. Appendix A provides a summary of all the possible case of combining the fuzzy commitment and fuzzy vault schemes. B provides an example showing the implementation of S-BIOFUSE.

2. Preliminaries

We discuss some fundamental concepts that are used throughout the paper in the following subsections.

2.1. Notations

Table 2 shows some important notations that are used in the paper.

2.2. Fuzzy commitment scheme

It was introduced by Juels and Wattenberg in 1999 [23]. Given a biometric data B in the form of a binary string, the Hamming distance between two biometric data strings B and B' is defined as the number of bit positions in which the two strings differ and is denoted as $d(B, B')$. A fuzzy commitment (FC) is a pair of two functions- commitment function, and a decommitment function with the following properties:

Table 1
Summary of security analysis in offline and online mode for all the possible constructions (denoted by S_i) formed by the combination of two biocryptosystems-fuzzy commitment (FC) and fuzzy vault (FV). Security bound is given in terms of the number of trials performed for brute force attack on the system. K_1 and K_2 are the secret keys, and C denotes the error correcting codeword used in fuzzy commitment scheme. H refers to the hash function. Bar^{-} indicates that a particular biometric template B_1 or B_2 is not known to the attacker. p and q refers to the internal security (measured in bits) of FC and FV respectively. Refer Section 5 for the details.

Modes		Cases $S_i, (1 \leqslant i \leqslant 4)$			
		FC-then-FC (S_1)	FV-then-FV (S_2)	FC-then-FV (S_3)	FV-then-FC (S_4)
Offline	$\bar{B_1}, B_2$	$2^{min(\lvert K_1\rvert,p)}$	$2^{min(\lvert K_1\rvert,q)}$	$2^{min(\lvert K_1\rvert,p)}$	$2^{min(\lvert K_1\rvert,q)}$
	$B_1, \bar{B_2}$	$2^{min(\lvert H(C_2)\rvert,p)}$	$2^{min(\lvert K_2\rvert,q)}$	$2^{min(\lvert K_2\rvert,q)}$	$2^{min(\lvert H(C_2)\rvert,p)}$
	$\bar{B_1}, \bar{B_2}$[1]	$2^{min(\lvert K_1\rvert,p)+\lvert K_1\rvert}+$ $2^{min(\lvert H(C_2)\rvert,p)}$	$2^{min(\lvert K_1\rvert,q)+\lvert K_1\rvert}+$ $2^{min(\lvert K_2\rvert,q)}$	$2^{min(\lvert K_1\rvert,p)+\lvert K_1\rvert}+$ $2^{min(\lvert K_2\rvert,q)}$	$2^{min(\lvert K_1\rvert,q)+\lvert K_1\rvert}+$ $2^{min(\lvert H(C_2)\rvert,p)}$
Online	$\bar{B_1}, B_2$	$2^{\lvert H(B_1 \Vert rand_1)\rvert}$	$2^{\lvert K_1\rvert}$	$2^{\lvert H(B_1 \Vert rand_1)\rvert}$	$2^{\lvert K_1\rvert}$
	$B_1, \bar{B_2}$	$2^{\lvert H(C_2)\rvert}$	$2^{\lvert K_2\rvert}$	$2^{\lvert K_2\rvert}$	$2^{\lvert H(C_2\rvert}$
	$\bar{B_1}, \bar{B_2}$	$2^{\lvert H(C_2)\rvert}$	$2^{\lvert H(K_2)\rvert}$	$2^{\lvert H(K_2)\rvert}$	$2^{\lvert H(C_2\rvert}$

[1] Time-memory trade-off attack complexity (explained in Section 5)

Table 2
Some other notations.

Notation	Description
I	Biometric characteristics given by user as a digital image
B	Biometric features representing the biometric template, where B' denotes another instance of B
$rand$	A pseudo-random number
C	Error correcting codeword
H	A cryptographic hash function
E	A symmetric key encryption scheme using block cipher modes of operation
V	A fuzzy vault with genuine and chaff points combined
K	A cryptographic key
FC	fuzzy commitment scheme
FV	fuzzy vault scheme
FFE	a format-preserving encryption module
FFD	a format-preserving decryption module
$\mathtt{P}_{\mathtt{en}}$	an instantiate of FC enrolment module
$\mathtt{P}_{\mathtt{au}}$	an instantiate of FC authentication module
$\mathtt{Q}_{\mathtt{en}}$	an instantiate of FV enrolment module
$\mathtt{Q}_{\mathtt{au}}$	an instantiate of FV authentication module
HD	helper data

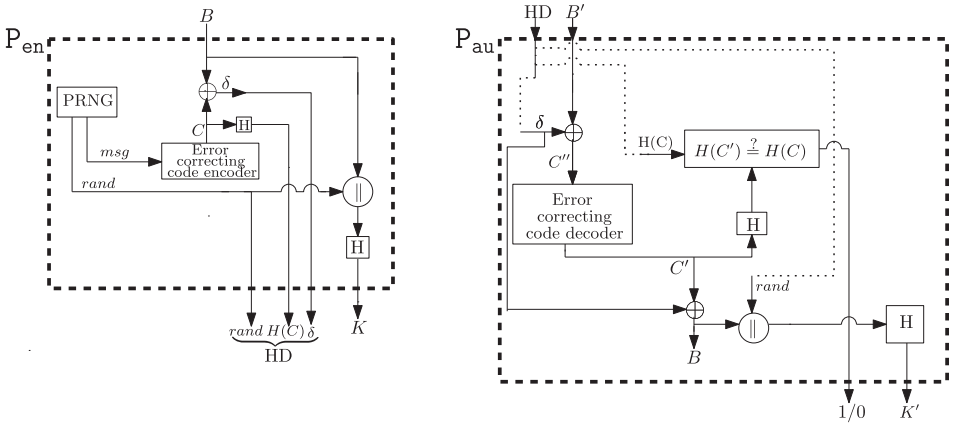
1. A commitment function P_{en} on input $B \in \{0,1\}^n$, selects a random error correcting code $C \in \{0,1\}^n$ generated from a random message msg and returns the secure sketch value denoted as $\delta \in \{0,1\}^n$ such that $\delta = B \oplus C$. The helper data HD constitutes $(rand, H(C), \delta)$, where $rand$ is a random message used to generate cryptographic key from B.
2. The de-commitment function P_{au} takes a n-bit query template B' and the helper data HD. It computes $(C'' \oplus (B \oplus B'))$. Provided an efficient decoding of error correcting codewords, if $d(B_1, B_1') \leqslant t$, where t is the maximum number of errors that can be corrected in the bit string, C'' is decoded to a nearest codeword C'. If $(H(C') = H(C))$, original B is recovered.

One of the applications of the fuzzy commitment scheme is to generate a secure cryptographic key, denoted as K of length k. Dodis et al. [10] give a basic key generation scheme where the key is generated by taking the hash of the original biometric template B concatenated with a random string $rand$. The key is recovered during authentication if the biometric sample matches with the sample provided during enrolment. The mechanism is discussed in Section 6. Even though the committed value C is not directly treated as a cryptographic key, nonetheless it should be selected from a large space of codewords to prevent any attack resulting in the disclosure of original biometric data B. The block diagram showing key generation during enrolment and key recovery during authentication is shown in Fig. 2.

2.3. Fuzzy vault scheme (set difference metric)

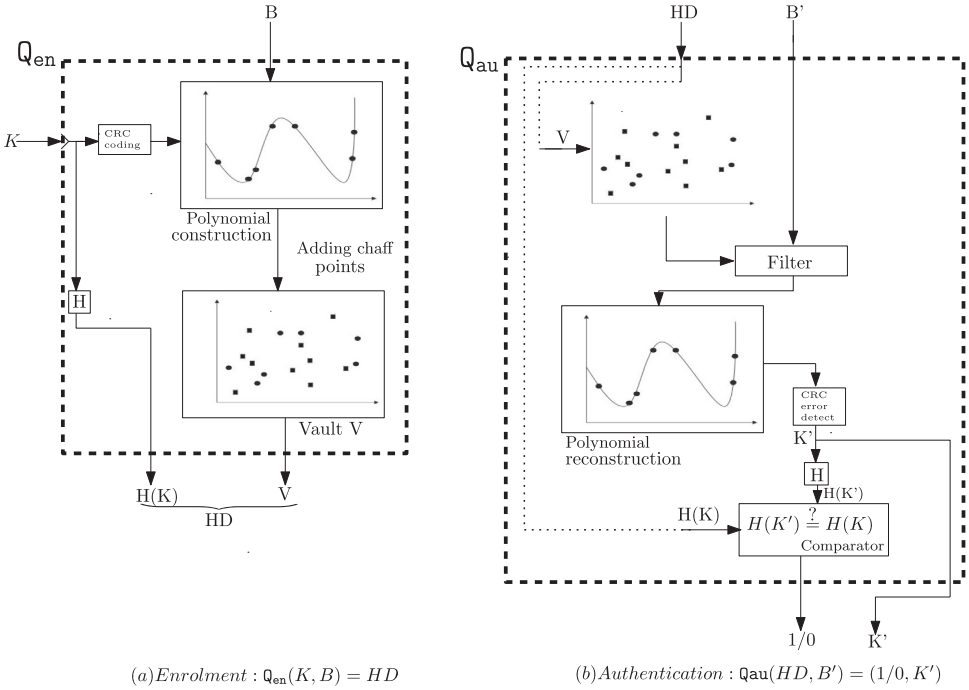
The fuzzy vault introduced by Juels and Sudan in 2002 [22] is generated by encoding the biometric features on a polynomial using the key as coefficients of a polynomial. If a similar biometric feature set of the same user overlaps with the original set, the key is decoded using the vault. The block diagram is shown in Fig. 3.

Let a biometric template B be denoted as an unordered set of l well separated features: $B = \{B_1, B_2, \ldots, B_l\}$. A secret key K is splitted into n parts given as $K = K^0 K^1 K^2 \ldots K^{n-1}$ satisfying $n \leqslant l$. A CRC code is appended to secret key to get a new secret



$(a) Enrolment : \mathtt{P_{en}}(B) = (HD, K)$ $(b) Authentication : \mathtt{P_{au}}(HD, B') = (1/0, K')$

Fig. 2. Enrolment and authentication phase of the fuzzy commitment scheme. The enrolment is shown by module $\mathtt{P_{en}}$ that takes input B and a random codeword C to generate a secure K and helper data HD (shown with dotted line). $\mathtt{P_{au}}$ recovers key with the help of helper data and biometric template B'.



$(a) Enrolment : \mathtt{Q_{en}}(K, B) = HD$ $(b) Authentication : \mathtt{Qau}(HD, B') = (1/0, K')$

Fig. 3. Enrolment and authentication phase of fuzzy vault scheme. The enrolment phase is shown by module $\mathtt{Q_{en}}$ that takes input B and a system's provided secret key K to generate a public vault V. The authentication phase is shown by $\mathtt{Q_{au}}$ which recovers key with the help of vault and biometric template B'.

having $(n+1)$ parts. Using the secret key as coefficients of polynomial denoted by $p(B_i) = K^n B_i^n + K^{n-1} B_i^{n-1} + K^{n-2} B_i^{n-2} + \cdots K^1 B_i^1 + K^0$ for $1 \leqslant i \leqslant l$, a genuine set $G = \{(B_i, p(B_i))\}_{i=1}^{l}$ is generated from each feature in the feature set B. To hide the genuine points from an attacker, random chaff points are added to the vault. A chaff point set $Ch = \{(z_i, u_i)\}_{i=l+1}^{r}$ is generated such that $z_i \notin \{B_i\}$ and $u_i \notin \{p(B_i)\}$. The genuine point set and chaff point set constitute a vault with total r points denoted as $V = G \cup Ch$. During authentication, if a sufficient number of points (candidates) in the query template overlaps with the biometric feature points in the vault, the polynomial of degree n is reconstructed, and the secret key K' is recovered. The CRC error detection is applied [34] to K' and the authentication step is repeated for a new set of candidates in case if error is detected. If no error is detected, it indicates that $K = K'$ with very high probability.

3. Related work

Several multi-biometric biocryptosystems have been analyzed in [20,37] for their use in biometric authentication systems. We describe some of the popular multi-biometric template protection schemes based on biocryptosystems along with their limitations.

3.1. Existing multi-biometric fusion schemes

As stated in ISO/IEC TR 24722 on multimodal and other multi-biometric fusion [1], the fusion between different biometric characteristics can be performed at three levels- feature level, score level and decision level fusion.

In feature-level fusion based biometric system, given the biometric templates extracted from different biometric characteristics, a single template of higher dimensions is generated with more discriminative information as compared to the individual templates. The main challenge is to combine features from different biometric characteristics with different representations. If the features are incompatible or heterogeneous (by distance, similarity), several embedding algorithms are proposed [32] that help in constructing a combined feature vector. Further, the embedding algorithms can be used in score-level or decision-level fusion approaches. Nandakumar et al. [33] proposed a feature level fusion scheme that derives a single multi-biometric feature set from the iris and the fingerprint templates and secure these features using the fuzzy vault approach. They showed that a vault constructed from multi-biometric feature set has more security and provides better recognition performance than the vault created by a feature set obtained from a single biometric characteristic. Kanade et al. [25] utilized face and iris templates to generate a single feature set and apply the fuzzy commitment scheme on it to obtain a high entropy cryptographic key. Nagar et al. [32] provided a practical implementation of feature-level fusion framework for both fuzzy vault and fuzzy commitment schemes that simultaneously protects the multiple templates of a user using a single secure sketch. Feature level fusion using multiple characteristics of a user proves to be significant in providing high privacy as compared to the single characteristics biometric systems since only the fused feature vector is stored on the server database. Further, it requires less storage since only the combined feature vector is stored in the database server. However, it requires additional feature extraction and transformation tools for the heterogeneous features (variable formats based on distance, similarity, etc.).

In the score level fusion, the individual similarity scores obtained from each unimodal systems are normalized and combined to obtain a reliable and accurate score. A brief overview of various ways to achieve score level fusion is given in [11,17] where a two-level score level fusion approach is proposed to integrate the scores obtained from cancelable templates derived from different biometric characteristics. The authors proposed mean-closure based weighting and rectangular area based weighting technique to obtain the overall fused score. A similar approach can be used for biocryptosystems. Score level fusion provides high recognition performance, but, it usually provides unpredictable performance as comparison scores of different characteristics may follow a different probability distribution. Another fusion framework is decision level fusion, where the individual accepts/rejects decisions from each unimodal systems are combined to get the final decision. The non–homogeneous features can be used without transformation and can be easily compared with the existing comparators. A decision level fusion is performed on fingerprint-based multi-biometric biocryptosystem [27]. Hash functions are used to protect each fingerprint. The system provides authentication if a certain number of fingerprints out of the total query fingerprints satisfies the threshold criteria. However, use of an extra secret key at the second level adds an extra parameter to the overall system which needs to be secured.

3.2. Other multi-biometric schemes involving biocryptosystems

A modular approach is proposed in [8,6] to construct a multi-biometric biocryptosystem where the output of one template protection scheme is fed into another scheme/module that provides the final output. In [6], authors, proposed cancelable secure sketch where the secure sketch is applied on cancelable biometric templates. The secure sketch provides biometric template protection, and cancelable biometric template prevents correlation attack and provides renewability property. Cimato et al. [8] proposed a modular approach where two secure sketch schemes are taken as independent modules to secure multi-biometric templates. Given two biometric templates, a secure value is constructed from the first template. The key is generated from the first biometric template as input. The key is then XORed with the second biometric template to generate helper data which is stored along with secure value on the server. The above approach is generalized by Fang et al. [13], where multiple biometric templates are combined in a cascaded manner while deploying the secure sketch framework [10] within the fuzzy commitment scheme. The advantage of the modular approach is that it easily allows the addition of biometric characteristics along with the heterogeneous templates. The limitation is that the overall security in the modular-level fusion scheme is defined by the security of secure sketch in the outermost module. A multi-biometrics cancelable approach has been proposed in [7] where a fuzzy extractor is combined with a bit-wise encryption scheme to generate a cancelable template as output. The scheme provides high security based on the assumption that getting access to one biometric template by an attacker is equivalent to getting both biometric templates of that user. The scheme provides high-performance rates for any binarized data. However, it would require pre-transformations for non-binarized input data such as fingerprint template.

4. Models and settings

We present the system's entities and possible attack scenarios.

4.1. System model and participants

The proposed system consists of users and a server. Two biocryptosystems are involved in the biometric system. The user can be an attacker who tries to get successful authentication by combining the credentials of a genuine user of the system. The user provides two biometric characteristics, iris/face and a fingerprint, one for each biocryptosystem. The helper data generated as outputs from the two biocryptosystems is stored corresponding to each user ID on the database present on the server.

4.2. Attack model

Our proposed scheme is based on the client–server setup. The end-user provides biometric data at the client-side. The storage of the data used for authentication is done on the server. Hence, the server is also responsible for user authentication.

We define the security of all the possible constructions in terms of attack complexity in online and offline modes. In the **online mode**, the attacker tries to authenticate itself in the real-time scenario by impersonating the biometric templates of a genuine user. On providing inputs as one or both biometric templates B_1 and B_2, the attacker gets the final output as 0 that indicates an authentication failure, or 1 that indicates successful authentication. In the **offline mode**, the goal of an attacker is to extract some secret information such as biometric templates, secret key, etc. from the simulated system. The attacker provides one or more known inputs- such as a key or a biometric data and tries to get the desired information. The extracted information is used in the online attack mode to perform successful authentication.

