Cancelable Multi-Biometric Approach Using Fuzzy Extractor and Novel Bit-Wise Encryption

Donghoon Chang, Surabhi Garg, Munawar Hasan, and Sweta Mishra

***Abstract*— The widespread deployment of multi-biometrics to authenticate users prompts the need for biometric systems with high recognition performance. Further, the biometric data, once leaked or stolen, remains compromised forever. Hence biometric security is of utmost importance. Existing biometric template protection schemes either degrade the recognition performance or they have issues with security and speed. We propose a cancelable multi-biometric authentication approach where a novel bit-wise encryption scheme transforms a biometric template to a protected template using a secret key generated from another biometric template. It fully preserves the number of bit-errors in the original and the protected template to ensure recognition performance equivalent to the performance of the unprotected systems. We introduce Algorithm I and Algorithm II for bit-wise encryption; both are defined over cryptographic-primitives- block cipher based encryption and keyed-hash function. We profile these algorithms on various hardware architectures to calculate the efficiency in terms of the time taken during enrolment and authentication phase. For Algorithm II, we observe that a 3.3 GHz desktop architecture takes about 18 milliseconds on an average of over 200 runs to authenticate a user. Additionally, we provide mathematical proof to show that the proposed scheme guarantees secrecy and irreversibility. The results of comparisons with the existing biometric template protection schemes on the various face and iris databases show that the proposed work provides significantly good recognition performance and efficiency, while it achieves high security. Finally, the bit-wise encryption scheme can be built over the commercial-off-the-shelf systems to achieve security with equivalent high performance.**

***Index Terms*— Multi-biometrics, cancelable biometric, bit-wise encryption, biometric security, fuzzy extractor.**

I. Introduction

THE advancement of biometrics in the widespread deployment of authentication systems such as India's Aadhaar project [1] emphasizes the need for the biometric systems with high recognition performance along with the protection of biometric data referred as biometric templates. Several multi-biometric systems have been proposed in the literature [2]–[5] that take multiple biometric templates as input for authentication. Such systems improve performance accuracy by reducing the inter-class similarity as well as provide resilience against spoof attacks [6], [7]. Generally, in the existing authentication systems, the biometric templates such as iriscodes generated from iris samples [8] are stored and compared in their original, unprotected form without any transformations. This practice raises serious security and privacy concerns as the unprotected biometric template can be leaked or stolen by an attacker [9]. Further, it is possible to recover the original biometric samples from the unprotected templates [10]–[12]. Unlike passwords, it is not possible to change the biometrics; therefore, once compromised, biometrics are lost forever. European Union (EU) General Data Protection Regulation 2016/679 [13] has classified biometric data as sensitive information with access to such data subject to the right to privacy. Besides, a biometric system should satisfy several privacy properties- irreversibility, unlinkability and renewability as mentioned in ISO/IEC IS24745:2011 [14]. In addition to these properties, the biometric recognition performance measured in terms of false match rate and false non-match rate should be high. These concerns bring about the need for biometric template protection schemes that could provide high recognition performance. One of the simple approaches is to perform one-way hashing of a biometric template. Any two samples of the same biometric instance are never the same. Thus bit-errors always exist in the biometric templates, which makes hashing in biometrics infeasible.

Recently, biometric and multi-biometric template protection schemes have been introduced [15]–[18] to hide any sensitive information about the original templates. These are commonly categorized as biometric cryptosystems or biocryptosystems, cancelable biometrics and homomorphic encryption schemes. Biocryptosystems [19]–[23] transform the original biometric templates into the biometric-dependent helper data (protected template) as well as they help in generating the strong cryptographic key from biometric templates using the fuzzy extractors. Cancelable biometrics [9], [24], [25] use a key or password dependent transformation function to transform the original biometric template into a cancelable template such that the comparisons between templates are performed in the transformed domain. Compared to the unprotected (denoted as baseline) biometric systems, recognition performance degrades due to the addition of noise in the form of bit-errors during the transformation. In the homomorphic encryption schemes [4], [26], [27], the comparisons for authentication are performed on the encrypted biometric templates. Further description of these schemes with their limitations is given in Section III.

Manuscript received June 4, 2019; revised December 1, 2019 and February 23, 2020; accepted March 11, 2020. Date of publication March 25, 2020; date of current version April 21, 2020. This work was supported in part by Irisys Company, Ltd., South Korea, and in part by the Indraprastha Institute of Information Technology, New Delhi, India. The work of Surabhi Garg was supported by the Ph.D. Fellowship from TCS, India. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Julien Bringer. *(Corresponding author: Surabhi Garg.)*

Donghoon Chang and Surabhi Garg are with the Department of Computer Science and Engineering, Indraprastha Institute of Information Technology, New Delhi 110020, India (e-mail: donghoon@iiitd.ac.in; surabhig@iiitd.ac.in).

Munawar Hasan is with Irisys Company, Ltd., Seoul, South Korea, and also with the Department of Computer Science and Engineering, Indraprastha Institute of Information Technology, New Delhi 110020, India (e-mail: munawar@irisys.co.kr; munawar1440@iiitd.ac.in).

Sweta Mishra is with the National Institute of Standards and Technology, Gaithersburg, MD 20899 USA, and also with the Department of Computer Science and Engineering, Shiv Nadar University, Greater Noida 201314, India (e-mail: sweta.mishra@snu.edu.in).