4.3. Security notion

In the fuzzy commitment scheme shown in Fig. 2, authentication is said to be successful when the hash of error correcting codeword $H(C')$ generated during the authentication is the same as the hash of codeword $H(C)$ stored during enrolment. If the size of the random message *msg* used to generate the codeword is small, i.e. $|msg| < 128$ bits, it would be easy to guess the codeword. Once the codeword is leaked, the original biometric template could be easily revealed which breaks the security of the fuzzy commitment scheme. We denote it as the internal security of the fuzzy commitment scheme, and it is given by p bits.

Similarly, in the fuzzy vault scheme shown in Fig. 3, the secret parameter is a system-generated key K that binds with the input biometric template B to generate a public vault V. Given a vault V constructed with a polynomial of degree n, the attacker can perform brute force to the vault by applying polynomial decoding algorithm on every combination of $(n+1)$ vault points to find out the one which gives the correct secret key K. It is calculated [31,34] as number trials (in bits) to filter $(n+1)$ genuine points out of all the r points in the vault. It is given as

$$= \log_2\left(\binom{r}{n+1} / \binom{G}{n+1}\right)$$

where G is the number of genuine points in the vault. The knowledge of $(n+1)$ genuine points could reveal the secret key K to the attacker. Generally, the number of trials used to get the secret key are less [34], which makes fuzzy vault insecure. We denote it as the internal security of the fuzzy vault scheme, and it is given by q bits. For the rest of the paper, we consider the internal security of the fuzzy commitment scheme as p bits and fuzzy vault scheme as q bits.

We made a few assumptions to analyze the security of BIOFUSE, which is described in Section 5.

The assumptions are:

- It is not possible for the attackers to get biometric characteristics using brute force attack with a probability of $1/2^{|B|}$, since the size of the input biometric template B, is generally large [34,14].
- The size of error correcting codeword is sufficiently larger than the key generated using a fuzzy commitment scheme, i.e. $|C| \gg |K|$. Further, error correcting codewords need to be of the same size as of the size of biometric templates.
- The length of both the keys K_1 and K_2 is $\geqslant 128$ bits to ensure the security of the overall scheme. Note that, K_1 and K_2 are the instances of the two keys which could belong to any of the fuzzy commitment or fuzzy vault scheme.
- The helper data denoted as *HD*, whether encrypted or unencrypted is stored on the database server as public data without any password protection.

5. Proposed work

Given the inputs as multi-biometric characteristics, we provide all the possible combinations that can be constructed from combining two biocryptosystems- fuzzy commitment scheme and fuzzy vault scheme to construct a

biocryptosystem-level fusion framework. Similar to a general biometric authentication system, BIOFUSE consists of two phases:

- **Enrolment phase** takes multi-biometric characteristics as input for biocryptosystems and generates secure, public values, i.e. helper data, that are stored on the server.
- **Authentication phase** authenticates the user only if both the biometric characteristics provided during authentication are from the genuine user.

We apply an exhaustive search to explore all the possible ways in which we can combine the two biocryptosystems-fuzzy vault and fuzzy commitment. We found a total of 84 cases. We describe the most basic way of combining two schemes ignoring a few combinations due to several limitations. Refer Appendix A for details.

We found that the most feasible way of combining two biocryptosystems is only by encrypting the helper data of one biocryptosystem with the help of a key derived from the other biocryptosystem. Without decryption of the helper data using the same key, authentication cannot be performed. The combinations or cases that are possible with two biocryptosystems-fuzzy commitment (FC) and fuzzy vault (FV) schemes can be categorized into 4 constructions:

1. Two fuzzy commitment schemes can be combined, denoted as FC-then-FC,
2. Two fuzzy vault schemes can be combined, denoted as FV-then-FV,
3. A fuzzy commitment scheme can be combined with FV scheme, denoted as FC-then-FV
4. A fuzzy vault scheme can be combined with FC scheme, denoted as FV-then-FC.

Security analysis. For each of the 4 constructions mentioned, we also evaluate the security of the constructed scheme. We analyze two modes -

- **Online mode:** The attacker interacts with the deployed biometric system. It provides biometric inputs as B_1 and/or B_2 and gets output as 1/0 that denotes match or no match respectively.
- **Offline mode:** In the offline mode, instead of interacting with the deployed biometric system, the attacker implement the underlying algorithms and functions of the deployed system in its simulated system, with several parameters. The attacker provides some known values such as a key or a biometric data to retrieve unknown secret data from the system. The secrets could be used in the online attack mode.

We further consider three threat models:

1. Attacker knows B_2 but B_1 is not known.
2. Attacker knows B_1 but B_2 is not known.
3. Both B_1 and B_2 are unknown to the attacker.

Thus, we do our analysis for all the 4 constructions of BIOFUSE under 2 modes of attack, each with 3 threat models.

In the next subsections, we describe each of the 4 constructions in detail. Each construction is denoted as S_i, where $1 \leqslant i \leqslant 4$. For each construction, we first describe its working followed by its security analysis.

5.1. FC-then-FC: S_1

Construction. FC-then-FC is constructed by combining two fuzzy commitment schemes, each taking a different instance of biometric template as input. Fig. 4 shows the enrolment and authentication phases.

Enrolment Phase: During the enrolment phase, $\mathtt{P}_{\mathtt{en}_1}$ generates a cryptographic key K_1 from the input biometric template B_1. It also generates a public, helper data denoted as HD_1. From the second input biometric data B_2, another helper data HD_2 is generated with the help of $\mathtt{P}_{\mathtt{en}_2}$ module. The key K_1 encrypts the helper data HD_2 to give an encrypted helper data HD_2^* using format-preserving encryption scheme *FFE*. The FFE helps in preserving the format of ciphertext and plaintext. The role of FFE is discussed later in Section 6. For similar reasons, we use FFE scheme in all our other three constructions as well. The key K_2 does not play any role in the enrolment phase. The HD_1 and HD_2^* are stored on the server as public values.

Authentication phase: During the authentication phase, $\mathtt{P}_{\mathtt{au}_1}$ helps to recover the key K_1'. If the biometric template B_1' given during authentication is similar to B_1, i.e. $d(B_1, B_1') < t$, where t is a pre-defined threshold and d is the hamming distance between two binary strings, then K_1' is a correct key (i.e. equal to K_1). The *FFD* takes the key and encrypted helper data as inputs and decrypts the helper data denoted as HD_2'. With the help of second biometric template B_2' and helper data HD_2', hash verification is performed inside the $\mathtt{P}_{\mathtt{au}_2}$ module as

$$H(C_2') = (H(C_2))' \tag{1}$$

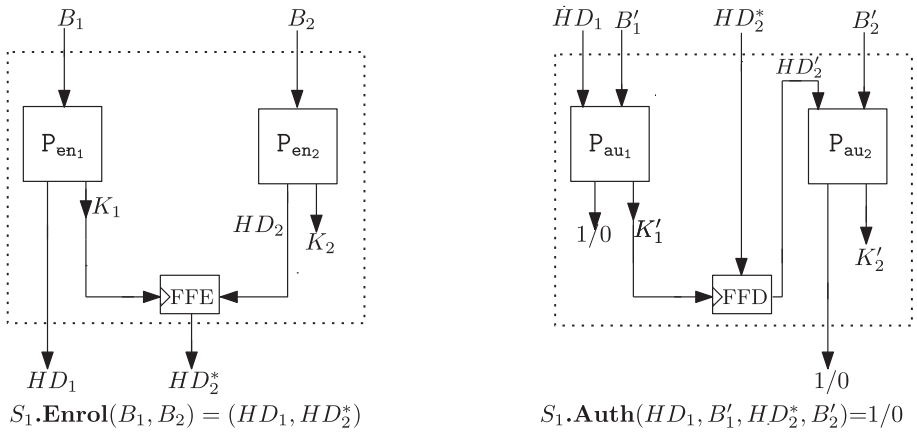


Fig. 4. Enrolment and authentication phase for the case: FC-then-FC. $\mathtt{P}_{\mathtt{en}_1}$ and $\mathtt{P}_{\mathtt{en}_2}$ represent the two instantiates of enrolment modules for fuzzy commitment scheme. $\mathtt{P}_{\mathtt{au}_1}$ and $\mathtt{P}_{\mathtt{au}_2}$ represent the two instantiates of authentication modules for fuzzy commitment scheme. *FFE* and *FFD* denotes the format-preserving encryption and decryption respectively. Refer Figs. 2 and 3 for internal workings of these modules. For other notations, refer Table 2.

where $H(C_2')$ represents the hash of codeword C_2' generated internally from P_{au_2} (second fuzzy commitment scheme) in the authentication phase. $(H(C_2))'$ represents the hash of the codeword C_2, which is stored as a part of helper data HD_2 during the enrolment in the encrypted form. If the condition is satisfied, the authentication is successful with output 1 and is shown as

$$S_1.Auth(HD_1, B_1', HD_2^*, B_2') \longrightarrow 1 \tag{2}$$

K_2' generated as output does not play any role in the authentication.

Security Analysis. We discuss the security analysis of S_1 in 2 different modes: online and offline mode as follows:

CASE A. (Offline mode setup). The attacker could know some of the secret parameters including the key(s), B_1 or B_2 as inputs. Given the public parameters as

$HD_1 = (H(C_1), rand_1, \delta_1)$ and

an encrypted helper data, HD_2^* such that

$FFE(K_1, HD_2) \longrightarrow HD_2^*$ where $HD_2 = (H(C_2), rand_2, \delta_2)$.

the attacker can perform several functions to generate the corresponding outputs as,

$S_1.Enrol(B_1, B_2) \longrightarrow (HD_1, HD_2^*)$ where

$\mathtt{P}_{\mathtt{en}_1}(B_1) \longrightarrow (HD_1, K_1)$,

$\mathtt{P}_{\mathtt{en}_2}(B_2) \longrightarrow (HD_2, K_2)$,

$\mathtt{P}_{\mathtt{au}_1}(HD_1, B_1') \longrightarrow (1/0, K_1')$,

$FFD(K_1', HD_2^*) \longrightarrow HD_2'$,

$\mathtt{P}_{\mathtt{au}_2}(HD_2', B_2') \longrightarrow (1/0, K_2')$.

The attacker can individually run any or all the above-mentioned functions or algorithms such as *FFE*, $\mathtt{P}_{\mathtt{en}_1}$, $\mathtt{P}_{\mathtt{en}_2}$, $\mathtt{P}_{\mathtt{au}_1}$, $\mathtt{P}_{\mathtt{au}_2}$ and *FFD* to get the respective outputs.

Under the offline mode setup, we analyse our first construction under three threat models as follows:

1. **Construction:** S_1
   Mode: Offline
   Threat Model: As attacker knows B_2' where $d(B_2, B_2') < t$, the aim is to recover B_1' or K_1' such that the authentication is successful.
   Since, B_1' is unknown, correct key $K_1' = K_1$ is unknown. The attacker can try to guess K_1' and perform the function *FFD* to get HD_2' given as,

$$FFD(K_1', HD_2^*) \longrightarrow HD_2' \tag{3}$$

$HD_2' = ((H(C_2))', rand_2', \delta_2')$. Further, the biometric template B_2' is known to the attacker, therefore, using the correct B_2' and all the values of δ_2' derived from the above equation, the attacker can perform $C_2'' = B_2' \oplus \delta_2'$ where C_2'' is decoded using error correcting codeword to obtain C_2'. From C_2', the attacker can then compute $H(C_2')$. Given $(H(C_2))'$, it can check the condition given in (1). If the condition is satisfied, the authentication is successful for the corresponding guessed key K_1', else a new value of K_1' is guessed and attack procedure is repeated.
   The success probability such that two hash values are equal is $1/|H(C_2)|$ which takes on an average $2^{|H(C_2)|}$ trials. However, the total number of possible values of keys K_1 used to decrypt the helper data to obtain a valid sample space of hash value

$(H(C_2))'$ is lower, given as $|K_1| \leqslant |H(C_2)|$. Therefore, the attacker would choose to guess the values of the key K_1 directly. Hence, the number of trials required to guess a correct key K_1', considering the internal security of fuzzy commitment scheme as p bits= $2^{min(|K_1|,p)}$, where $|.|$ denotes the size of parameter.

Given correct key K_1' and the known random value $rand_1$ obtained from the public helper data HD_1, the attacker can guess the correct biometric template B_1' with high probability,[1] using pre-image attack such that $H(B_1'||rand_1) = K_1'$. Therefore, the number of trials required to guess B_1', given a correct key K_1' is equal to $2^{|K_1|}$.

2. **Construction:** S_1
Mode: Offline
Threat Model: As attacker knows B_1' where $d(B_1, B_1') < t$, the aim is to recover B_2' or K_2' such that the authentication is successful.
The attacker can generate the key K_1' with the help of known B_1' and helper data HD_1 using the underlying algorithm of P_{au_1} module as
$\mathtt{P}_{\mathtt{au}_1}(HD_1', B_1') \longrightarrow (1, K_1')$.
Since, $d(B_1, B_1') < t$, output is 1. The generated K_1' is a correct key and is equal to K_1 which was generated during the enrolment phase. Attacker can then decrypt the given HD_2^* with the help of key K_1' using (3) to get $HD_2' = ((H(C_2))', rand_2', \delta_2')$. Note that K_2' does not play any role in enrolment or authentication phase. The attacker can try to guess B_2' directly such that $C_2'' = B_2' \oplus \delta_2'$ where C_2'' is decoded using error correcting codeword to obtain C_2'. From C_2', the attacker can compute $H(C_2')$ to checks the condition given in (1). Therefore, the number of trials needed to guess a correct B_2' with high probability[2] is equal number of trials needed to guess a correct hash of codeword $H(C_2'')$ that will match with the given hash $(H(C_2))'$. It is equal to $2^{min(|H(C_2)|,p)}$ trials, considering the internal security of second fuzzy commitment scheme as p bits.

3. **Construction:** S_1
Mode: Offline
Threat Model: As attacker does not know B_1' and B_2', the aim is to recover any or all of the B_1', K_1', K_2' and B_2' such that authentication is successful.
We use format-preserving encryption[2] in our proposed scheme to encrypt the helper data HD_2. Since B_1' is unknown, the attacker can guess key K_1' and perform the function FFD as shown in (3) to get HD_2' from HD_2^*. However, since format-preserving encryption scheme is used, the format of HD_2' remains same and is equal to the format of HD_2^*, irrespective of the secret key applied for decryption. Therefore, the attacker cannot guess whether the key K_1' is correct or not by observing the format of HD_2'. The attacker would perform an exhaustive search on all the possible values of key K_1'. For each possible K_1', it will decrypt the helper data HD_2^* to get the decrypted helper data $HD_2' = ((H(C_2))', rand_2', \delta_2')$. Thus the attacker would store all the possible values of $(H(C_2))'$ corresponding to each guessed K_1' in the form of a table.
The attacker can then guess the value of $H(C_2')$ and check if $H(C_2')$ matches with one of the hashes $(H(C_2))'$ stored in the table to satisfy (1). Note that attacker would prefer to guess the hash of codeword directly rather than the codeword since $|C_2| \gg |H(C_2)|$. If the guessed $H(C_2')$ does not match with any of the table entries, the attacker can guess a new hash value. Therefore, the best attack possible when both B_1' and B_2' are unknown is time-memory tradeoff attack given by security bound denoted as $T \times M$. T denotes the time that includes the number of trials taken for guessing all the possible values of key given as K_1' considering the internal security of fuzzy commitment scheme as p bits = $2^{min(|K_1|,p)}$. M denotes the memory used to store all the possible decrypted values $(H(C_2))'$ as a part of helper data. It is given as $2^{|K_1|}$. Thus the security would be given as time-memory tradeoff attack bound plus the number of trials of $H(C_2)$ that are needed to match the two hash values, considering internal security of fuzzy commitment scheme as p bits. It is given as $2^{min(|K_1|,p)+|K_1|} + 2^{min(|H(C_2)|,p)}$.
Given a valid K_1' (which validates the condition for successful authentication) and $rand_1$, with high probability[2], the attacker can guess the correct biometric template B_1' using pre-image attack such that $H(B_1'||rand_1) = K_1'$ in $2^{|K_1|}$ trials.
The attacker can recover B_2' with high probability[2] by guessing B_2' such that $C_2'' = B_2' \oplus \delta_2'$ where C_2'' is decoded using error correcting codeword to obtain C_2' with the condition given in (1).
The number of trials needed to guess a correct B_2' with high probability[2] is equal to guessing a correct hash of codeword $H(C_2')$ that will match with the given hash $(H(C_2))'$. It is equal to $2^{|H(C_2)|}$ trials.

CASE B. (Online mode setup). The attacker gives inputs in the form of biometric templates B_1 and B_2 to get the final output as 1 or 0 which indicates whether the authentication is successful or not. Given the public parameters as
$HD_1 = (H(C_1), rand_1, \delta_1)$ and
an encrypted helper data, HD_2^* such that
$FFE(K_1, HD_2) \longrightarrow HD_2^*$ where $HD_2 = (H(C_2), rand_2, \delta_2)$,
the attacker can run only the following 2 functions to generate the corresponding outputs as,

[1] considering false match rate (FMR), false non-match rate (FNMR) as negligible
[2] The rationale behind the use of format-preserving encryption scheme is given in Section 6

$S_1.Enrol(B_1, B_2) \longrightarrow (HD_1, HD_2^*)$,
$S_1.Auth(HD_1, B_1', HD_2^*, B_2') \longrightarrow 1/0$
Under the online mode setup, we analyse our first construction under three threat models as follows:

1. **Construction:** S_1
   Mode: Online
   Threat Model: As attacker knows B_2' where $d(B_2, B_2') < t$, the aim is to get successful authentication, i.e. it satisfies (2). Since, B_1' is unknown, K_1' is also unknown. The attacker can input the random biometric templates B_1'. The system would generate corresponding K_1' as an application of the fuzzy commitment scheme given as $H(B_1'\|rand_1) \longrightarrow K_1'$ (details of key generation are provided in Section 6 in (15). In general, the key K_1' is derived after truncation from the hash output $H(B_1\|rand_1)$ or it can be of same size as of hash output. Using K_1', the system would decrypt the helper data HD_2^* internally to get decrypted helper data HD_2' using (3), where $HD_2' = ((H(C_2))', rand_2', \delta_2')$.
   B_2' is known to the attacker, therefore, using the correct B_2' and the derived values of δ_2', the system would perform error correcting code decoding by performing $C_2'' = B_2' \oplus \delta_2'$ where C_2'' is decoded using error correcting codeword to obtain C_2'. The system would then verifies the condition given in (1). The process is repeated until the authentication is successful. Since the key K_1' is used to decrypt the helper data, the attacker would need to perform $H(B_1'\|rand_1)$ number of trials of B_1' to get all the possible values of key K_1'. Therefore, the number of trials of B_1' required to get successful authentication = $2^{|H(B_1\|rand_1)|}$ when key K_1' is derived after truncation from the hash output $H(B_1'\|rand_1)$ and is equal to $2^{|K_1|}$ if key $|K_1'| = |H(B_1'\|rand_1)|$.
2. **Construction:** S_1
   Mode: Online
   Threat Model: As attacker knows B_1' where $d(B_1, B_1') < t$, the aim is to get successful authentication, i.e. it satisfies (2). Attacker provides B_1' as input. Using it, the system would generate the key K_1' as $\mathtt{P}_{\mathtt{au}_1}(HD_1', B_1') \longrightarrow (1, K_1')$. Since, $d(B_1, B_1') < t, K_1'$ is a correct key with high probability[2] and is equal to K_1 generated during the enrolment phase. The key is used by system to decrypt HD_2^* to get the correct helper data as shown in (3). HD_2' is a correctly decrypted helper data and is given as $HD_2' = ((H(C_2))', rand_2', \delta_2')$.
   The attacker can guess B_2' such that on providing B_2' to the module P_{au_2}, the system would compute $C_2'' = B_2' \oplus \delta_2'$ where C_2'' is decoded using error correcting codeword to obtain C_2'. If the condition given in (1) is satisfied, the system will show successful authentication. Thus, the attacker would know if the given B_2' is correct or not.
   Therefore, the number of trials of B_2' required for successful authentication is equal to guessing a correct hash of codeword $H(C_2')$ that matches with the given hash $(H(C_2))'$. It sums to $2^{|H(C_2)|}$ trials. Note that K_2' does not play any role in the enrolment or authentication phase.
3. **Construction:** S_1
   Mode: Online
   Threat Model: As the attacker does not know B_1' and B_2', the aim is to get successful authentication, i.e. it satisfies (2). In the online attack mode, the attacker has no memory. The attacker needs to guess all possible combinations of B_1' and B_2' such that (1) is satisfied which gives output 1 to denote a successful authentication. Therefore, the number of trials that need to be performed to obtain the correct matching of two hash values is equivalent to the pre-image attack complexity, assuming H as a random oracle. Thus the number of trials is equal to $2^{|H(C_2)|}$.

5.1.1. FV-then-FV: S_2

Construction. FV-then-FV is constructed by combining two fuzzy vault schemes; each takes a different instance of the biometric template as input. Fig. 5 shows the enrolment and authentication phases.

Enrolment Phase: During the enrolment phase, $\mathtt{Q}_{\mathtt{en}_1}$ generates a helper data HD_1 using a random secret key K_1 and the input biometric template B_1. From the second input biometric data B_2 and another secret key K_2, helper data HD_2 is generated with the help of $\mathtt{Q}_{\mathtt{en}_2}$ module. Both the keys are internally generated by the system. The key K_1 encrypts the helper data HD_2 to generate a transformed helper data HD_2^* using format-preserving encryption scheme *FFE*.

Authentication phase: During the authentication phase, $\mathtt{Q}_{\mathtt{au}_1}$ helps to recover the key K_1'. If the biometric template B_1' given during authentication is similar to B_1, i.e. $|B_1 - B_1'| < \epsilon$, which denotes the set difference between two biometric templates, then K_1' is a correct key (i.e. equal to K_1). The *FFD* takes the key and encrypted helper data as input and decrypts the helper data denoted as HD_2'. The second biometric template B_2' and helper data HD_2' are given as inputs to $\mathtt{Q}_{\mathtt{au}_2}$. If the set difference between two biometric templates denoted as $|B_2 - B_2'| < \epsilon$, the key K_2' is recovered such that

$$H(K_2') = (H(K_2))' \tag{4}$$

where, $H(K_2')$ represents the hash of key K_2' generated from Q_{au_2} (fuzzy vault scheme) in the authentication phase. $(H(K_2))'$ represents the hash of the key K_2, which is stored as a part of helper data HD_2 during the enrolment in the encrypted form. If the above condition is satisfied, 1 is given as the output to show a successful authentication as

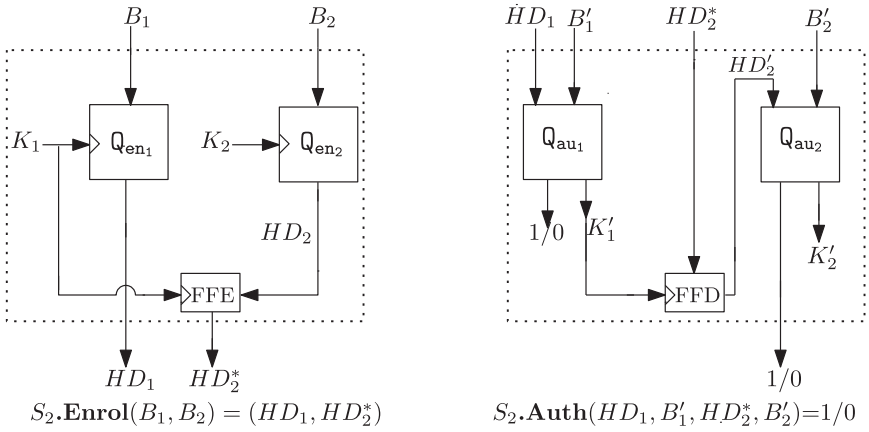


Fig. 5. Enrolment and authentication phase for the case: FV-then-FV. Q_{en_1} and Q_{en_2} represent the two instantiates of enrolment modules for fuzzy vault scheme respectively. Similarly, Q_{au_1} and Q_{au_2} represent the two instantiates of authentication modules for fuzzy vault scheme respectively. *FFE* and *FFD* denotes the format-preserving encryption and decryption respectively.

$$S_2.Auth(HD_1, B_1', HD_2^*, B_2') \longrightarrow 1 \tag{5}$$

Security Analysis. We discuss the security analysis of S_2 in 2 different modes: online and offline mode as follows:

CASE A. (Offline mode setup). In the mentioned setup, the attacker can know any of the secret parameters including the key(s), B_1 or B_2 as inputs. Given the public parameters as

$HD_1 = (V_1, H(K_1))$, and

an encrypted helper data, HD_2^* such that

$FFE(K_1, HD_2) \longrightarrow HD_2^*$ where $HD_2 = (V_2, H(K_2))$

the attacker can perform several functions to generate the corresponding outputs as,

$S_2.Enrol(B_1, B_2) \longrightarrow (HD_1, HD_2^*)$ where

$Q_{en_1}(B_1, K_1) \longrightarrow HD_1$,

$Q_{en_2}(B_2, K_2) \longrightarrow HD_2$,

$Q_{au_1}(HD_1, B_1') \longrightarrow (1/0, K_1')$,

$FFD(K_1', HD_2^*) \longrightarrow HD_2'$,

$Q_{au_2}(HD_2', B_2') \longrightarrow (1/0, K_2')$.

The attacker can individually run any or all the above-mentioned functions or algorithms such as *FFE*, Q_{en_1}, Q_{en_2}, Q_{au_1}, Q_{au_2} and *FFD* to get the respective outputs.

Under the mentioned setup, we analyse our first construction under three threat models as follows:

1. **Construction:** S_2
   Mode: Offline
   Threat Model: As attacker knows B_2' where $|B_2 - B_2'| < \epsilon$, the aim is to recover B_1' or K_1' such that the authentication is successful.
   Since, B_1' is unknown, correct key $K_1' = K_1$ is unknown. The attacker can guess a random value of K_1' and can decrypt the encrypted helper data HD_2^* as

$$FFD(K_1', HD_2^*) \longrightarrow HD_2' \tag{6}$$

   where $HD_2' = V_2' \| (H(K_2))'$. Since B_2' is known to the attacker, therefore, using the correct B_2' and the derived value of vault V_2' corresponding to the guessed key K_1', the attacker can derive the key K_2' using polynomial interpolation. Computing $H(K_2')$, the attacker can check the condition as given in (4). If the condition is satisfied, the authentication is successful for the corresponding guessed key K_1', else a new value of K_1' is guessed, and attack procedure is repeated.
   Since the success probability such that two hash values are equal is computed as $1/|H(K_2)|$, on average the total number of trials required are $2^{|H(K_2)|}$. However, the total number of possible values of keys K_1 used to decrypt the helper data to obtain the valid sample space of hash value $(H(K_2))'$ is smaller, given as $|K_1| \leqslant |H(K_2)|$. Therefore, the attacker would choose to guess the values of the key K_1 directly. Hence, the number of trials required to guess a correct key K_1', considering the internal security of fuzzy vault scheme as q bits= $2^{min(|K_1|, q)}$, where $|.|$ denotes the size of parameter.
   Given correct key K_1' and V_1', the attacker can recover the correct biometric template B_1' with high probability[2], by separating out the genuine and chaff points.

2. **Construction:** S_2
   Mode: Offline
   Threat Model: As attacker knows B_1' where $|B_1 - B_1'| < \epsilon$, the aim is to recover B_2' or K_2' such that the authentication is successful.
   The attacker can generate the key K_1' with the help of known B_1' and public helper data HD_1 as $\mathcal{Q}_{au_1}(HD_1, B_1') \longrightarrow (1, K_1')$. Since, $|B_1 - B_1'| < \epsilon, K_1'$ is a correct key and is equal to K_1 which is generated during the enrolment phase. Attacker can decrypt the encrypted helper data HD_2^* with the help of the correct key using (6) to get a correctly decrypted helper data $HD_2' = (V_2' || (H(K_2))')$. Given $(H(K_2))'$, the attacker can guess all the possible values of key K_2'. It can check if any of the guessed key satisfies (4).
   Therefore, the number of trials required to guess the correct key K_2', while considering the security of fuzzy vault scheme as q bits = $2^{min(|K_2|,q)}$. Note that the pre-image attack complexity will not hold here when $|K_2| \leqslant |H(K_2)|$, where H is the given hash function that acts as a random oracle.
   With the help of correct key K_2' and decrypted value of vault V_2', B_2' can be recovered with high probability[2] by separating out the genuine and chaff points.
3. **Construction:** S_2
   Mode: Offline
   Threat Model: As attacker does not know B_1' and B_2', the aim is to recover any or all of the B_1', K_1', K_2' and B_2' such that authentication is successful.
   We use format-preserving encryption in our proposed scheme to encrypt the helper data HD_2. Since B_1' is unknown, the attacker can guess key K_1' and perform the function *FFD* as shown in (6) to get HD_2' from HD_2^*. However, since format-preserving encryption scheme is used, the format of HD_2' remains same and is equal to the format of HD_2^*, irrespective of the secret key applied for decryption. Therefore, the attacker cannot guess whether the key K_1' is correct or not by observing the format of HD_2'. The attacker can perform an exhaustive search on all the possible values of key K_1'. For each possible K_1', it can decrypt the helper data HD_2^* to get the decrypted helper data $HD_2' = V_2' || (H(K_2))'$.
   The attacker can store all the possible values of $(H(K_2))'$ corresponding to each guessed K_1' in the form of a table. It can guess the value of K_2' and can compute $H(K_2')$ to check if $H(K_2')$ matches with one of the hashes stored in the table by satisfying (4). If the guessed K_2' for which $H(K_2')$ does not match with any of the table entries, the attacker could guess a new key K_2'.
   Therefore, the best attack possible when both B_1' and B_2' are unknown is time-memory tradeoff attack given by security bound denoted as $T \times M$. T denotes the time that includes the number of trials taken for guessing all the possible values of key given as K_1' considering the internal security of fuzzy vault scheme as q bits = $2^{min(|K_1|,q)}$ and M denotes the memory used to store all the possible decrypted values $(H(K_2))'$ as a part of helper data. It is given as $2^{|K_1|}$. Thus the security would be given as time-memory tradeoff attack bound plus the number of trials of K_2 that are needed to match the two hash values, considering internal security of fuzzy vault scheme as q bits. It is given as $2^{min(|K_1|,q)+|K_1|} + 2^{min(|K_2|,q)}$.
   With the help of a valid key K_1' and the given vault V_1 (as a part of helper data HD_1), B_1' can be recovered by separating out the genuine and chaff points with high probability[2].
   With the help of a valid key K_2' and the vault V_2' decrypted by a valid key K_1', B_2' can be recovered by separating out the genuine and chaff points with high probability[2].

CASE B. (Online mode setup). In the mentioned setup, the attacker gives inputs in the form of biometric templates B_1 and B_2 to get the final output as 1 or 0 which indicates whether the authentication is successful or not. Given the public parameters as

$HD_1 = V_1 || H(K_1)$ and
an encrypted helper data, HD_2^* such that
$FFE(K_1, HD_2) \longrightarrow HD_2^*$ where
$HD_2 = V_2 || H(K_2)$
the attacker can run only the following 2 functions to generate the corresponding outputs as,
$S_2.Enrol(B_1, B_2) \longrightarrow (HD_1, HD_2^*)$ and
$S_2.Auth(HD_1, B_1', HD_2^*, B_2') \longrightarrow 1/0$
Under the mentioned setup, we analyse our first construction under three threat models as follows:

1. **Construction:** S_2
   Mode: Online
   Threat Model: As attacker knows B_2' where $|B_2 - B_2'| < \epsilon$, the aim is to get successful authentication, i.e. it satisfies (5). Since, B_1' is unknown, K_1' is also unknown. The attacker can provide B_1' as input. The system can derive K_1' by applying polynomial interpolation on the B_1' and V_1 obtained as a part of public helper data HD_1. The key is derived as $\mathcal{Q}_{au_1}(HD_1, B_1') \longrightarrow (1/0, K_1')$.

Using the derived K'_1, the system can decrypt the encrypted helper data internally using (6) to get the decrypted helper data $HD'_2 = V'_2 \| (H(K_2))'$. It implies that for different values of guessed B'_1, corresponding values of vault V'_2 would be derived. B'_2 is known to the attacker, therefore, using the correct B'_2 and all the possible values of V'_2, the system can derive the corresponding key K'_2. It can compute $H(K'_2)$ and verify it with $(H(K_2))'$ obtained as a part of helper data. If (4) is satisfied, K'_2 is a correct key, else the process is repeated with a new value of B'_1.

Note that the attacker can only give B'_1 as input to the system and not the key K'_1. Therefore, the number of trials of B'_1 needed to get correct key K'_1 which further leads to successful authentication is equal to $2^{|K_1|}$, where $|.|$ denotes the size of parameter. We consider that within $2^{|K_1|}$ trials of B'_1, the system would be able to get a correct key K'_1 that will decrypt the helper data correctly with high probability[2].

2. **Construction:** S_2
   Mode: Online
   Threat Model: As attacker knows B'_1 where $|B_1 - B'_1| < \epsilon$, the aim is to get successful authentication, i.e. it satisfies (5). Attacker can provide B'_1 as input. The system can the generate the key K'_1 using the helper data HD_1 and B'_1. Since, $|B_1 - B'_1| < \epsilon), K'_1$ is a correct key, equal to K_1. The key can be used to decrypt HD^*_2 to get the correct helper data through (6) as $HD'_2 = (V'_2 \| (H(K_2))')$.
   Attacker can guess the biometric template B'_2. With the help of B'_2 and correct value of vault V'_2, the key K'_2 is generated by system using polynomial interpolation. If B'_2 is correct, with high probability[2], K'_2 is correctly generated by the system, satisfying (4).
   Since $(H(K_2))'$ is always correct, therefore, the number of trials of B'_2 required for successful authentication is equal to the sample space of $|K_2| = 2^{|K_2|}$. Note that the pre-image attack complexity will not hold here when $|K_2| \leqslant |H(K_2)|$, where H is the given hash function that acts as a random oracle. We consider that within $2^{|K_2|}$ trials of B'_2, the system would be able to get a correct key K'_2 to authenticate the attacker successfully with high probability[2].
3. **Construction:** S_2
   Mode: Online
   Threat Model: As the attacker does not know B'_1 and B'_2, the aim is to get successful authentication, i.e. it satisfies (5). In the online attack mode, the attacker has no memory. The attacker would guess all possible combinations of B'_1 and B'_2 such that (4) is satisfied which gives output 1 to denote a successful authentication. Therefore, the number of trials of B'_1 and B'_2 that need to be performed to obtain correct matching of two hash values is equivalent to the pre-image attack complexity, assuming H as a random oracle. Thus the number of trials is equal to $2^{|H(K_2)|}$. Note that the attacker cannot obtain or guess the key K_2 during an online attack and gets only 1/0 as the output. Therefore, the scope of performing collision attack on the system by finding two keys, K_2 and K'_2 such that their hash matches with each other is negligible.