Digital Object Identifier 10.1109/TIFS.2020.2983250

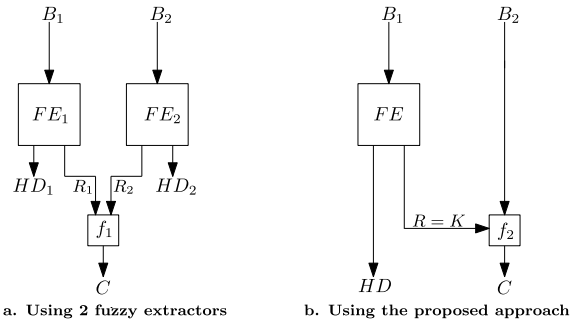


Fig. 1. Multi-biometric template protection approaches to generate the protected template C. K denotes the secret key. **a. Use of two fuzzy extractors FE_1 and FE_2:** On input as biometric template B_i, FE_i generates helper data HD_i and a secret, random string R_i where $i = 1, 2$. f_1 denotes concatenation or XOR operation. **b. Use of a single FE:** Combines FE with a transformation function f_2 (proposed bit-wise encryption) to generate $R = K$ and HD.

A. Motivation and Contributions

Multi-biometric authentication systems involve the fusion of two or more biometric instances, where fusion can be performed on feature level [2], [28]–[30], decision level [3] or score level [31], [32]. Existing fusion approaches have several limitations in terms of adaptability and security. A naive multi-biometric fusion approach using biocryptosystems would be to implement two fuzzy extractors, denoted as FE_1 and FE_2 for two input biometric templates B_1 and B_2 as shown in Fig. 1a. The major limitations of this approach are:

- The fuzzy extractors require encoding and decoding of the error correcting codewords which usually takes large computations [33] due to the complex operations involved. The approach shown in Fig. 1a with two fuzzy extractors would require twice of such complex computations.
- The helper data generated as one of the outputs of fuzzy extractor FE often leaks some linkability information about B [17]. Reusable fuzzy extractors are proposed [34], [35] to mitigate this issue. However, they require huge storage. For example, a biometric template of 1024 bits would take about 1 Terabytes of storage space to handle 25% bit errors [36]; in our case, iriscode is of 10240 bits. The implementation of two reusable fuzzy extractors with almost double storage requirement (in comparison to one reusable fuzzy extractor) would be impractical in the real-life scenarios where a huge dataset is deployed.
- If one of the biometric characteristics possess a large number of bit errors, for example, the binarized face or fingerprint template has more bit errors than the iriscode [5], it would be difficult for the fuzzy extractor to correct all the bit errors because of the limit on error correction capabilities [20].

These limitations motivate us towards another approach, as shown in Fig 1b. Biometric templates are used to generate strong, uniform random strings from the fuzzy extractors; which can be treated as the cryptographic keys [20]. Such key generated by one biometric template can be used to transform another biometric template of the same user in a multi-biometric system. The first biometric template B_1 is used to generate a random string R using a fuzzy extractor FE. The transformation function f_2 transforms the second biometric template B_2 into a protected template C, considering $R = K$. Only one fuzzy extractor is implemented, which will require comparatively low storage and fewer computations. The second biometric template can be transformed by the function f_2 in the following ways using the key K while dealing with bit errors during the transformation, given a pre-defined threshold.

1) Directly encrypt the biometric template using block cipher modes of operation [37]. But, it needs the decryption of protected template during verification which could reveal the original template to the attacker.
2) Cancelable biometrics such as Bloom filter based templates [38], [39] provide good performance and efficiency. Still, the error rates are high due to the transformation of biometric templates into the bloom filter based structures.
3) Homomorphic Encryption [4], [26] is used to encrypt the biometric template. However, it requires huge computational time to perform verification.

To mitigate the limitations of existing transformation approaches, we propose a non-invertible, bit-wise encryption scheme (that represents f_2 in Fig. 1b). Each bit of the second biometric template, B_2 is transformed into a corresponding one-bit output in the protected template C. The key derived from the first biometric template B_1 ensures that even if an attacker makes random queries to the oracle (client's device), the transformed template generated will be completely random since the secret key would be unique for each instance. In this paper, we use 2 baselines to represent the unprotected multi-biometric system: Baseline-A uses open source libraries and software for biometric recognition, and Baseline-B uses the Commercial-off-the-shelf (COTS) systems.

This paper makes the following contributions.

- We propose a cancelable multi-biometric approach for biometric authentication systems that generates a cancelable protected template from two biometric samples given as input. The cancelable template generated is unlinkable, irreversible and renewable.
- To the best to the authors' knowledge, a novel way of using a bit-wise encryption scheme is proposed to transform the original biometric template into the protected cancelable template. It completely preserves the number of errors in the protected and the original biometric template; thus provide the recognition accuracy equivalent to the unprotected (Baseline-A) system's accuracy.
- We propose two different algorithms, Algorithm I and Algorithm II to perform bit-wise encryption. The efficiency analysis of the proposed system measured in terms of time across various profiled hardware is summarized in Table IV and Table V.
- We provide the in-depth security analysis of our proposed scheme in terms of irreversibility, unlinkability and renewability. Further, the mathematical proof shows that our bit-wise encryption scheme preserves secrecy and irreversibility.
- A detailed analysis on recognition performance and efficiency (measured in time) of the proposed system along the comparisons with existing biometric template protection schemes emphasize the scope of deployment of our proposed work in real-time scenarios. Further,

our bit-wise encryption scheme can be built over COTS systems to provide equivalent high performance along with security.