5.1.2. FC-then-FV: S_3

Construction. FC-then-FV is constructed by combining a fuzzy commitment scheme (FC) with a fuzzy vault scheme (FV). Fig. 6 shows the enrolment and authentication phases.

Enrolment Phase: During the enrolment phase given an input biometric template B_1, $\mathbb{P}_{en}$ generates a helper data HD_1 and a cryptographic key K_1. From the second input biometric data B_2, another helper data HD_2 is generated with the help of a random secret key K_2. The key K_1 encrypts the helper data HD_2 to generate a transformed helper data HD^*_2 using a format-preserving encryption scheme *FFE*.

Authentication phase: During the authentication phase, given HD_1 and B'_1 as inputs, $\mathbb{P}_{au}$ helps to recover the key K'_1. If the biometric template B'_1 given during authentication is similar to B_1, i.e. $d(B_1, B'_1) < t$, where t is a pre-defined threshold and d is the hamming distance between two binary strings, then K'_1 is a correct key (i.e. equal to K_1). The *FFD* takes the key and encrypted helper data as inputs and decrypts the helper data as HD'_2. The second biometric template B'_2 and helper data HD'_2 are given as inputs to $\mathbb{Q}_{au}$. If the set difference between two biometric templates denoted as $|B_2 - B'_2| < \epsilon$, the key K'_2 is recovered so that

$$H(K'_2) = (H(K_2))' \tag{7}$$

$H(K'_2)$ represents the hash of key K'_2 generated from the module $\mathbb{Q}_{au}$ (fuzzy vault scheme) in the authentication phase. $(H(K_2))'$ denotes the hash of the key K_2 which is stored as a part of helper data HD_2 during the enrolment in the encrypted form. If the above condition is satisfied, 1 is given as output to show a successful authentication denoted as

$$S_3.Auth(HD_1, B'_1, HD^*_2, B'_2) \longrightarrow 1 \tag{8}$$

Security Analysis. We discuss the security analysis of S_3 in 2 different modes: online and offline mode as follows:

CASE A. (Offline mode setup). In the mentioned setup, the attacker can know any of the secret parameters including the key(s), B_1 or B_2 as inputs. Given the public parameters as

$HD_1 = (H(C_1), rand_1, \delta_1)$ and

an encrypted helper data, HD^*_2 such that

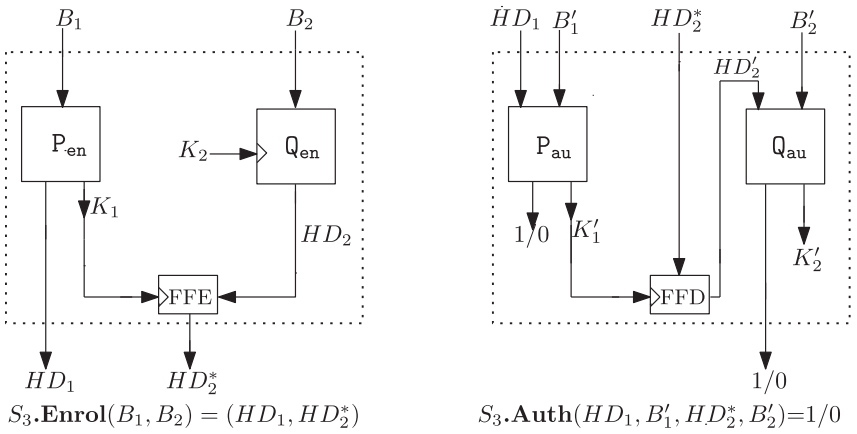


Fig. 6. Enrolment and authentication phase for the case: FC-and-FV. $\mathtt{P}_{\mathtt{en}}$ and $\mathtt{Q}_{\mathtt{en}}$ represent the enrolment modules for fuzzy commitment and fuzzy vault scheme respectively. Similarly, $\mathtt{P}_{\mathtt{au}}$ and $\mathtt{Q}_{\mathtt{au}}$ represent the authentication modules for fuzzy commitment and fuzzy vault scheme respectively. Refer Figs. 2 and 3 for details. *FFE* and *FFD* denotes the format-preserving encryption and decryption respectively.

$FFE(K_1, HD_2) \longrightarrow HD_2^*$ where $HD_2 = (V_2, H(K_2))$
the attacker can perform several functions to generate the corresponding outputs as,
$S_3.Enrol(B_1, B_2) \longrightarrow (HD_1, HD_2^*)$ where
$FFE(K_1, HD_2) \longrightarrow HD_2^*$,
$\mathtt{P}_{\mathtt{en}}(B_1) \longrightarrow (HD_1, K_1)$,
$\mathtt{Q}_{\mathtt{en}}(B_2, K_2) \longrightarrow HD_2$,
$\mathtt{P}_{\mathtt{au}}(HD_1, B_1') \longrightarrow (1/0, K_1')$,
$FFD(K_1', HD_2^*) \longrightarrow HD_2'$,
$\mathtt{Q}_{\mathtt{au}}(HD_2', B_2') \longrightarrow (1/0, K_2')$
The attacker can individually run any or all the above-mentioned functions or algorithms such as *FFE*, $\mathtt{P}_{\mathtt{en}}, \mathtt{Q}_{\mathtt{en}}, \mathtt{P}_{\mathtt{au}}, \mathtt{Q}_{\mathtt{au}}$ and *FFD* to get the respective outputs.
Under the mentioned setup, we analyse our first construction under three threat models as follows:

1. **Construction:** S_3
Mode: Offline
Threat Model: As attacker knows B_2' where $|B_2 - B_2'| < \epsilon$, the aim is to recover B_1' or K_1' such that the authentication is successful.
Since, B_1' is unknown, correct key $K_1' = K_1$ is unknown. The attacker can guess a random value of K_1' and decrypt the encrypted helper data HD_2^* as

$$FFD(K_1', HD_2^*) \longrightarrow HD_2' \tag{9}$$

where $HD_2' = V_2' \| (H(K_2))'$. Since, B_2' is known to the attacker, therefore, using the correct B_2' and the derived value of vault V_2' corresponding to the guessed key K_1', the attacker can derive the key K_2' using polynomial interpolation. Using $H(K_2')$, the attacker can check the condition as given in (7). If the condition is satisfied, the authentication is successful for the corresponding guessed key K_1', else a new value of K_1' is guessed and attack procedure is repeated.
Since the success probability such that two hash values are equal is computed as $1/|H(K_2)|$, on average the total number of trials required are $2^{|H(K_2)|}$. However, the total number of possible values of keys K_1 used to decrypt the helper data to obtain the valid sample space of the hash value $(H(K_2))'$ is smaller, given as $|K_1| \leqslant |H(K_2)|$. Therefore, the attacker would choose to guess the value of the key K_1 directly. Hence, the number of trials required to guess a correct key K_1', considering the internal security of fuzzy commitment scheme as p bits= $2^{min(|K_1|,p)}$, where $|.|$ denotes the size of parameter.
Given correct key K_1' and the known random value $rand_1$ obtained from public HD_1, the attacker can guess the correct biometric template B_1' with high probability[2], using pre-image attack by performing the following function given as,

$$H(B_1' \| rand_1) = K_1'$$

Therefore, the number of trials required to guess B_1' given a correct key K_1' is equal to $2^{|K_1|}$.
2. **Construction:** S_3
Mode: Offline
Threat Model: As attacker knows B_1' where $d(B_1, B_1') < t$, the aim is to recover B_2' or K_2' such that the authentication is

successful.

The attacker can generate the key K_1' with the help of known B_1' and helper data HD_1 as $\mathbb{P}_{au}(HD_1, B_1') \longrightarrow (1, K_1')$. Since, $d(B_1, B_1') < t$), the output key K_1' is a correct key and is equal to K_1 as generated during the enrolment phase. Attacker can then decrypt the encrypted helper data HD_2^* with the help of key K_1' using (9) to get the original helper data $HD_2' = (V_2' || (H(K_2))')$.

Given $(H(K_2))'$ as the correct helper data, the attacker can guess all the possible values of key K_2' and compute $H(K_2')$. It can check if the hash of any of the guessed key K_2' satisfies the condition given in (7). Therefore, the number of trials required to guess the correct key K_2', while considering the security of fuzzy vault scheme as q bits = $2^{min(|K_2|,q)}$. Note that the pre-image attack complexity will not hold when $|K_2| \leqslant |H(K_2)|$, where H is the given hash function that acts as a random oracle.

With the help of correct key K_2' and decrypted value of vault V_2', B_2' can be recovered with high probability[2] by separating out the genuine and chaff points.

3. **Construction:** S_3
   Mode: Offline
   Threat Model: As attacker does not know B_1' and B_2', the aim is to recover any or all of the B_1', K_1', K_2' and B_2' such that authentication is successful.
   We use format-preserving encryption in our proposed scheme to encrypt the helper data HD_2. Since B_1' is unknown, the attacker can guess key K_1' and perform the function FFD as shown in (9) to get HD_2' from HD_2^*. However, since format-preserving encryption scheme is used, the format of HD_2' remains same and is equal to the format of HD_2^*, irrespective of the secret key applied for decryption. Therefore, the attacker cannot guess whether the key K_1' is correct or not by observing the format of HD_2'. The attacker would have to perform an exhaustive search on all the possible values of key K_1'. For each possible K_1', it can decrypt the helper data HD_2^* using (9) to get the decrypted helper data $HD_2' = V_2' || (H(K_2))'$. Thus the attacker can store all the possible values of $(H(K_2))'$ corresponding to each guessed K_1' in the form of a table.
   The attacker can guess the value of K_2' and check if $H(K_2')$ matches with one of the hashes stored in the table to satisfy (7). If the guessed K_2' for which $H(K_2')$ does not match with any of the table entries, the attacker could guess a new key K_2'. Therefore, the best attack possible when both B_1' and B_2' are unknown is time-memory tradeoff attack given by security bound denoted as $T \times M$. T denotes the time that includes the number of trials taken for guessing all the possible values of key given as K_1', considering the internal security of fuzzy commitment scheme as p bits = $2^{min(|K_1|,p)}$ and M denotes the memory used to store all the possible decrypted values of helper data and is given as $2^{|K_1|}$. Thus the security would be given as time-memory tradeoff attack bound plus the number of trials of K_2 that are needed to match the two hash values, considering internal security of fuzzy vault scheme as q bits. It is given as $2^{min(|K_1|,p)+|K_1|} + 2^{min(|K_2|,q)}$.
   Given a valid K_1' which satisfies (7) and $rand_1$, with high probability[2], the attacker can guess the correct biometric template B_1' using pre-image attack such that $H(B_1' || rand_1) = K_1'$ in $2^{|K_1|}$ trials. With the help of a valid key K_2' which satisfies (7) and the vault V_2' decrypted by a valid key K_1', B_2' can be recovered by separating out the genuine and chaff points with high probability[2].

CASE B. (Online mode setup). In the mentioned setup, the attacker gives inputs in the form of biometric templates B_1 and B_2 to get the final output as 1 or 0 which indicates whether the authentication is successful or not. Given the public parameters as

$HD_1 = (H(C_1), rand_1, \delta_1)$ and
an encrypted helper data, HD_2^* such that
$FFE(K_1, HD_2) \longrightarrow HD_2^*$ where
$HD_2 = V_2 || H(K_2)$.
the attacker can run only the following 2 functions to generate the corresponding outputs as,
Given, $S_3.Enrol(B_1, B_2) \longrightarrow (HD_1, HD_2^*)$ and
$S_3.Auth(HD_1, B_1', HD_2^*, B_2') \longrightarrow 1/0$
Under the mentioned setup, we analyze our first construction under three threat models as follows:

1. **Construction:** S_3
   Mode: Online
   Threat Model: As attacker knows B_2' where $|B_2 - B_2'| < \epsilon$, the aim is to get successful authentication, i.e. it satisfies (8). Since B_1' is unknown, K_1' is also unknown. The attacker can input a random or guessed biometric template B_1'. The system can then generate the corresponding K_1' as an application of the fuzzy commitment scheme given as $H(B_1' || rand_1) \longrightarrow K_1'$ (details of key generation are provided in Section 6 in (15). In general, the key K_1' is derived after truncation from the hash output $H(B_1 || rand_1)$, or it can be of the same size as of hash output. Using the derived K_1', the system can decrypt the helper data internally to get decrypted helper data HD_2' with the help of (9), where $HD_2' = V_2' || (H(K_2))'$. It implies that

for different values of the B_1', corresponding values of vault V_2' would be derived by the system. B_2' is known to the attacker and is provided as another input. Using the correct B_2' and the possible values of V_2', the system can derive the corresponding key K_2'. It can compute $H(K_2')$ and verify it with the hash value $(H(K_2))'$ obtained as the part of helper data HD_2'. If (7) is satisfied, the key K_2' is a correct key, else the process is repeated with a new value of B_1'.

Since the key K_1' is used to decrypt the helper data, the attacker needs to perform $H(B_1'||rand_1)$ number of trials of B_1' to get all the possible values of key K_1'. Therefore, the number of trials of B_1' required to get successful authentication = $2^{|H(B_1||rand_1)|}$ when key K_1' is derived after truncation from the hash output $H(B_1'||rand_1)$ and is equal to $2^{|K_1|}$ if key $|K_1'| = |H(B_1'||rand_1)|$.

2. **Construction:** S_3
   Mode: Online
   Threat Model: As attacker knows B_1' where $d(B_1, B_1') < t$, the aim is to get successful authentication, i.e. it satisfies (8).
   Attacker can provide B_1' as input, using which the system can generate the correct key K_1' as $\mathtt{P}_{au}(HD_1', B_1') \longrightarrow (1, K_1')$. Since, $d(B_1, B_1') < t, K_1'$ is a correct key with high probability[2] and is equal to K_1 generated during the enrolment phase. The key can be used by system to decrypt HD_2^* to get the correct helper data $HD_2' = (V_2'||(H(K_2))')$ using (9).
   The attacker can guess B_2' such that on providing B_2' to the module Q_{au}, the system will compute the key K_2' using B_2' and polynomial interpolation on correct value of vault V_2'. If B_2' is correct, with high probability[2], K_2' is correctly derived and is verified using the condition given in (7).
   Since $(H(K_2))'$ is always correct, therefore, the number of trials of B_2' required for successful authentication is equal to the sample space of $|K_2| = 2^{|K_2|}$. Note that the pre-image attack complexity will not hold when $|K_2| \leqslant |H(K_2)|$, where H is the given hash function that acts as a random oracle. We consider that within $2^{|K_2|}$ trials of B_2', the system would be able to get a correct key K_2' to authenticate the attacker successfully with high probability[2].
3. **Construction:** S_3
   Mode: Online
   Threat Model: As the attacker does not know B_1' and B_2', the aim is to get successful authentication, i.e. it satisfies (8).
   In the online attack mode, the attacker has no memory. The attacker would need to guess all possible combinations of B_1' and B_2' such that (7) is satisfied which gives output 1 to denote a successful authentication. Therefore, the number of trials that need to be performed to obtain the correct matching of two hash values is equivalent to the pre-image attack complexity, assuming H as a random oracle. Thus the number of trials is equal to $2^{|H(K_2)|}$. Note that the attacker cannot obtain or guess the key K_2 during an online attack. Therefore, the scope of performing collision attack on the system by finding two keys, K_2 and K_2' such that their hash matches with each other is negligible.

5.1.3. FV-then-FC: S_4

Construction FV-then-FC is constructed by combining a fuzzy vault scheme (FV) with a fuzzy commitment scheme (FC). Fig. 7 shows the enrolment and authentication phases.

Enrolment Phase: During the enrolment phase, $\mathtt{Q}_{en}$ generates a helper data HD_1, given a secret cryptographic key K_1 generated internally and the user's biometric template B_1. From the second input biometric data B_2, another helper data HD_2 is generated along with a key K_2. The key K_1 encrypts the helper data HD_2 to generate a transformed helper data HD_2^* using a format-preserving encryption scheme *FFE*. K_2 does not play any role in enrolment.

Authentication phase: During the authentication phase, given HD_1 and B_1' as inputs, $\mathtt{Q}_{au}$ helps to recover the key K_1'. If the biometric template B_1' given during authentication is similar to B_1, i.e. $|B_1 - B_1'| < \epsilon$, then K_1' is a correct key (i.e. equal to K_1). The *FFD* takes the key and encrypted helper data as inputs and generates the decrypted helper data HD_2'. With the help of second biometric template B_2' and helper data HD_2', hash verification is performed inside the $\mathtt{P}_{au}$ module such that

$$H(C_2') = (H(C_2))' \tag{10}$$

where $H(C_2')$ represents the hash of codeword C_2' generated internally from P_{au_2} (fuzzy commitment scheme) in the authentication phase. $(H(C_2))'$ represents the hash of the codeword C_2, which is stored as a part of helper data HD_2 during the enrolment in the encrypted form. If the condition is satisfied, authentication is successful with output 1 and is denoted as

$$S_4.Auth(HD_1, B_1', HD_2^*, B_2') \longrightarrow 1 \tag{11}$$

K_2' generated as output does not play any role in the authentication.

Security Analysis. We discuss the security analysis of S_4 in 2 different modes: online and offline mode as follows:

CASE A. (Offline mode setup). In the mentioned setup, the attacker may know some of the secret parameters including the key(s), B_1 or B_2 as inputs. Given the public parameters as

$HD_1 = (V_1, H(K_1))$ and

an encrypted helper data, HD_2^* such that

$FFE(K_1, HD_2) \longrightarrow HD_2^*$ where

$HD_2 = (H(C_2), rand_2, \delta_2)$,

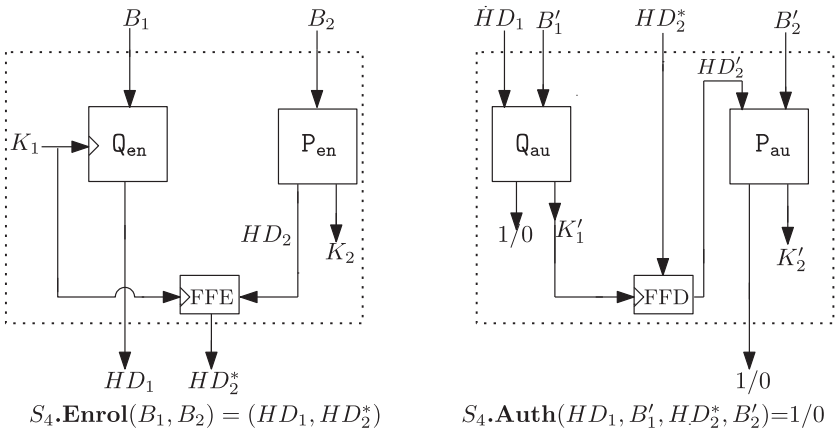


Fig. 7. Enrolment and authentication phase of possible combinations from FV-then-FC. $\mathtt{P}_{\mathtt{en}}$ and $\mathtt{Q}_{\mathtt{en}}$ represent the enrolment modules for fuzzy commitment and fuzzy vault scheme respectively. Similarly, $\mathtt{P}_{\mathtt{au}}$ and $\mathtt{Q}_{\mathtt{au}}$ represent the authentication modules for fuzzy commitment and fuzzy vault scheme respectively. Refer Figs. 2 and 3 for details. *FFE* and *FFD* denotes the format-preserving encryption and decrypti.on respectively.

the attacker can perform several functions to generate the corresponding outputs as,
$S_4.Enrol(B_1, B_2) \longrightarrow (HD_1, HD_2^*)$ where
$\mathtt{Q}_{\mathtt{en}}(B_1, K_1) \longrightarrow HD_1$,
$\mathtt{P}_{\mathtt{en}}(B_2) \longrightarrow (HD_2, K_2)$,
$\mathtt{Q}_{\mathtt{au}}(HD_1, B_1') \longrightarrow (1/0, K_1')$,
$FFD(K_1', HD_2^*) \longrightarrow HD_2'$,
$\mathtt{P}_{\mathtt{au}}(HD_2', B_2') \longrightarrow (1/0, K_2')$
The attacker can individually run any or all the above-mentioned functions or algorithms such as *FFE*, $\mathtt{P}_{\mathtt{en}}$, $\mathtt{Q}_{\mathtt{en}}$, $\mathtt{P}_{\mathtt{au}}$, $\mathtt{Q}_{\mathtt{au}}$ and *FFD* to get the respective outputs.

Under the mentioned setup, we analyse our first construction under three threat models as follows:

1. **Construction:** S_4
   Mode: Offline
   Threat Model: As attacker knows B_2' where $d(B_2, B_2') < t$, the aim is to recover B_1' or K_1' such that the authentication is successful.
   Since, B_1' is unknown, correct key $K_1' = K_1$ is unknown. The attacker can guess K_1' and perform the function *FFD* to get HD_2' given as,

$$FFD(K_1', HD_2^*) \longrightarrow HD_2' \tag{12}$$

   where $HD_2' = ((H(C_2))', rand_2', \delta_2')$. Further, the biometric template B_2' is known to the attacker, therefore, using the correct B_2' and all the values of δ_2' derived from the helper data, the attacker can perform $C_2'' = B_2' \oplus \delta_2'$ where C_2'' is decoded using error correcting codeword to obtain C_2'. From C_2', the attacker can compute $H(C_2')$ to check if the condition in (10) is satisfied. If the condition is satisfied, the authentication is successful for the corresponding guessed key K_1', else a new value of K_1' is guessed and attack procedure is repeated.
   Since the success probability such that two hash values are equal is computed as $1/|H(C_2)|$, on average the total number of trials required are $2^{|H(C_2)|}$. However, the total number of possible values of keys K_1 used to decrypt the helper data to obtain a valid sample space of hash value $(H(C_2))'$ is smaller, given as $|K_1| \leqslant |H(C_2)|$. Therefore, the attacker would choose to guess the values of the key K_1 directly. Hence, the number of trials required to guess a correct key K_1', considering the internal security of fuzzy vault scheme as q bits= $2^{min(|K_1|,q)}$, where $|.|$ denotes the size of parameter.
   Given correct key K_1' and V_1', the attacker can recover the correct biometric template B_1' with high probability[2], by separating out the genuine and chaff points.
2. **Construction:** S_4
   Mode: Offline
   Threat Model: As attacker knows B_1' where $|B_1 - B_1'| < \epsilon$, the aim is to recover B_2' or K_2' such that the authentication is successful.

The attacker can generate the key K_1' with the help of known B_1' and helper data HD_1 as $\mathcal{Q}_{gu}(HD_1', B_1') \longrightarrow (1, K_1')$.
$|B_1 - B_1'| < \epsilon, K_1'$ is a correct key and is equal to K_1 generated during the enrolment phase. Attacker can then decrypt the given HD_2^* with the help of key K_1' using (12) to get the correct $HD_2' = ((H(C_2))', rand_2', \delta_2')$.
Note that K_2' does not play any role in enrolment or authentication phase. The attacker can guess B_2' directly such that $C_2'' = B_2' \oplus \delta_2'$ where C_2'' is decoded using error correcting codeword to obtain C_2'. From C_2', the attacker can compute $H(C_2')$. Attacker can then check the condition given in (10). Therefore, the number of trials needed to guess a correct B_2' with high probability[2]is equal number of trials needed to guess a correct hash of codeword $H(C_2'')$ that will match with the given hash $(H(C_2))'$. It is equal to $2^{min(|H(C_2)|,p)}$ trials, considering the internal security of second fuzzy commitment scheme as p bits.

3. **Construction:** S_4
Mode: Offline
Threat Model: As attacker does not know B_1' and B_2', the aim is to recover any or all of the B_1', K_1', K_2' and B_2' such that authentication is successful.
We use format-preserving encryption in our proposed scheme to encrypt the helper data HD_2. Since B_1' is unknown, the attacker can guess key K_1' and can perform the function *FFD* as shown in (12) to get HD_2' from HD_2^*. However, since format-preserving encryption scheme is used, the format of HD_2' remains same and is equal to the format of HD_2^*, irrespective of the secret key applied for decryption. Therefore, the attacker cannot guess whether the key K_1' is correct or not by observing the format of HD_2'. The attacker would have to perform an exhaustive search on all the possible values of key K_1'. For each possible K_1', it can decrypt the helper data HD_2^* to get the decrypted helper data HD_2' which is given as $HD_2' = ((H(C_2))', rand_2', \delta_2')$. The attacker can store all the possible values of $(H(C_2))'$ corresponding to each guessed K_1' in the form of a table.
The attacker can then guess the value of $H(C_2')$ and can check if $H(C_2')$ matches with one of the hashes $(H(C_2))'$ stored in the table to satisfy (10). Note that the attacker would prefer to guess the hash of codeword directly rather than the codeword since $|C_2| \gg |H(C_2)|$. If the guessed $H(C_2')$ does not match with any of the table entries, the attacker can guess a new hash value.
Therefore, the best attack possible when both B_1' and B_2' are unknown is time-memory tradeoff attack given by security bound denoted as $T \times M$. T denotes the time that includes the number of trials taken for guessing all the possible values of key given as K_1' considering the internal security of fuzzy vault scheme as q bits = $2^{min(|K_1|,q)}$. M denotes the memory used to store all the possible decrypted values $(H(C_2))'$ as a part of helper data. It is given as $2^{|K_1|}$. Thus the security would be given as time-memory tradeoff attack bound plus the number of trials of $H(C_2)$ that are needed to match the two hash values, considering internal security of fuzzy commitment scheme as p bits. It is given as $2^{min(|K_1|,q)+|K_1|} + 2^{min(|H(C_2)|,p)}$.
Given a valid K_1' and public vault V_1, the attacker can guess the correct biometric template B_1' with high probability[2], by separating out the genuine from chaff points.
The attacker can recover B_2' with high probability[2] by guessing B_2' such that $C_2'' = B_2' \oplus \delta_2'$ where C_2'' can be decoded using error correcting codeword to obtain C_2' with the condition that $H(C_2') = (H(C_2))'$. The number of trials needed to guess a correct B_2' with high probability[2] is equal to guessing a correct hash of codeword $H(C_2')$ that will match with the given hash $(H(C_2))'$. It is equal to $2^{|H(C_2)|}$ trials.

CASE B. (Online mode setup). In the mentioned setup, the attacker gives inputs in the form of biometric templates B_1 and B_2 to get the final output as 1 or 0 which indicates whether the authentication is successful or not. Given the public parameters as
$HD_1 = (V_1 \| H(K_1))$ and
an encrypted helper data, HD_2^* such that
$FFE(K_1, HD_2) \longrightarrow HD_2^*$ where $HD_2 = (H(C_2), rand_2, \delta_2)$,
the attacker can run only the following 2 functions to generate the corresponding outputs as,
$S_4.Enrol(B_1, B_2) \longrightarrow (HD_1, HD_2^*)$,
$S_4.Auth(HD_1, B_1', HD_2^*, B_2') \longrightarrow 1/0$
Under the mentioned setup, we analyze our first construction under three threat models as follows:

1. **Construction:** S_4
Mode: Online
Threat Model: As attacker knows B_2' where $d(B_2, B_2') < t$, the aim is to get successful authentication, i.e. it satisfies (11). Since, B_1' is unknown, K_1' is also unknown. The attacker can input the random biometric templates B_1'. The system can then generate a corresponding K_1' by applying polynomial interpolation on the guessed B_1' and given V_1 obtained from HD_1. The key is generated as $\mathcal{Q}_{au}(HD_1, B_1') \longrightarrow (1/0, K_1')$. Using the derived K_1', the system can decrypt the helper data internally to get

decrypted helper data $HD'_2 = ((H(C_2))', rand'_2, \delta'_2)$.
It implies that for different values of guessed B'_1, corresponding secure sketch value δ'_2 would be derived. B'_2 is known to the attacker that can be provided as input to the system. Therefore, using the correct B'_2 and the derived values of δ'_2, the system can perform error correcting code decoding by performing $C''_2 = B'_2 \oplus \delta'_2$ where C''_2 is decoded using error correcting codeword to obtain C'_2. System can verify the condition given in (10). The process is repeated until the authentication is successful.
Note that the attacker can only give B'_1 as input to the system and not the key K'_1. Therefore, the number of trials of B'_1 needed to get correct key K'_1 which further leads to successful authentication is equal to $2^{|K_1|}$, where $|.|$ denotes the size of parameter. We consider that within $2^{|K_1|}$ trials of B'_1, the system would be able to get a correct key K'_1 that will decrypt the helper data correctly with high probability[2].

2. **Construction:** S_4
Mode: Online
Threat Model: As attacker knows B'_1 where $|B_1 - B'_1| < \epsilon$, the aim is to get successful authentication, i.e. it satisfies (11). Attacker can provide B'_1 as input. Using it, the system can generate the key K'_1 as $\mathrm{P}_{\mathrm{au}}(HD'_1, B'_1) \longrightarrow (1, K'_1)$. Since, $|B_1 - B'_1| < \epsilon, K'_1$ is a correct key with high probability[2] and is equal to K_1 generated during the enrolment phase. The key can be used by system to decrypt HD^*_2 to get the correct helper data $HD'_2 = ((H(C_2))', rand'_2, \delta'_2)$. using (12).
The attacker can provide B'_2 such that on providing B'_2 to the module P_{au}, the system can compute $C''_2 = B'_2 \oplus \delta'_2$ where C''_2 is decoded using error correcting codeword to obtain C'_2. If the condition given in (10) is satisfied, the system would show successful authentication. Thus, the attacker can get to know if the given B'_2 is correct or not.
Therefore, the number of trials of B'_2 required for successful authentication is equal to guessing a correct hash of codeword $H(C'_2)$ that will match with the given hash $(H(C_2))'$. It sums to $2^{|H(C_2)|}$ trials. Note that K'_2 does not play any role in the enrolment or authentication phase.

3. **Construction:** S_4
Mode: Online
Threat Model: As the attacker does not know B'_1 and B'_2, the aim is to get successful authentication, i.e. it satisfies (11). In the online attack mode, the attacker has no memory. The attacker would need to guess all possible combinations of B'_1 and B'_2 such that (10) is satisfied which gives output 1 to denote a successful authentication. Therefore, the number of trials that need to be performed to obtain the correct matching of two hash values is equivalent to the pre-image attack complexity, assuming H as a random oracle. Thus the number of trials is equal to $2^{|H(C_2)|}$.

6. Secure construction of BIOFUSE: S-BIOFUSE (S_3)

In Section 5, we discussed the security analysis of four possible constructions in which we can combine two biocryptosystems. We found that the security bound for all the four constructions are comparable to each other. However, we consider the case FC-then-FV denoted by S_3 as the most secure case out of all, and we named it as S-BIOFUSE. The limitations of the other 3 cases are:

- **FC-then-FC:** The fuzzy commitment scheme takes a binary string as input. Hence in the case of fingerprint, etc. where features are not present in binary format, an additional feature extraction or transformation tool would be required.
- **FV-then-FV:** One of the vault data V_1 will remain unencrypted, which is prone to multiple security attacks such as brute force attack on vault, correlation attack, etc. Further, to use any biometric template with features in the binary string format, additional feature extraction or transformation tool would be required with the fuzzy vault.
- **FV-then-FC:** Similar to the above case, one of the vault data V_1 will remain unencrypted which is prone to multiple security attacks such as brute force attack on vault, correlation attack, etc.

We now discuss the enrolment and authentication phase of S-BIOFUSE in details.

6.1. Enrolment phase

The enrolment phase is described using Algorithm 1 and is shown in Fig. 8. The user provides first biometric characteristics, I_1 that represents the iris image (let say) from which a binary string B_1 known as iriscode is extracted [9].

To handle the errors present in the iriscode, an error correcting codeword C_1 is generated internally from a secret, pseudorandom message *msg*.

A secure sketch value δ_1 is constructed using fuzzy commitment scheme as $\delta_1 \longleftarrow B_1 \oplus C_1$. Further, a tweak value T used by the format-preserving encryption scheme is generated as

$$T \longleftarrow H_1(I_1 \| 1) \tag{13}$$

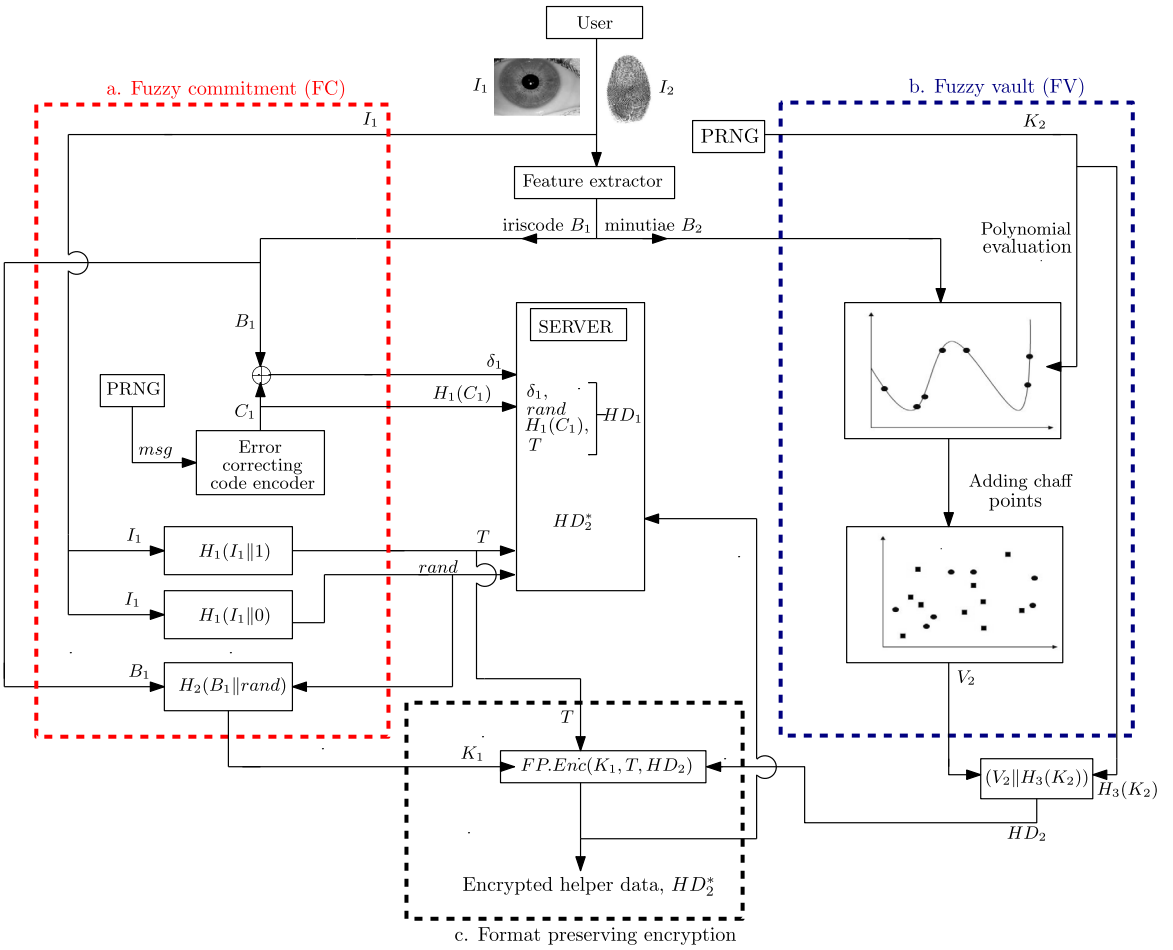


Fig. 8. Enrolment Phase- a. The fuzzy commitment scheme takes iris I_1 from which iriscode B_1 is extracted. It generates helper data HD_1 denoted as $(H(C_1), rand_1, \delta_1, T)$ along with the cryptographic key K_1. **b.** The Fuzzy vault scheme takes fingerprint template as minutiae B_2 along with an internal secret key K_2. Helper data HD_2 is generated as output and is denoted as $(V_2, H(K_2))$. **c.** HD_2 is encrypted by format-preserving encryption *FP.Enc* using key K_1 to generate encrypted output HD_2^*.