Organization: The rest of the paper is organized as follows. Section II delivers the preliminaries used in the paper. Section III describes the related template protection schemes. Section IV presents the model and system settings for the proposed work. Section V explains the proposed construction of multi-biometric authentication system. Section VI provides the design rationale of the proposed construction. Implementation details and performance evaluation of the proposed scheme are discussed in Section VII. Section VIII presents the detailed security analysis of our proposed work. The conclusions are summarized in Section IX. The Appendix provides the mathematical proof and examples for our proposed algorithms.

II. Preliminaries

A. Notations

B denotes the original biometric template represented in binary format, C denotes the protected template, (x, y) are the x and y-coordinates in the biometric template that represent the bit-wise position for a particular bit as $b_{(x,y)}$, K denotes a random, cryptographic key of length k, H denotes the hash function, HD denotes the helper data generated by fuzzy extractor.

B. Definitions

We use several concepts and functions in our construction as discussed below.

- **Fuzzy Extractor (FE):** A fuzzy extractor is defined by two procedures [20]- Generation procedure, Gen on input $B \in \mathcal{M}$, outputs a random string $R \in \{0,1\}^n$ and a helper data string denoted as $HD \in \{0,1\}^*$. $\mathcal{M}$ denotes the Hamming distance metric space. Reproduction procedure, Rep takes $B' \in \mathcal{M}$ and $HD \in \{0,1\}^*$ as inputs. The correctness property of fuzzy extractors guarantees that if $\|B \oplus B'\| \leq t$, and $(R, HD) \longleftarrow Gen(B)$, then $Rep(B', HD) = R$ else there is no guarantee about the output. Here, t is the error correcting capability of the underlying error correcting codes and $\|.\|$ denotes the Hamming distance between two strings.
- **Binary p-bit mapping:** It is a p-bit binary representation of the decimal value. The binary number
$$a_{p-1}2^{p-1} + a_{p-2}2^{p-2} + \cdots + a_0$$
is denoted as $a_{p-1}a_{p-2}\cdots a_0$ where $a_i \in \{0,1\}$ and p is the number of digits to the left of the binary (radix) point. For example, $\text{binary}_{16}(val)$ denotes 16-bit binary representation of the value val.
- **byte-xor:**The byte-xor of a d-bit binary number $a_{d-1}a_{d-2}\cdots a_0$ where $a_i \in \{0,1\}$ splits the whole string into individual byte strings. These strings are byte-wise XORed to output a single byte denoted as
$$(a_{d-1}a_{d-2}\cdots a_{d-8}) \oplus (a_{d-9}\cdots a_{d-16}) \ldots \oplus (a_7a_6\cdots a_0)$$

III. Related Work

In this section, we discuss several existing biometric and multi-biometric template protection schemes. We primarily focus on template protection schemes for iris and face.

A. Biometric Cryptosystems or Biocryptosystems

Cimato *et al.* [40] proposed a modular biocryptosystem where two secure sketch schemes (as individual modules) are connected such that the key generated from the first module is XORed with the second biometric template. The above approach is generalized by Fang *et al.* [41], where multiple biometric templates are combined in a cascaded manner while deploying the secure sketch framework [20]. The security of the external module defines the overall security in the modular-level fusion scheme. In [42], authors proposed a cancelable secure sketch where the secure sketch is applied on cancelable biometric templates. The secure sketch provides secure biometric template protection, and cancelable biometric template prevents correlation attack and provides renewability property. Nagar *et al.* [29] 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. It provides high security. However, the overall performance is degraded as well as it requires adapted feature extraction tools for implementation. A decision level fusion is performed on fingerprint-based multi-biometric biocryptosystem [3]. Hash functions are used to protect each fingerprint. However, the additional key used as a security parameter needs extra security.

B. Cancelable Biometric Systems

The cancelable biometrics, particularly for iris templates, are broadly classified into the following two categories: *Salting*, and *Non-invertible transforms*. Salting adds random noise to the original biometric template. Few examples are GRAY SALT, BIN SALT [24] where noise is added to iris templates and BioEncoding [43] where consistent iriscode bits are mapped to random bit values. Another approach based on random permutation (GRAY COMBO, BIN COMBO) is proposed in [24] where the rows of iriscode are translated and concatenated according to a secure value. In the non-invertible transforms, the transformations dependent on secret parameters such as a key or a password are performed. An adaptive Bloom filters based approach [38], [39], [44] is a popular example where a hash function defined by a simple binary to integer transformation maps the columns of the biometric template to the bloom filters. The comparisons take place by calculating the Hamming distance metrics on Bloom filter arrays. Bloom filter based cancelable biometric indexing using binary tree structure has been proposed [45]–[47] to reduce biometric comparisons workload. A multi-biometric fusion based on bloom filter approach is proposed [5] for face and iris templates where the biometric templates are transformed to bloom filter based templates. The fusion is then done by an OR operation between the corresponding bloom filter arrays. Notwithstanding the fact that bloom filter-based systems satisfy unlinkability, irreversibility and renewability, the performance drop persists.