Algorithm 1: Enrolment Phase

Input: Iris image I_1, Fingerprint image I_2, iriscode B_1, minutiae set B_2

Output: Helper data HD_1, HD_2^*

▷ **Fuzzy commitment scheme:** outputs $HD_1 = (H(C_1), rand_1, \delta_1, T)$

$C_1 \xleftarrow{len} msg$ ▷ msg denotes a random secret message

$\delta_1 = B_1 \oplus C_1$

$T = H_1(I_1\|1)$ ▷ H_1, H_2, H_3: instantiates of hash functions

$rand_1 = H_1(I_1\|0)$

$K_1 = H_2(B_1\|rand_1)$

▷ **Fuzzy vault scheme:** outputs $HD_2 = V_2\|H_3(K_2)$

for $i = 1$ to l ▷ l: number of minutiae points in set

  $p(B_{2_i}) = K_2^{n-1}B_{2_i}^{n-1} + K_2^{n-2}B_{2_i}^{n-2} + \cdots K_2^1 B_{2_i}^1 + K_2^0$ ▷ $p(B_{2_i})$: polynomial construction with degree n

  $G = \{(B_{2_i}, p(B_{2_i}))\}$ ▷ G: genuine points

for $i = l + 1$ to r ▷ r: total number of points in vault

  $Ch = \{(z_i, u_i)\}$ is generated such that $z_i \notin \{B_{2_i}\}$ and

(*continued on next page*)

Algorithm 1 (*continued*)

$u_i \notin \{p(B_{2_i})\}$ ▷
Ch: chaff points
$V_2 = G \cup Ch$
▷ **Format-preserving Encryption:** outputs encrypted message HD_2^*
$HD_2^* = FP.Enc(K_1, T, HD_2)$

A random value $rand_1$ is also generated from the input I_1 as

$$rand_1 \longleftarrow H_1(I_1 \| 0) \tag{14}$$

These output values- $\delta_1, H_1(C_1), rand_1$ and T are stored as helper data HD_1 on the database server. Note that tweak T will be different every time (for similar but not identical samples of a particular instance) since it is generated from the hash of biometric characteristics. It prevents nonce misuse in the format-preserving encryption scheme. For simplicity, we do not include T while calculating security bounds in Section 5. H_1, H_2, H_3 denotes the instances of hash function that acts as a random oracle. Using the random value $rand_1$ in (14), a key K_1 is generated from biometric template B_1 through fuzzy extractor as

$$K_1 \longleftarrow H_2(B_1 \| rand_1) \tag{15}$$

Simultaneously, using the second biometric characteristics I_2 that represents a fingerprint, unique features known as minutiae points are extracted [18] and quantized to generate a set B_2 of the unordered points known as genuine points. A vault is constructed from the minutiae points and the random chaff points using an internal random key K_2 as proposed by Nandakumar et.al [34].

In BIOFUSE, instead of implementing CRC [38] to verify the correct polynomial during authentication, we use the hash of the secret key K_2 to verify the reconstruction of correct polynomial during authentication. HD_2 represents the concatenation of vault V_2 and the hash of the secret key used in the fuzzy vault scheme.

The key K_1 encrypts the helper data HD_2 using format-preserving encryption as $HD_2^* \longleftarrow FP.Enc(K_1, T, HD_2)$.

6.2. Authentication phase

The authentication phase is described by Algorithm 2 and is shown in Fig. 9. During authentication, using the first biometric characteristics I_1', iriscode B_1' is extracted. Using fuzzy de-commitment scheme, the error correcting codeword C'' is obtained which is decoded to get C'. If $H_1(C_1) = H_1(C_1')$, the original template $B_1'' = B_1$ is recovered correctly which derives the key $K_1' = K_1$ using (15).

Algorithm 2: Authentication Phase

Input: Iris template I_1', Fingerprint template I_2', extracted iriscode B_1',
extracted minutiae set B_2',
Server: Helper data:HD_1, HD_2^*
Output: 1/0

▷ **Fuzzy commitment scheme:** outputs key K_1'
$HD_1 = (H(C_1), rand_1, \delta_1)$
$C_1'' = B_1' \oplus \delta_1$
C_1'' is decoded to C_1' if $d(B_1, B_1') < t$, ▷ t: number of errors in biometric template
if $H_1(C_1') = H_1(C_1)$
$B_1'' = C_1' \oplus \delta_1$, ▷ $B_1'' = B_1$
$K_1' = H_2(B_1'' \| rand_1)$
▷ **Format-preserving Decryption:** outputs decrypted message HD_2'
$HD_2' = FP.Dec(K_1, T, HD_2^*)$
$HD_2' = (V_2' \| (H_3(K_2))')$
▷ **Fuzzy vault scheme:** outputs the secret key K_2'
Using $V_2' = G \cup Ch$ and B_2', reconstruct the polynomial that returns key K_2'
if $(H_3(K_2') = (H_3(K_2))')$
Return 1, User is successfully authenticated

The ciphertext HD_2^* is decrypted using the format-preserving decryption function as

$$HD_2' \longleftarrow FP.Dec(K_1', T, HD_2^*)$$

On successful decryption, the vault V_2' and $(H_3(K_2))'$ are obtained. From the second input biometric characteristics I_2', the genuine feature set B_2' is generated. Using polynomial interpolation, the polynomial is reconstructed that recovers the secret key $K_2' = K_2$. The user is successfully authenticated after the key is validated by comparing the hashes of stored key and the recovered key as $H_3(K_2) = (H_3(K_2))'$.

6.3. Rationale behind the use of format-preserving encryption

Format-preserving encryption (FPE) [12] refers to encryption of data in a way such that the format of output (ciphertext) is same as the format of input (plaintext), including the length of data. For an example, if a credit card number consists of 14 digits where digits can take value 0–9, the format of the encrypted credit card after using format-preserving encryption would also remain the same, i.e., 14 digits with each digit accepting value from 0–9. Format-preserving encryption is designed for data that is not necessarily binary and can be any finite set of symbols, like the decimal numerals. Thus, FPE encrypts sensitive information and can be used for encryption in database applications which do not support changes to the format or length of data.

We use format-preserving encryption to encrypt the helper data generated by the second biocryptosystem (fuzzy commitment or fuzzy vault). The key used to encrypt the helper data is derived from the first biocryptosystem (fuzzy commitment or fuzzy vault). In the offline attack mode, assume that the attacker doesn't know any of the biometric templates B_1 or B_2. In the case when encryption is done using block cipher mode of operations such as AES-256 with CBC mode with key K_1 derived from the first biocryptosystem, it may be possible for the attacker to guess the correct key K_1 by checking the format of the decrypted message. For an incorrect key, the decrypted message would give a pseudorandom data of a fixed length as an output which can help an attacker to validate if the key is correct or not. Thus for a system with two biocryptosystems, each with security parameter let say K_1 and K_2 respectively, the overall security bound (in terms of brute force attack complexity) of the system will become $(2^{K_1} + 2^{K_2})$.

Whereas in the case of encryption of helper data with a format-preserving encryption scheme, it is impossible for an attacker to guess the correct key since the decrypted message would always be in the format same as the original plaintext, whether the key is correct or not. Thus, the attacker won't be able to distinguish a correct guess of the key K_1 from an incorrect. The best attack possible would be time-memory tradeoff attack given by security bound as shown in Section 5. The security bound is approximated as $(2^{|K_1|} \times 2^{|K_1|})$. Considering $|K_1| \approx |K_2|$, it can be observed that the bound is considerably higher than the security bound provided by the scheme without format-preserving encryption being used. An example depicting the importance of FPE over AES block cipher is shown in Appendix B.

6.4. Security analysis of S-BIOFUSE (S_3)

We analyze the privacy and security properties of S-BIOFUSE (S_3).

6.4.1. Key inversion attack

In the fuzzy vault, for successful authentication, the following equation needs to be verified: $(H_3(K_2))' = H_3(K_2')$, where $(H_3(K_2))'$ is the hash of original key stored on the server during enrolment and $H_3(K_2')$ is the hash of guessed or the recovered key during authentication. If the hash of key is stored directly on the server without encryption, the attacker can directly verify the match between both the hash values by brute force attack on key and can get unauthorized access to the system, without the need of any of the biometric templates. It is known as the key inversion attack. In BIOFUSE, for successful authentication, the user has to decrypt the encrypted message HD_2^* first using format-preserving encryption scheme which further requires cryptographic key K_1 to be generated from the first biometric template shown in (15). Using the decrypted message HD_2, the hash of key $(H_3(K_2))'$ is retrieved. Thus, the key inversion attack is difficult to achieve in our proposed scheme.

6.4.2. Blend substitution attack

In the blend substitution attack [38], the attacker modifies some of the points in the fuzzy vault in multiple ways. It can be done either by removing genuine or chaff points from the vault or by adding some of its genuine points. If the attacker can add a sufficient number of its feature points to the vault, it can get access to the system without affecting the authentication of a genuine user. Thus, the secret key can be revealed to the attacker. In case if an attacker can replace most of the genuine points of the legitimate user with its points, the genuine user won't be able to access the vault.

In S-BIOFUSE, to attack the system via blend substitution attack, the attacker has first to decrypt the encrypted vault using format-preserving encryption scheme which needs key K_1 obtained from the first biometric template. Thus, encryption of vault in BIOFUSE makes blend substitution attack difficult.

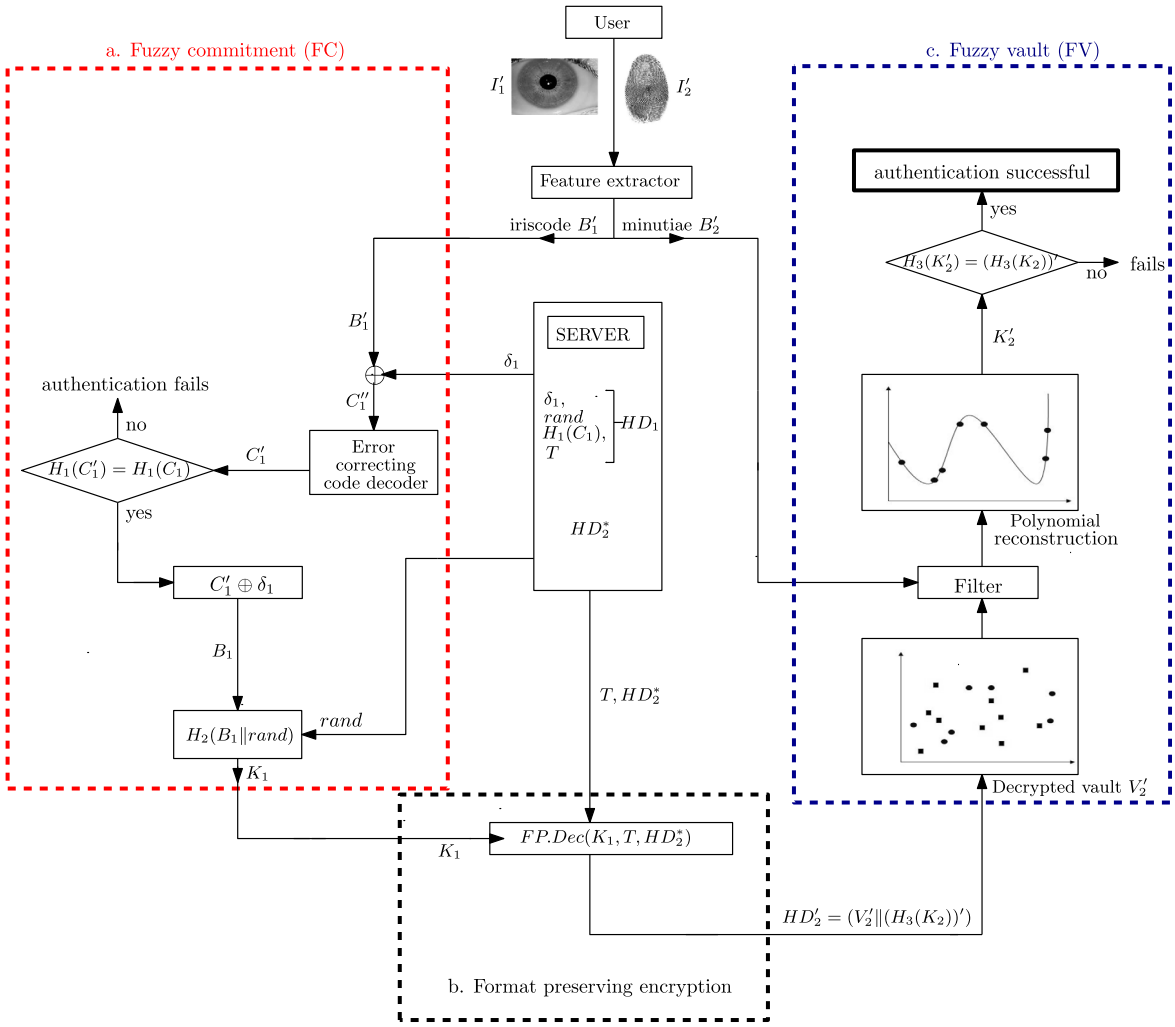


Fig. 9. Authentication phase. a. Fuzzy commitment scheme takes input B_1' from the user and using δ_1 stored, it generates the codeword C_1'' decoded to C_1' if the user is genuine. The key K_1 is generated by recovering original B_1. **b.** The key K_1 is used to decrypt HD_2^* using FPE scheme to get the original helper data as $HD_2' = V_2' \| (H_3(K_2))'$ **c.** Using V_2' and query minutiae point's set B_2' obtained from I_2', the secret key K_2' is derived if sufficient number of query points matches with the vault points. A user is authenticated if $H_3(K_2')$ is equal to $(H_3(K_2))'$.

6.4.3. Correlation attack

In multiple vaults of the same user, genuine points correlate well that violates unlinkability. It results in a correlation attack in which an attacker can detect genuine correlated points from multiple vaults and thus separates the chaff points. In some techniques [4], a password is used to transform the genuine points by permuting them so that correlation cannot be found in multiple vaults or chaff points are deterministic [38] which is an overhead. In S-BIOFUSE, no additional security parameter has been used to provide unlinkability.

Let V_1 be the vault generated for one particular application. Using the user's biometric template, T and key K_1 values are obtained from (13) and (15) and are used to encrypt the vault V_1 through format-preserving encryption scheme. In the case when multiple vaults $\{V_2, V_3, \ldots V_x\}$ of the same user are needed for multiple applications, the K_1 and T generated for each application from the same user will be completely different for different applications due to similar but not identical biometric characteristics. Since different vaults of the same user are encrypted by different values of T and K_1, it is not possible to find a correlation between these vaults across different applications.

7. Experiments and results

We perform several experiments for fingerprints, iris and face characteristics on the publicly available databases as given in Table 3. The example images are shown in Fig. 10. To create a virtual multi-modal database, considering one-to-one correspondence, we combined the samples from the two different instances. We deleted the extra samples if any. We consider

Table 3
Iris and face database description.

Modalities	Database(s)	Subjects	Samples	Resolution
Fingerprint	FVC2002- DB1,DB3,DB3 [29]	100	8	500×500[1]
Iris	IITD [26]	448	5	320×240
Iris	CASIA-Iris-Interval[2]	337	5	640×480
Face	XM2VTSDB (CDS001) [39]	295	4	720×576

[1] Enhanced resolution
[2] http://biometrics.idealtest.org/

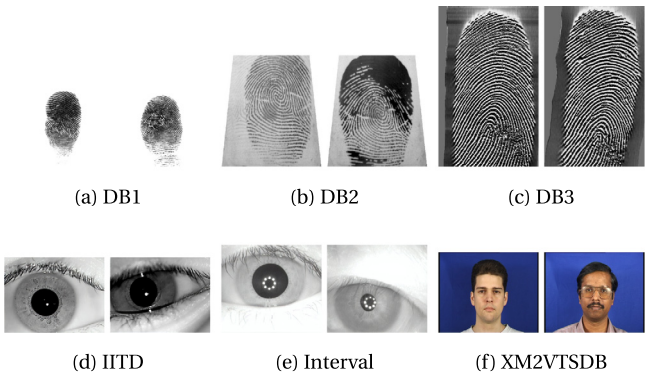


Fig. 10. Example images from the selected databases.

70 subjects from each combined dataset as genuine subjects and 30 subjects as impostors, with left and right samples treated as mutually independent samples. The first biometric characteristic is given as input to the fuzzy commitment scheme, whereas, the second characteristic is fed to the fuzzy vault scheme.

We use open source libraries and software for iris and face feature extraction. For iriscode generation, we use OSIRIS [40] and University of Salzburg Iris Toolkit v1.0 [43]. We perform the feature extraction using Daugman-like 1D-Log Gabor (LG) algorithm proposed by Masek [30] to generate iriscodes of size $512 \times 20 = 10240$ bits. For face features extraction, we use the FaceRecLib of the free signal and image processing toolbox Bob[3] [2,3] to obtain a cropped 4×8 sub-image with $32 \times 2400 = 76800$ bits. Hamming distance is used to compare the face and iris templates. For fingerprint database, we utilize the state-of-the-art commercial-off-the-shelf (COTS) feature extractor and matcher, VeriFinger [49] (Neurotechnology).

We compute the theoretical results [27] for our experiments in order to compare our recognition performance results with other start-of-the-art approaches. For the implementation of fuzzy commitment, we suggest using the BCH code. BCH codes are simple and are suitable choice [5,28] of error correcting codes for the implementation of fuzzy extractors involved in the fuzzy commitment schemes. We select $(1023, 46, 219)$-BCH code for iriscode, which is sampled 10 times to cover 10230 bits of iriscode. Similarly, for the face template, $(1023, 46, 219)$-BCH code is sampled 75 times to generate a codeword of $76,800$ bits while ignoring the last few bits. In our work, we assume that there exists some error correcting code (beyond the scope of the paper) that can correct all the errors in the biometric templates.

Further, we consider the matching of fingerprints using the commercial matchers, which gives approximately the same performance as given by the fuzzy vault with certain parameters. Table 4 shows the parameters that we suggest and use for implementing the proposed algorithms. The format-preserving encryption scheme is implemented using the standard NIST source code[4] [12] with 256 bits key K_1 and 128 bits tweak T. We use SHA-256 as the hash function in the implementation throughout the paper. The key and tweak values are transformed from the byte array to hex strings. An example showing the format-preserving encryption on a given helper data is provided in B.

7.1. Recognition performance evaluation

Biometric recognition performance measure depicts how correctly the users in a biometric system are authenticated. We plot the ROC curve (receiver operating characteristic curve) in Fig. 11 that demonstrates the false match rates against the true-match rates to evaluate the recognition performance. The scores are converted to a common range before they can be fused, known as normalization. It is done using the min–max normalization [17]. The following cases are used as the

[3] http://idiap.github.io/bob/
[4] http://www.lib4dev.in/info/capitalone/fpe/95807844

Table 4
Parameters used for Implementation

Parameters	Value (in bits)
Size of Iriscode's template	10240
Size of Face binarized template	76800
Length of keys K_1 and K_2	256
Size of tweak T used in *FC* scheme	64–128
Size of *rand* used in *FC* scheme	128
Order of polynomial used in *FV* scheme	8–9

Table 5
Performance evaluation along with security comparison of various multi-biometric biocryptosystem approaches. TMR refers to true match rate and FMR refers to false match rate.

Approaches	Database used	TMR at 0.01 FMR	TMR at 0.1 FMR	Security (in terms of attack complexity)	Approaches	Database used	TMR at 0.01 FMR	TMR at 0.1 FMR	Security (in terms of attack complexity)
Score-level [32,11]	IITD-DB1	0.99	0.99	$2^{\lvert K_1\rvert}+2^{\lvert K_2\rvert}$	Decision-AND [47]	IITD-DB1	0.98	0.99	$2^{\lvert K_1\rvert}+2^{\lvert K_2\rvert}$
	IITD-DB2	0.99	0.99			IITD-DB2	0.98	0.99	
	IITD-DB3	0.99	0.99			IITD-DB3	0.98	0.99	
	Interval-DB1	0.99	0.99			Interval-DB1	0.94	0.98	
	Interval-DB2	0.99	0.99			Interval-DB2	0.94	0.96	
	Interval-DB3	0.99	0.99			Interval-DB3	0.94	0.97	
	Face-DB1	0.89	0.92			Face-DB1	0.71	0.81	
	Face-DB2	0.97	0.99			Face-DB2	0.70	0.79	
	Face-DB3	0.95	0.97			Face-DB3	0.71	0.80	
Decision-OR [27]	IITD-DB1	0.99	0.99	$2^{\lvert K_1\rvert}+2^{\lvert K_2\rvert}$	Proposed	IITD-DB1	**0.98**	**0.99**	$2^{\lvert K_1\rvert}\times 2^{\lvert K_1\rvert}$
	IITD-DB2	0.99	0.99			IITD-DB2	**0.98**	**0.99**	
	IITD-DB3	0.99	0.99			IITD-DB3	**0.98**	**0.99**	
	Interval-DB1	0.99	0.99			Interval-DB1	**0.94**	**0.98**	
	Interval-DB2	0.99	0.99			Interval-DB2	**0.94**	**0.96**	
	Interval-DB3	0.99	0.99			Interval-DB3	**0.94**	**0.97**	
	Face-DB1	0.96	0.98			Face-DB1	**0.71**	**0.81**	
	Face-DB2	0.99	0.99			Face-DB2	**0.70**	**0.79**	
	Face-DB3	0.95	0.99			Face-DB3	**0.71**	**0.80**	

underlying architecture for the various existing state-of-the-art and other related multi-biometric template protection approaches. In these cases, to compare the results with our proposed approach, we consider 2 biometric characteristics, with a fuzzy commitment and a fuzzy vault scheme in order to get consistency in the performance measure.

- **Score-Level:** In the score level fusion approach [32,11], the scores from individual biometric characteristics are summed up to generate a final score value (using sum-rule [17]).
- **Decision-OR:** In the decision-OR fusion [27], a Boolean OR operation is performed between the scores of individual components to generate a final decision value.
- **Decision-AND:** In the decision-AND fusion [47], a Boolean AND operation is performed between the scores of individual components to generate a final decision value.

Our proposed scheme (given as Proposed case) is compared with these three cases. The performance accuracy, along with security comparisons, is given in Table 5 computed from the Fig. 11.

Proposed: Our proposed scheme is based on a decision level fusion where a Boolean AND operation is performed between the scores of individual components to generate a final decision. S-BIOFUSE ensures that the user would be authenticated when both biometric characteristics provided by the user are matched, given a threshold. The format-preserving encryption is used in the approach. The performance measure for the proposed scheme would be the same as the performance of the Decision-AND approach.

Following are the observations:

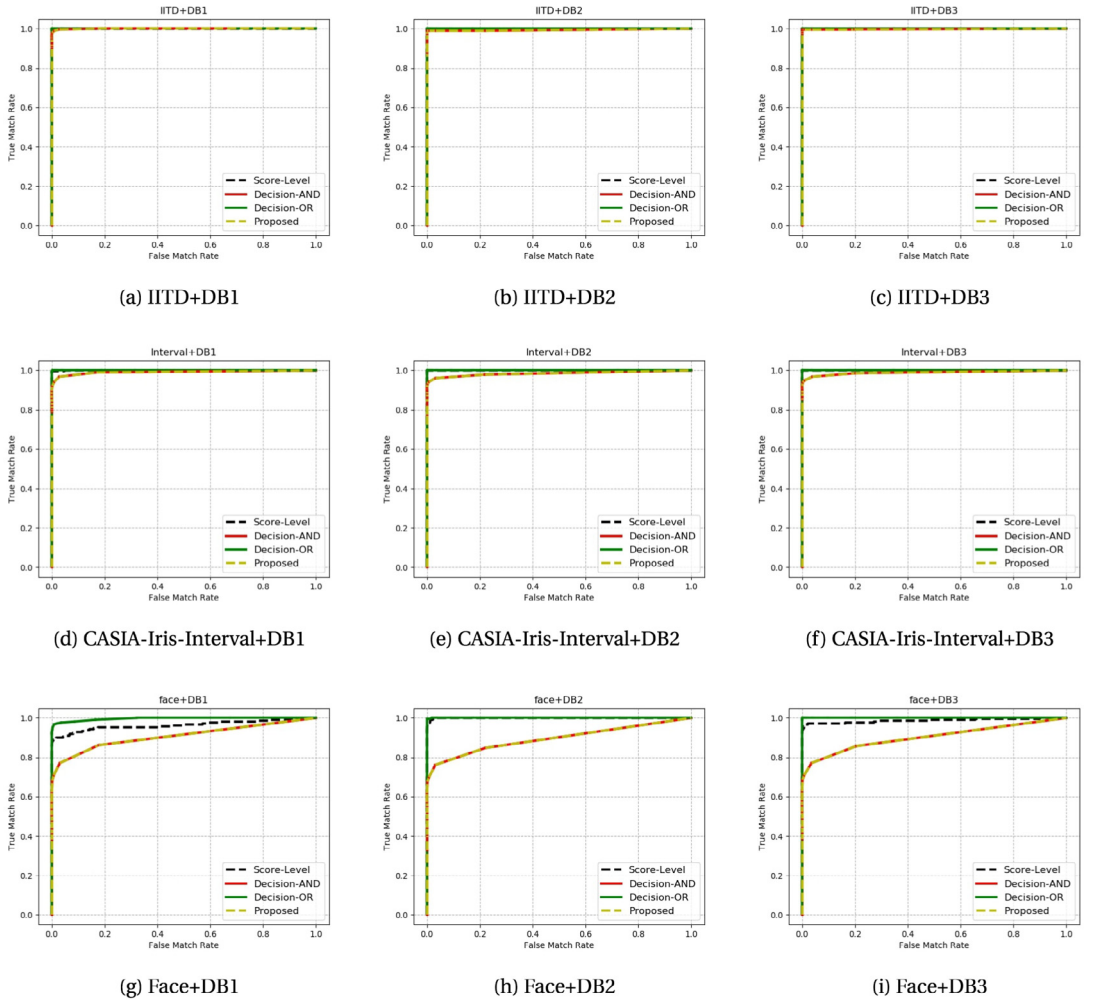


Fig. 11. Recognition performance evaluation for multi-biometric databases (a) IITD + DB1 (b) IITD + DB2 (c) IITD + DB3 (d) CASIA-Iris-Interval + DB1 (e) CASIA-Iris-Interval + DB2 (f) CASIA-Iris-Interval + DB3 (g) Face + DB1 (h) Face + DB2 (i) Face + DB3. Refer web version to interpret colors in figure legends.

- Our proposed approach gives the recognition performance in terms of true match rate equal to 0.98 or 98% on the IITD-DB1 virtual database.
- The decision-level fusion with a Boolean OR operation gives the best performance (0.99) among all the cases. It is because of the underlying OR operation, which allows the user to get authentication even if one of the two biometric characteristics is correctly matched. However, the security of such a scheme relies only on one of the biocryptosystem involved, which gives the security of $min(|K_1|, |K_2|)$ bits. Hence, it is highly insecure and is not preferred in the security-sensitive scenarios. Further, we observe that the security of score-level fusion is similar to the security of decision level fusion with OR operation.
- In general, the decision-level fusion with a Boolean AND operation provides low-performance accuracy, when compared to others. It is due to the AND operation involved, which is a strict criterion for authentication. However, the accuracy varies with databases and the matching approach used.
- The performance of Decision-AND case and our proposed scheme is the same due to the underlying decision level fusion with AND operation in both cases. The difference between these two schemes lies in the security of the schemes. Our proposed scheme is implemented with the format-preserving encryption scheme which ensures the security of $(|K_1| + |K_2|)$ bits. Whereas, the Decision-AND scheme with no format-preserving encryption in the existing state-of-the-art approaches would result in weaker security, as mentioned in the design rationale in Section 6.
- In Table 5, the virtual database with iris and fingerprint characteristics gives a high true match rate of 0.94 and above at 0.01 FMR for all the 4 cases.

- The face database combined with fingerprint gives true match rate of about 0.70 or 71% for our proposed approach. The accuracy could be certainly increased by using commercial-off-the-shelf tools.

As inferred from the related work, in the existing approaches- score level [32,11], decision-level [27] AND/OR approaches, if one of the involved biocryptosystem is compromised, the security of the whole system is compromised. Hence, the security in all these 3 cases would be given in terms of attack complexity as $2^{|K_1|} + 2^{|K_2|}$. K_1 and K_2 are the security parameters of the two biocryptosystems, respectively. In our proposed scheme, the decision-level fusion with AND operation is implemented along with format-preserving encryption scheme, which makes the attack complexity equal to $2^{|K_1|} \times 2^{|K_1|}$ which is equivalent to the desired security bound $2^{|K_1|} \times 2^{|K_2|}$ in the case when $|K_1| = |K_2|$. Thus, the proposed scheme implemented with decision-level fusion with AND operation, combined with format-preserving encryption provides the highest security level with significantly good performance accuracy.

8. Conclusions and future work

We proposed a generic, biocryptosystem level fusion framework known as BIOFUSE for the design of a multi-biometric cryptosystem that protects multiple biometric templates of a user. The two most popular biocryptosystems- fuzzy commitment and fuzzy vault are fused at the biocryptosystem level using a cryptographic primitive known as format-preserving encryption scheme. To get unauthorized access to the system, an attacker needs to impersonate all the input multi-biometric templates simultaneously, which is highly improbable. No additional security parameter is used for the construction of BIOFUSE. On comparing the recognition performance of our proposed scheme with existing multi-biometric cryptosystems, we observe 0.98 true match rate at 0.01 false match rate on a virtual IITD-DB1 database. We thoroughly analyze the security of all the constructions of BIOFUSE. Even though the security provided by all the constructions is comparable, we deduce that only one construction named as S-BIOFUSE (S_3) is the most reliable and secure. S-BIOFUSE achieves the security bound $\approx 2^{2\times|K_1|}$ which is similar to the desired security level ($2^{|K_1|+|K_2|}$) for a system with 2 biocryptosystems, each with key K_1 and K_2 respectively. BIOFUSE is not limited to any particular biometric characteristics or biocryptosystem and can be scaled accordingly for multiple application scenarios. S-BIOFUSE mitigates the significant attacks on existing fuzzy vault scheme such as blend substitution attack, brute force attack, key inversion attack and correlation attack.

While the main aim of our work is to provide a thorough analysis of security provided by multi-biometric cryptosystems, the experimental results of our proposed scheme on various multi-biometric databases show significant comparable recognition performance accuracy as compared to the existing multi-biometric cryptosystems. It shows that S-BIOFUSE provides high security along with the good accuracy and reliability to the biometric systems. As a part of future work, we would like to work on the efficient implementation of error correcting codes for iris and face databases to increase the overall performance of the underlying biocryptosystems.

CRediT authorship contribution statement

Donghoon Chang: Supervision. **Surabhi Garg:** Conceptualization, Methodology, Investigation, Writing - original draft. **Mohona Ghosh:** Methodology, Validation, Writing - review & editing. **Munawar Hasan:** Methodology, Validation, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. All possible combinations of fuzzy commitment and fuzzy vault

We found a total of 84 combinations in which a fuzzy commitment scheme can be combined with a fuzzy vault scheme. For simplicity, we considered only the most basic ways of combination. We take the input and output parameters of fuzzy commitment and fuzzy vault scheme, as shown in Fig. 1. We then construct all the possible cases by performing XOR, concatenation operation and encryption function between chosen two parameters; one parameter from each biocryptosystem.

Following are the observations from Table A.6:

- Helper data generated from the fuzzy vault or fuzzy commitment scheme or the biometric template that denotes the biometric features in a set form (we denote it as B_{FV}), for an example- minutiae points, cannot be XORed or concatenated with another helper data, key or a biometric template due to the difference in formats. Similarly, other parameters which are of different formats cannot be combined using concatenation or XOR operations. A symbol **X** shows these cases.

Table A.6
All possible 84 cases in which 2 biocryptosystems can be combined. *FC* and *FV* in subscripts represent that the particular parameter belongs to fuzzy commitment or fuzzy vault respectively. $\oplus$ represents XOR operation, $\|$ denotes concatenation and E_K represents encryption by block cipher modes of operation using a key K. **X** and ✓denotes if the case is not possible or possible, respectively. The meanings of symbols a and b are discussed in the section below.