C. Homomorphic Encryption Schemes for Biometric Security

Introduced by Rivest *et al.* in 1978 [48], it provides computation over the encrypted domain. Given two encrypted messages $E_k(m_1)$ and $E_k(m_2)$, $E_k(m_1 * m_2) = E_k(m_1) *$

$E_k(m_2)$ where $*$ can be any operation. A somewhat homomorphic encryption scheme with a MAC-based approach is proposed [26] for secure biometric authentication where the server stores the encrypted biometric template. The client stores the decryption key to decrypt the comparison score sent by the server during authentication. The computation time required is large, which makes the deployment impractical. Gomez-Barrero *et al.* [4] propose a multi-biometric template protection scheme in which the fusion of signature and fingerprint is done at the feature level, score level and decision level to generate a combined template encrypted using the homomorphic encryption. The scheme is cancelable and shows good performance. However, in addition to high computation time, the secret key stored on the server is assumed to be secure. In practice, it may lead to loss of privacy in case if the secret key is leaked. A lightweight encryption scheme [49] (PassBio) is recently proposed for user-centric biometric authentication such that the comparisons are performed on the encrypted data with a threshold. The matrix multiplications involved with huge latency make the scheme considerably slower than other baseline approaches.

IV. Models and Settings

In this section, we present the architecture for the proposed system. Throughout the paper, our scheme adheres to this framework. We propose a generalized, modular framework with several components including feature extractor, fuzzy extractor, bit-wise encryption module and the matcher or the comparator module.

A. System Model and Participants

The proposed system consists of a client and a server. The client contains a device, that is capable of extracting user's biometric data like iris, face or fingerprint. Some examples of user device could be Android/ iOS powered smart-phones or a personal desktop. The server is an online entity (a service provider) which is honest but curious. Fig. 2 describes the schematic model of our proposed construction. The standard feature extraction library, fuzzy extractor and bit-wise encryption libraries are stored on the client's device. The biometric comparisons are performed on server-side.

B. Threat Model

We focus on the following attacks to ensure the privacy of user's biometric data along with the system's security.

- User-side participants can try to get the original biometric template from the protected template stored on the server. Further, they may collude arbitrary with other users of the system to achieve reversibility, an attempt to break the system's security. In the proposed setup, the client device would act as a random oracle. The adversary is allowed to perform any number of queries on the client device. The user may also try to get linkability information from the information stored on the client or server-side. Further, a client's device might come under the offline attack where the adversary can read the memory etc. of the device.
- In real-world scenarios, a server can come under two classes of attacks, passive attack (where an attacker is incapable of interacting with the server but is able to listen to the data routed to the server) and active attack (the attacker by any means is able to take control of the server and has access to the underlying database).

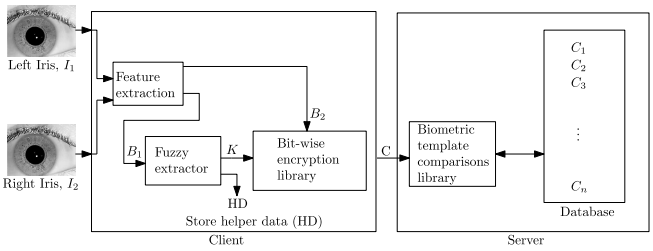


Fig. 2. Schematic model of the proposed work.

V. Proposed Work

Our scheme comprises of two steps: Key generation using fuzzy extractor and bit-wise encryption to generate a cancelable template. The key generation step is different for enrolment and authentication, whereas the bit-wise encryption step is common for both the phases.

Algorithm 1 Cancelable Biometric Template Construction Using Bit-Wise Biometric Template Expansion

Input Original right biometric template B_2 extracted from the biometric data I_2.
$B_2 = \{b_{(0,0)}, b_{(0,1)}, \ldots, b_{(u-1,v-1)}\}$, tweak T, key K
Output Protected biometric template
$C = \{c_{(0,0)}, c_{(0,1)}, \ldots, c_{(u-1,v-1)}\}$
For each bit $b_{(x,y)}$ of B:

function EXPANSION-1($b_{(x,y)}, (x, y)$)
  $binary_p(x) \longleftarrow x$
  $binary_p(y) \longleftarrow y$
  $M_{b_{(x,y)}} \longleftarrow b_{(x,y)} \| binary_p(x) \| binary_p(y)$
end function
function TRANSFORMATION($K, M_{b_{(x,y)}}, T$)
  $D_{b_{(x,y)}} \longleftarrow F(K, M_{b_{(x,y)}}, T)$ ▷ *where F can be F_H or F_E, $|D| = d$*
end function
function EXTRACTION-BIT($D_{b_{(x,y)}}, b_{(x,y)}, (x, y), T, K$)
  $M_{1-b_{(x,y)}} \longleftarrow$ EXPANSION-1($(1 - b_{(x,y)}), (x, y)$)
  $D_{(1-b_{(x,y)})} \longleftarrow$ TRANSFORMATION($K, M_{(1-b_{(x,y)})}, T$)
  $in \longleftarrow$ (byte-xor($D_{b_{(x,y)}} \oplus D_{(1-b_{(x,y)})}$))$\%d$ ▷ *selection of random index in*
  for k in (in to (in+(d−1))$\%d$) **do** ▷ Circular looping
    if $D_{b_{(x,y)}}[k] != D_{(1-b_{(x,y)})}[k]$ **then**
      $pos \longleftarrow k$
    end if
  end for
  $c_{(x,y)} \longleftarrow D_{b_{(x,y)}}[pos]$
  Return $c_{(x,y)}$
end function

A. Key Generation

During the enrolment phase, the user provides first biometric characteristics I_1 (left iris sample let say), from which a binary string B_1 is extracted [50]. A random error correcting codeword W_1 is generated by the system to handle the bit-errors