Choice of first biocryptosystem	Choice of operation	Choice of second biocryptosystem					
		B_{FC}	HD_{FC}	K_{FC}	B_{FV}	K_{FV}	HD_{FV}
B_{FC}	$\oplus$	a	**X**	a	**X**	**X**	**X**
	$\|$	a	**X**	a	**X**	**X**	**X**
HD_{FC}	$\oplus$	**X**	**X**	**X**	**X**	**X**	**X**
	$\|$	**X**	**X**	**X**	**X**	**X**	**X**
K_{FC}	$\oplus$	**X**	**X**	a	**X**	**X**	**X**
	$\|$	**X**	**X**	a	**X**	**X**	**X**
	$E_{K_{FC}}$	b	✓(S_1)	a	b	**X**	✓(S_3)
B_{FV}	$\oplus$	**X**	**X**	**X**	**X**	**X**	**X**
	$\|$	**X**	**X**	**X**	**X**	**X**	**X**
K_{FV}	$\oplus$	**X**	**X**	a	**X**	**X**	**X**
	$\|$	**X**	**X**	a	**X**	**X**	**X**
	$E_{K_{FV}}(.)$	b	✓(S_4)	a	b	**X**	✓(S_2)
HD_{FV}	$\oplus$	**X**	**X**	**X**	**X**	**X**	**X**
	$\|$	**X**	**X**	**X**	**X**	**X**	**X**

- Biometric template in the binary string format (we denote it as B_{FC}), for example- iriscode or face can be concatenated with another similar format template or the key generated from the fuzzy commitment scheme (we denote it as K_{FC}). However, it provides no extra security to the system. Such cases are represented by symbol a.
- XOR or concatenation operation between the keys that belong to two biocryptosystems or encryption of one key using the other provides no extra security since the transformed key after XOR or encryption has no role in the authentication. Such cases are represented by symbol a. Note that we consider the key used in fuzzy vault scheme (we denote it as K_{FV}), as the system's generated internal key; hence it cannot be transformed by any of the operations.
- It is not possible to encrypt any of the biometric template using the keys since encryption on similar but not identical biometric template during authentication would completely change the data, leading to authentication failure. We represent such cases by symbol b.
- The four possible cases (S_i) shown by ✓symbol are described in Section 5.

Appendix B. An example of the proposed algorithm: S-BIOFUSE (S_3 : *FC*-then-*FV*)

In Table B.7, we provide an example depicting the working of our proposed scheme. We use iriscode and a fingerprint sample in the example. The length of tweak T and *rand* values is 128 bits (by omitting the extra bits from the 256 bits hash output). The key size is equal to 256 bits for both K_1 and K_2.

Following is an example showing the encryption and decryption of HD_2 using format-preserving encryption (FPE). The implementation details are given in Section 7. Further, we compare FPE with a block cipher based encryption algorithm-AES 256 with output in $GF(2^8)$.

K_1=39*c*9*ff*0*d*1*ffd*9640*da*47*eb*638*fb*1*d*5*c*8295413*cb*8*be*8*fc*053*b*63*c*2*a*3*b*232*e*4*ea*,

T= *bca*574*d*6773*fd*57*d*845*fcf*271*cc*69830.

The helper data contains vault points in the numeric form (with radix = 10) and the hash of the key, $H_3(K_2)$ in the form of a binary string and is given as,

$HD_2 = (V_2 \| H_3(K_2))$= 90 424 67 693 ...13281 369 315 13 $\|$ 00100 ...10000.

For simplifying the implementation of FPE, we encrypt the numeric data (vault points) and binary data (hash of key K_2) individually, while ignoring the white spaces between the vault points.

Case-1: Using FPE with correct key K_1

By applying format-preserving encryption on HD_2, we get

HD_2^*= 46 131 10 508 ...90573 163 008 23 $\|$ 11000 ...01101.

Note that, FPE preserves the length and format of the helper data, which further means that the numeric value is encrypted to a numeric format and the binary string is encrypted to generate a corresponding encrypted binary string. Also, the length of the original helper data and the encrypted helper data is the same. For example, the first vault point, 90 is encrypted to 37, both are numeric and have the same length equals to 2. The decrypted helper data is obtained by using format-preserving decryption with key K_1. We can parse the decrypted helper data according to the length of vault points and binary string respectively as in the original helper data. It is given as.

HD_2'= 90 424 67 6 ...13281 369 315 13 $\|$ 00100 ...10000.

Note, the length of data in HD_2 is public and does not reveal any significant information about the helper data itself.

$HD_2' = HD_2$ since the key is correct.

Table B.7
Examples for S-BIOFUSE.

Enrolment Input: I_1, I_2,	**Authentication Input:** I_1', I_2',
$B_1 = 1111111100 \ldots..1100000111$,	$B_1' = 1111100000 \ldots..1100000111$,
B_2= [(90 424 67 6), (93 450 56 19), ...]	B_2'= [(98 259 56 15), (98 250 247 14), ...]
Output: HD_1, HD_2^*	**Server:** HD_1, HD_2^*
▷ **Fuzzy commitment scheme:**	**Output:** 1/0
outputs $HD_1 = (H_1(C_1), rand_1, \delta_1, T)$	▷ **Fuzzy commitment scheme:**
$\delta_1 = B_1 \oplus C_1$ =1101110100 ... 0000110001	outputs key K_1'
$H_1(C_1) = 0xf7cadb0f \ldots a080b3dc$	Using HD_1 and given input B_1',
$T = 0xbca57 \ldots 9830$	C_1'' is decoded to C_1', we get
$rand_1 = 0x67c7 \ldots 7f07$	$B_1'' = B_1 = 1111111100 \ldots..1100000111$
$K_1 = 0x39c9ff \ldots 232e4ea$	$K_1' = 0x39c9ff \ldots 232e4ea$
▷ **Fuzzy vault scheme:**	▷ **Format-preserving Decryption:**
outputs $HD_2 = V_2 \| H_3(K_2)$	outputs decrypted message HD_2'
From minutiae points B_2, generate vault	$HD_2' = FP.Dec(K_1, T, HD_2^*)$
V_2= 90 424 67 6 ...13281 369 315 13	=90 424 67 6 ...13281 369 315 13 || 00100 ...10000
$H_3(K_2)$ = 00100 ...10000	▷ **Fuzzy vault scheme:**
▷ **Format-preserving Encryption:**	outputs the secret key K_2'
outputs encrypted message HD_2^*	reconstruct the polynomial to return K_2'
$HD_2^* = FP.Enc(K_1, T, HD_2)$	$(H_3(K_2') = (H_3(K_2))')$
= 46 131 10 508 ...90573 163 008 23 || 11000 ...01101	= 00100 ...10000
	Return 1, User is successfully authenticated

Case-2: Using format-preserving decryption with the incorrect key $K_1' \neq K_1$

Let K_1'= **123d***ff*0*d*1*ffd*9640*da*47*eb*638*fb*1*d*5*c*8295413*cb*8*be*8*fc*053*b*63*c*2*a*3*b*232*e*4*ea*.

The decrypted helper data is given as,

HD_2'= 37 639 60 4 ...76899 643 657 41 || 11111 ...10011.

$HD_2' \neq HD_2$. However, it has the same format and length (vault points in numeric form and hash of key as binary string) as that of the original helper data HD_2.

Case-3: The encryption and decryption of the helper data is done using the AES-256 algorithm

We use the same values of helper data HD_1 and keys K_1 and K_1' as used in Case-1 and Case-2.

Encryption of HD_2 using key K_1 will give,

HD_2^*=9*f*2314093*bb* ... *fbbdf*47*ab*918*f*7*b*989||57930*aec*14 ... *d*7*ed*3*d*4842*fb*

The decryption of HD_2^* with the correct key K_1 will generate $HD_2' = HD_2$ after hex to ASCII code conversion.

It is given as,

HD_2'= 90 424 67 6 ...13281 369 315 13 || 00100 ...10000.

Whereas, the decryption of HD_2^* with the incorrect key K_1' will give helper data as,

HD_2'= 5*b*659*ac*446*c*2 ... 850*bdb*4*b*6*c*62*b*5*b*6||27*afe*3*b*4*df* ... 13888*bf*9*c*871.

that gives a pseudorandom (gibberish) data when hex to ASCII code conversion is performed on it.

It can be inferred from the example mentioned above that in case of AES-256 based encryption, the attacker can easily guess if the key used in decryption is correct or incorrect by checking the format (which in case of AES is a pseudorandom data) of the decrypted helper data after ASCII code conversion. In the case of format-preserving encryption, even when an incorrect key is used, the format of decrypted helper data is the same as the format of original helper data. It ensures that the attacker is not able to perform brute force on the key K_1. Thus, FPE helps in efficiently encrypting and decrypting the helper data; even though the helper data constitute the values of different formats.

References

[1] 24722, I. Information technology – Biometrics – Multimodal and other multibiometric fusion, Technical Report, International Organization for Standardization, 2015.

[2] A. Anjos, M. Günther, T. de Freitas Pereira, P. Korshunov, A. Mohammadi, S. Marcel, Continuously reproducing toolchains in pattern recognition and machine learning experiments, in: International Conference on Machine Learning (ICML), 2017, http://publications.idiap.ch/downloads/papers/2017/Anjos_ICML2017-2_2017.pdf.

[3] A. Anjos, L.E. Shafey, R. Wallace, M. Günther, C. McCool, S. Marcel, Bob: a free signal processing and machine learning toolbox for researchers, in: 20th ACM Conference on Multimedia Systems (ACMMM), Nara, Japan, 2012.

[4] F. Benhammadi, K.B. Bey, Password hardened fuzzy vault for fingerprint authentication system, Image and Vision Computing 32 (2014) 487–496.

[5] R.C. Bose, D.K. Ray-Chaudhuri, On a class of error correcting binary group codes, Information and Control 3 (1960) 68–79.

[6] J. Bringer, H. Chabanne, B. Kindarji, The best of both worlds: Applying secure sketches to cancelable biometrics, Science of Computer Programming 74 (2008) 43–51.

[7] D. Chang, S. Garg, M. Hasan, S. Mishra, Cancelable multi-biometric approach using fuzzy extractor and novel bit-wise encryption, IEEE Transactions on Information Forensics and Security (2020).

[8] S. Cimato, M. Gamassi, V. Piuri, R. Sassi, F. Scotti, 2008. Privacy-aware biometrics: Design and implementation of a multimodal verification system, in: Computer Security Applications Conference, 2008. ACSAC 2008. Annual, IEEE, 2008, pp. 130–139.

[9] J. Daugman, The importance of being random: statistical principles of iris recognition, Pattern Recognition 36 (2003) 279–291.
[10] Y. Dodis, L. Reyzin, A. Smith, Fuzzy extractors: How to generate strong keys from biometrics and other noisy data, in: International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2004, pp. 523–540.
[11] R. Dwivedi, S. Dey, Score-level fusion for cancelable multi-biometric verification, Pattern Recognition Letters (2018).
[12] M. Dworkin, Recommendation for block cipher modes of operation: methods for formatpreserving encryption, NIST Special Publication 800 (2016) 38G.
[13] C. Fang, Q. Li, E.C. Chang, Secure sketch for multiple secrets, in: International Conference on Applied Cryptography and Network Security, Springer, 2010, pp. 367–383.
[14] M. Gomez-Barrero, C. Rathgeb, J. Galbally, C. Busch, J. Fierrez, Unlinkable and irreversible biometric template protection based on bloom filters, Information Sciences 370 (2016) 18–32.
[15] M. Gomez-Barrero, C. Rathgeb, G. Li, R. Ramachandra, J. Galbally, C. Busch, Multi-biometric template protection based on bloom filters, Information Fusion 42 (2018) 37–50.
[16] F. Hao, R. Anderson, J. Daugman, Combining crypto with biometrics effectively, IEEE Transactions on Computers 55 (2006) 1081–1088.
[17] M. He, S.J. Horng, P. Fan, R.S. Run, R.J. Chen, J.L. Lai, M.K. Khan, K.O. Sentosa, Performance evaluation of score level fusion in multimodal biometric systems, Pattern Recognition 43 (2010) 1789–1800.
[18] A. Jain, L. Hong, On-line fingerprint verification, in: Pattern Recognition, 1996, Proceedings of the 13th International Conference on, IEEE, 1996, pp. 596–600.
[19] A.K. Jain, K. Nandakumar, A. Nagar, Biometric template security, EURASIP Journal on Advances in Signal Processing 2008 (2008) 113.
[20] A.K. Jain, A. Ross, Multibiometric systems, Communications of the ACM 47 (2004) 34–40.
[21] A.K. Jain, A. Ross, S. Prabhakar, An introduction to biometric recognition, IEEE Transactions on Circuits and Systems for Video Technology 14 (2004) 4–20.
[22] A. Juels, M. Sudan, A fuzzy vault scheme, in: Information Theory, 2002. Proceedings. 2002 IEEE International Symposium on, IEEE, 2002, p. 408.
[23] A. Juels, M. Wattenberg, A fuzzy commitment scheme, in: Proceedings of the 6th ACM Conference on Computer and Communications Security, ACM, 1999, pp. 28–36.
[24] S. Kanade, D. Camara, E. Krichen, D. Petrovska-Delacrétaz, B. Dorizzi, Three factor scheme for biometric-based cryptographic key regeneration using iris, in: Biometrics Symposium, 2008. BSYM'08, IEEE, 2008, pp. 59–64.
[25] S. Kanade, D. Petrovska-Delacr°etaz, B. Dorizzi, Obtaining cryptographic keys using feature level fusion of iris and face biometrics for secure user authentication, in: Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on IEEE, 2010, pp. 138–145.
[26] A. Kumar, A. Passi, Comparison and combination of iris matchers for reliable personal authentication, Pattern Recognition 43 (2010) 1016–1026.
[27] C. Li, J. Hu, J. Pieprzyk, W. Susilo, A new biocryptosystem-oriented security analysis framework and implementation of multibiometric cryptosystems based on decision level fusion, IEEE Transactions on Information Forensics and Security 10 (2015) 1193–1206.
[28] F.J. MacWilliams, N.J.A. Sloane, The Theory of Error-Correcting Codes, vol. 16, Elsevier, 1977.
[29] D. Maltoni, D. Maio, A.K. Jain, S. Prabhakar, Handbook of Fingerprint Recognition, Springer Science & Business Media, 2009.
[30] L. Masek, Recognition of human iris patterns for biometric identification, 2003.
[31] P. Mihailescu, The fuzzy vault for fingerprints is vulnerable to brute force attack, 2007, arXiv preprint arXiv:0708.2974.
[32] A. Nagar, K. Nandakumar, A.K. Jain, Multibiometric cryptosystems based on feature-level fusion, IEEE Transactions on Information Forensics and Security 7 (2012) 255–268.
[33] K. Nandakumar, A.K. Jain, Multibiometric template security using fuzzy vault, in: Biometrics: Theory, Applications and Systems, 2008. BTAS 2008. 2nd IEEE International Conference on, IEEE, 2008, pp. 1–6.
[34] K. Nandakumar, A.K. Jain, S. Pankanti, Fingerprint-based fuzzy vault: Implementation and performance, IEEE Transactions on Information Forensics and Security 2 (2007) 744–757.
[35] K. Nandakumar, A.K. Jain, Biometric template protection: Bridging the performance gap between theory and practice, IEEE Signal Processing Magazine 32 (2015) 88–100.
[36] I. Natgunanathan, A. Mehmood, Y. Xiang, G. Beliakov, J. Yearwood, Protection of privacy in biometric data, IEEE Access (2016).
[37] I. Natgunanathan, A. Mehmood, Y. Xiang, G. Hua, G. Li, S. Bangay, An overview of protection of privacy in multibiometrics, Multimedia Tools and Applications 77 (2018) 6753–6773.
[38] M.T. Nguyen, Q.H. Truong, T.K. Dang, Enhance fuzzy vault security using nonrandom chaff point generator, Information Processing Letters 116 (2016) 53–64.
[39] J. Ortega-Garcia, J. Fierrez, F. Alonso-Fernandez, J. Galbally, M.R. Freire, J. Gonzalez-Rodriguez, C. Garcia-Mateo, J.L. Alba-Castro, E. Gonzalez-Agulla, E. Otero-Muras, et al, The multiscenario multienvironment biosecure multimodal database (bmdb), IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (2009) 1097–1111.
[40] N. Othman, B. Dorizzi, S. Garcia-Salicetti, Osiris: An open source iris recognition software, Pattern Recognition Letters 82 (2016) 124–131.
[41] V.M. Patel, N.K. Ratha, R. Chellappa, Cancelable biometrics: A review, IEEE Signal Processing Magazine 32 (2015) 54–65.
[42] C. Rathge, A. Uhl, P. Wild, Reliability-balanced feature level fusion for fuzzy commitment scheme, in: Biometrics (IJCB), 2011 International Joint Conference on IEEE, 2011, pp. 1–7.
[43] C. Rathgeb, A. Uhl, P. Wild, H. Hofbauer, Design decisions for an iris recognition sdk, Handbook of Iris Recognition, Springer, 2016, pp. 359–396.
[44] G.D.P. Regulation, Regulation (eu) 2016/679 of the european parliament and of the council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46, Official Journal of the European Union (OJ) 59 (2016) 294.
[45] M. Sandhya, M.V. Prasad, Biometric template protection: A systematic literature review of approaches and modalities, in: Biometric Security and Privacy, Springer, 2017, pp. 323–370.
[46] A. Stoianov, Security of error correcting code for biometric encryption, in: Privacy Security and Trust (PST). 2010 Eighth Annual International Conference on IEEE, 2010, pp. 231–235.
[47] M. Sudhamani, M. Venkatesha, K. Radhika, Fusion at decision level in multimodal biometric authentication system using iris and finger vein with novel feature extraction, in: 2014 Annual IEEE India Conference (INDICON), IEEE, 2014, pp. 1–6.
[48] A.B.J. Teoh, J. Kim, Error correction codes for biometric cryptosystem: An overview 32 (2015) 39–49.
[49] VeriFinger, Neurotechnology verifinger sdk, version: 11.2. www.neurotechnology.com/verifinger.html, Accessed: May 1, 2020.
[50] K. Xi, J. Hu, Bio-cryptography, Handbook of Information and Communication Security (2010) 129–157.