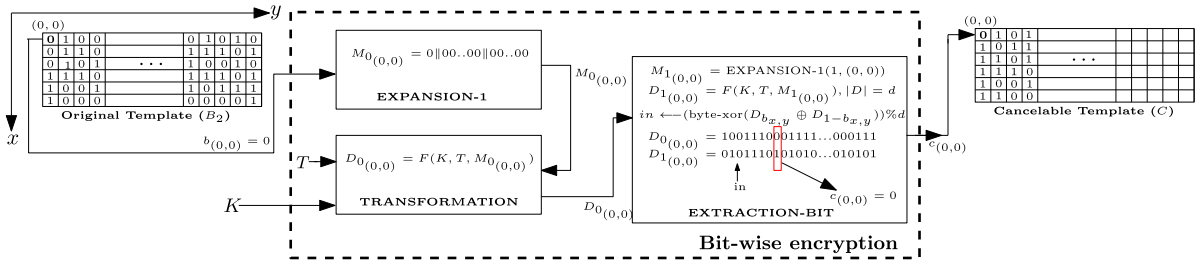


Fig. 3. Cancelable biometric template construction using bit-wise biometric template expansion. Example shows the mapping from original to protected iriscode template for a bit position $(0, 0)$.

present in the biometric template. Here $|B| = |W|$. A public helper data HD_1 is constructed as $HD_1 \longleftarrow B_1 \oplus W_1$. The cryptographic key of length k is derived by taking the hash of the codeword and is denoted as $K \longleftarrow H(W_1)$

During the authentication phase, from the similar biometric characteristics I_1', biometric template B_1' is extracted. The error correcting code W_1' is generated as $W_1' \longleftarrow B_1' \oplus HD_1$. If $\|B_1 \oplus B_1'\| \leq t$ is satisfied, W_1' is correctly decoded using t-error-correcting code to get the codeword $W_1'' = W_1$. The secret key K is recovered as $H(W_1'')$.

B. Bit-Wise Encryption

Given a secret key K derived from the fuzzy extractor during enrolment phase, a bit $b_{(x,y)}$ of the biometric template at coordinates (x, y) is encrypted using the underlying cryptographic primitives such as hash function or block cipher based encryption, to generate a random bit $c_{(x,y)}$ at the corresponding coordinates (x, y). We define this mapping of each input-bit $b_{(x,y)}$ to an output-bit $c_{(x,y)}$ as bit-wise encryption for a particular coordinate (x, y). It is given as $b_{(x,y)} \longrightarrow c_{(x,y)}$ such that if for a particular coordinates (x, y),

$$\left.\begin{array}{ll} 1_{(x,y)} \longrightarrow 0_{(x,y)}, & \text{then } 0_{(x,y)} \longrightarrow 1_{(x,y)} \\ \text{and if } 1_{(x,y)} \longrightarrow 1_{(x,y)}, & \text{then } 0_{(x,y)} \longrightarrow 0_{(x,y)} \end{array}\right\} \quad (1)$$

C. Algorithmic Explanation of the Bit-Wise Encryption Approach

The Algorithm I and Algorithm II are defined with three functions, namely, Expansion, Transformation and Extraction. Further, the Expansion function expands each input bit to some pre-defined length and can follow two different approaches denoted by Expansion-1 and Expansion-2. The expanded bits are operated with the key through Transformation function. The Extraction function can be represented as Extraction-Bit or Extraction-XOR and returns one-bit from the output of the Transformation function.

*1) **Algorithm I:** Cancelable Biometric Template Construction Using Bit-Wise Biometric Template Expansion:* Fig. 3 shows the block diagram for the proposed construction using Algorithm I. Refer Appendix B for the example of Algorithm I. Let an unprotected biometric template B_2 is extracted from the second biometric input I_2. The B_2 is represented as a two dimensional matrix with u rows and v columns, where each bit is represented as $b_{(x,y)}$, where $(0 \leq x < u)$ and $(0 \leq y < v)$. T denotes a random, public parameter known as a tweak.

1) **Function EXPANSION-1 to return** $M_{b_{(x,y)}}$
Expand the bit $b_{(x,y)}$ by concatenating the bit $b_{(x,y)}$ with the binary values of corresponding coordinates (x, y) to get an expanded bit string $M_{b_{(x,y)}}$. The binary values are obtained by using Binary p-bit mapping function defined in Section II.

2) **Function TRANSFORMATION to return** $D_{b_{(x,y)}}$
For every expanded bit string $M_{b_{(x,y)}}$, apply a key-based transformation function F on it as discussed in the following cases.
Case 1: $F =$ Encryption function, F_E: Block cipher modes of operation using a suitable mode [37] is used with the help of a key K. An arbitrary length tweak is concatenated with the expanded bit-string message bit $M_{b_{(x,y)}}$. The output of transformation is given as $D_{b_{(x,y)}}$ of length d as

$$D_{b_{(x,y)}} \longleftarrow F_E(K, (M_{b_{(x,y)}} \| T))$$

Case 2: $F =$ keyed-hash, F_H: Transformation function is the keyed-hash which involves a cryptographic hash function and the key K with the output given as

$$D_{b_{(x,y)}} \longleftarrow F_H(K, M_{b_{(x,y)}} \| T)$$

3) **Function EXTRACTION-BIT to return** $c_{(x,y)}$ Perform EXPANSION-1 and TRANSFORMATION functions on the complement bit $(1 - b_{(x,y)})$ to derive a corresponding transformed bit string $D_{(1-b_{(x,y)})}$ of length d. Select a random index. There are two ways to get the index value:

- the first position of bit string is taken as index, i.e. $in \longleftarrow 0$
- A random index position given as

$$in \longleftarrow (\text{byte-xor}(D_{b_{(x,y)}} \oplus D_{(1-b_{(x,y)})}))\%d \quad (2)$$

Starting from the selected index in, perform a circular lookup in both the transformed bit strings obtained from $b_{(x,y)}$ and $(1 - b_{(x,y)})$ to find the first position pos with dissimilar bits such that $D_{b_{(x,y)}}[k]! = D_{(1-b_{(x,y)})}[k]$. The corresponding bit $D_{b_{(x,y)}}[pos]$ at the particular position pos in the string $D_{b_{(x,y)}}$ represents the one-bit output $c_{(x,y)}$ at coordinates (x, y) in the protected biometric template C.

2) Algorithm II: Cancelable Biometric Template Construction Using Bit-Wise XOR Operation: Fig. 4 shows the block diagram for the proposed construction using Algorithm II. Refer Appendix B for the example showing the working of Algorithm II. Considering the same format of the biometric template as in Algorithm I, the Algorithm II consists of the following functions:

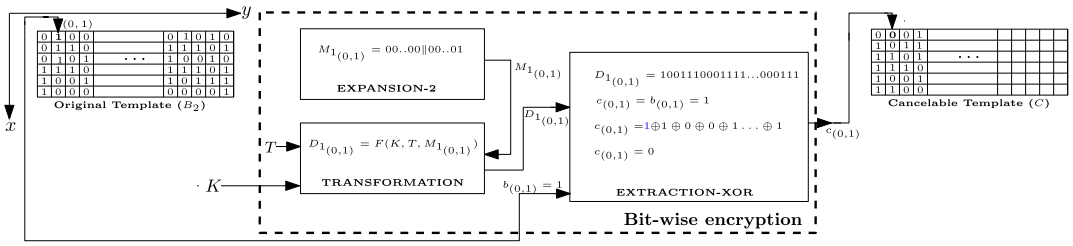


Fig. 4. Cancelable biometric template construction using bit-wise XOR operation. Example shows the mapping from original to protected iriscode template for a bit position (0, 1).

Algorithm 2 Cancelable Biometric Template Construction Using Bit-Wise XOR Operation

Input Original right biometric template B_2 extracted from the biometric data I_2.
$B_2 = \{b_{(0,0)}, b_{(0,1)}, \ldots, b_{(u-1,v-1)}\}$, tweak T, key K
Output Protected biometric template
$C = \{c_{(0,0)}, c_{(0,1)}, \ldots, c_{(u-1,v-1)}\}$
For each bit $b_{(x,y)}$ of B:

function EXPANSION-2((x, y))
  $binary_p(x) \longleftarrow x$
  $binary_p(y) \longleftarrow y$
  $M_{b_{(x,y)}} \longleftarrow binary_p(x) \| binary_p(y)$
end function
function TRANSFORMATION(K, $M_{b_{(x,y)}}$, T)
  $D_{b_{(x,y)}} \longleftarrow F(K, M_{b_{(x,y)}}, T)$ ▷ *Where F can be F_H or F_E, $|D| = d$*
end function
function EXTRACTION-XOR($b_{(x,y)}$, $D_{b_{(x,y)}}$)
  $D_{b_{(x,y)}} = D_{b_{(x,y)_0}} D_{b_{(x,y)_1}} \ldots D_{b_{(x,y)_{(d-1)}}}$
  $c_{(x,y)} \longleftarrow b_{(x,y)}$ ▷ *$c_{(x,y)}$ initialized with $b_{(x,y)}$*
  for $j = 0$ to $d - 1$ **do**
    $c_{(x,y)} \longleftarrow c_{(x,y)} \oplus D_{b_{(x,y)_j}}$
  end for
  return $c_{(x,y)}$
end function

1) **Function EXPANSION-2 to return** $M_{b_{(x,y)}}$
Concatenate the binary values of corresponding coordinates (x, y) for a particular bit $b_{(x,y)}$ to get an expanded bit string $M_{b_{(x,y)}}$
2) **Function TRANSFORMATION to return** $D_{b_{(x,y)}}$
The transformation is performed in the similar way as described in Algorithm I to output a transformed intermediate bit string $D_{b_{(x,y)}}$ of length d.
3) **Function EXTRACTION-XOR to return** $c_{(x,y)}$
   - The transformed bit string $D_{(x,y)}$ of length d is represented as

$$D_{b_{(x,y)}} = D_{b_{(x,y)_0}} D_{b_{(x,y)_1}} \ldots D_{b_{(x,y)_{(d-1)}}}$$

   - It is bit-wise XORed with the input bit $b_{(x,y)}$ to obtain the corresponding one-bit output $c_{(x,y)}$ in the protected template C as

$$c_{(x,y)} = b_{(x,y)} \oplus D_{b_{(x,y)_0}} \oplus D_{b_{(x,y)_1}} \oplus \ldots \oplus D_{b_{(x,y)_{(d-1)}}}$$

VI. Design Rationale

In this section, we discuss the significance of functions and parameters considered for the design of our proposed construction.

A. Role of Bit-Wise Encryption

Each bit $b_{(x,y)}$ of original template is mapped to the corresponding bit $c_{(x,y)}$ in the protected template, i.e.,$b_{(x,y)} \longrightarrow c_{(x,y)}$. The bit-wise encryption ensures that for each input bit, the probability of guessing the corresponding protected bit is no more than 2^{-1}. Further, any error in the original template reflects only in the corresponding bit in the protected template. Therefore, unlike the existing cancelable schemes as discussed in Section III, the number of bit errors remains the same in both the original and the protected template.

B. Role of Expansion Using Coordinates (x, y)

It ensures that each input bit is mapped to a corresponding output bit at the same bit positions, thus binds both the input and the output bits. Further, in Expansion-1, the concatenation of coordinates x and y helps to distinguish each input corresponding to a bit $b_{(x,y)}$ from others. Hence it provides different inputs to the Transformation function F resulting in a different output $D_{b_{(x,y)}}$ for each input bit $b_{(x,y)}$. In the EXTRACTION-XOR step (Algorithm II), $b_{(x,y)}$ is XORed to the output of transformation step, hence the use of $b_{(x,y)}$ as input in the Expansion-2 would not add any extra security to the scheme.

If expansion is not done, then input to the Transformation function could be 1 or 0 which would lead to only 2 different outputs, either corresponding to 1 or 0 after the transformation step. In such case, with the knowledge of transformation output corresponding to even a single bit of input biometric template, the attacker can reveal almost the whole input biometric template.

C. Role of Transformation Function F

The Transformation function F can be an encryption function such as any block cipher with the modes of operation or a keyed-hash such as HMAC. The properties of underlying cryptographic primitives ensure the secure transformation of the expanded message $M_{b_{(x,y)}}$ to generate a random transformed string $D_{b_{(x,y)}}$ of length d. Randomization makes it difficult to predict the input from $D_{b_{(x,y)}}$.

D. Role of Biometric Derived Key K

Case 1 *(A Single Key Is Generated by the System for All the Users (Contradictory Case)):* Assume that the attacker

has access to the cancelable template stored on the database. In such a scenario, the targeted-user attacker can attack by querying the oracle (client's device) with a random query biometric template B_i' for the user i. The attacker can also attempt to get authenticated by impersonating a genuine user with its biometric data (plaintext) using zero-effort attack [51], [52]. It can then obtain the corresponding random template C_i' in the protected domain using the common system's key. Since the bit-wise encryption is a deterministic mapping of input bit to output bit, and the key is same for all the users, the attacker can get the original (correct) biometric template B_i from another known protected template C_i by mapping C_i accessed from the database to B_i according to the known mapping (C_i', B_i').

Case 2 ((Proposed Approach)- A Biometric-Derived Key Generated From the Fuzzy Extractor for a Particular Instance): In this case, the key is unique for each instance and is dependent on the biometric data of that instance. Thus the attacker cannot query the client's device as an oracle by sending multiple random query templates. This is because the random biometric template will generate a random (incorrect) key. It results in the wrong transformation by bit-wise encryption approach.

E. Role of Tweak T

Tweak T is a public data of an arbitrary length. The tweak is used for padding in the input bits before the input is transformed using the Transformation function F for both the Algorithm I and Algorithm II. The tweak can or cannot be a unique value. Use of tweak in our system does not reveal any information about the biometric data of a user.

F. Role of Extraction Function

The Extraction function returns one-bit from the d-bit output of the transformation function F. It is designed to provide one-wayness, i.e., it should not be possible to get the d-bits output of function F from the one-bit output of the Extraction function. Even if an attacker gets access to the key K involved during the transformation, it is computationally infeasible to decrypt the protected template's bits to get the original iriscode. This is due to the truncation of one-bit from the whole d-bit output of F.

G. Role of Selecting a Disagreeing Bit Position pos in Algorithm I

The random string $D_{b_{(x,y)}}$ is traversed starting from index in to get a position pos with dissimilar bits. It ensures that if for a particular coordinate (x, y), bit $'1'$ maps to $'0'$ in the cancelable template, then bit $'0'$ will surely map to bit $'1'$ for the same instance and vice versa as shown in (1). This preserve the number of errors in the unprotected and protected (cancelable) domain. If the condition is unsatisfied, then the bits $'1'$ and $'0'$ taken as input from two different samples of the same instance at coordinates (x, y) might both map to either bit $'1'$ or bit $'0'$ in the cancelable template, increasing the bit-errors during authentication.

H. Role of Selecting Random Index in for Extraction-Bit Function in Algorithm I

From the empirical data, we observed that for index 0, most of the positions pos that are selected with dissimilar bits are from the range $0-7$ out of the total $d = 256$ bit positions. Whereas, for the random index value (generated using (2)), the values of dissimilar positions are uniformly distributed across the whole 256 bits output range, that makes guessing of a bit $c_{(x,y)}$ highly unpredictable. Thus, selecting a random index over index value of 0 is preferable.

TABLE I

IRIS AND FACE DATABASE DESCRIPTION

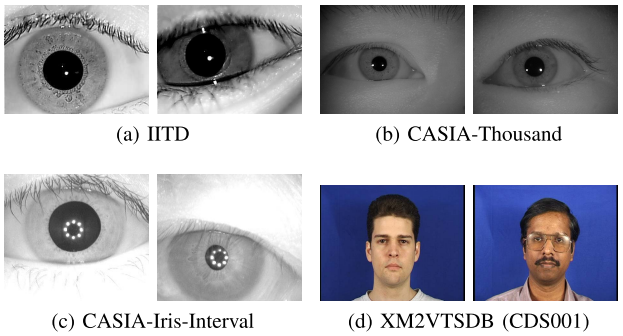
Modalities	Database(s)	Subjects	Samples	Resolution
Iris	IITD [53]	448	5	320×240
Iris	CASIA-Iris-Thousand[1]	2000	10	640×480
Iris	CASIA-Iris-Interval[1]	337	5	640×480
Face	XM2VTSDB (CDS001) [54]	295	4	720×576

[1]http://biometrics.idealtest.org/

TABLE II

VIRTUAL MULTI-MODAL DATABASES WITH GENUINE AND IMPOSTOR DISTRIBUTION. FACE DENOTES XM2VTSDB DATABASE

Database	Genuine	Impostor
IITD-CASIA Left Iris & Right Iris	128	2079
CASIA-Iris-Interval Left Iris & Right Iris	128	36
IITD-CASIA & CASIA-Iris-Interval	256	81
IITD-CASIA & Face	256	39
CASIA-Iris-Interval & Face	256	39



(a) IITD (b) CASIA-Thousand

(c) CASIA-Iris-Interval (d) XM2VTSDB (CDS001)

Fig. 5. Example images from the selected databases.

VII. EXPERIMENTS AND PERFORMANCE ANALYSIS

We evaluate the experiments for iris and face on the publicly available databases given in Table I. The example images are shown in Fig. 5. We consider the left and right instances as mutually independent subjects. To create a multi-modal database, we combined the samples from the two different instances with one-to-one correspondence, while deleting the extra samples. We combine the IITD database with CASIA-Iris-Thousand database, denoted as IITD-CASIA database in which all the (2000×10) samples from CASIA-Iris-Thousand database are taken as impostors. The distribution of genuine and impostor samples for multi-modal datasets is shown in Table II. We have not considered the case where both the input templates are from face characteristics. Implementing fuzzy extractor (with error correcting code) for face template as input is inefficient due to its large size (76, 800 bits).

For Baseline-A, the existing template protection schemes and our proposed algorithms, we use open source libraries and software for feature extraction. For iriscode generation, we use OSIRIS [55] and University of Salzburg Iris Toolkit v1.0 [56]. Feature extraction is performed with Daugman-like 1D-Log Gabor (LG) algorithm proposed by Masek [57] to generate

[1]http://biometrics.idealtest.org/

TABLE III
PARAMETERS USED FOR IMPLEMENTATION

Parameters	Value (in bits)
Size of Iriscode's template	10240
Size of Face binarized template	76800
Length k of key K	128
Size of fixed-length tweak T	128
Size of arbitrary-length tweak T	64-128

iriscodes of size $512 \times 20 = 10240$ bits. 16 bloom filters are constructed considering best parameters [45] with height of iriscode $= 10$, the width of each block of iriscode $= 32$. For face features extraction based on local Gabor pattern histogram sequences, we use the FaceRecLib of the free signal and image processing toolbox Bob[2] [58], [59]. We cropped each image to obtain central 4×8 sub-image with $32 \times 2400 = 76800$ bits. 960 bloom filters are generated for binarized face templates, each of size 16 bits as described in [5], [39]. All the comparisons for the face and iris templates are made by using the Hamming distance metric between two binarized templates. Circular bit-shifts are often done to align two iriscodes [50], to compensate for the binary misalignment during comparisons. Hamming distance is computed for each shift position (i.e. relative tilt angle, in our case, ± 4 bits), and the minimum Hamming distance is taken as the final comparison score. For Baseline-B, we utilize the state-of-the-art commercial-off-the-shelf (COTS) matchers that provide the comparison scores. We use VeriEye [60] (Neurotechnology) for iris and VeriLook [61] (Neurotechnology) for the face feature extraction and comparisons.

A. Parameters for Transformation Function

In this section, we briefly discuss the parameters used for transforming the original biometric templates to the protected cancelable templates.

We use BCH code for the implementation of the fuzzy extractor. It has been observed from the literature [62], [63] that BCH codes are simple and are the suitable choice of error correcting codes for the implementation of fuzzy extractors.

BCH Codes: Bose-Chaudhuri-Hocquenghem codes are random error-correcting cyclic codes constructed using polynomials over the Galois field. For any positive integers ($q \geq 3$) and ($t < 2^{(q-1)}$), there exists a t-error-correcting BCH code (n, χ, t) with a random, secret message of length χ with the following parameters [64], [65]: Block length gives the size of error correcting codeword and is denoted as $n = 2^q - 1$. The number of parity-check bits which are used to recover the original codeword from a received codeword is denoted as $n - \chi \leq qt$, and minimum distance gives the minimum distance between any two codewords, such that each codeword corrects a maximum of t error bits. It is given as $d_{min} \geq 2t + 1$. Considering approximately 20% errors in the iriscode, for iriscode of length 10240 bits we selected $(1023, 46, 219)$-BCH Code. We sampled the $(1023, 46, 219)$-BCH Code 10 times to cover 10230 bits of iriscode, ignoring the last few bits. The number of bit-errors in binarized face templates are high; however, with an efficient error correcting code, the face templates can be used.

[2]http://idiap.github.io/bob/

Table III shows the parameters that we suggest and use for implementing the proposed algorithms. To impose security while implementing the proposed algorithms, SHA-2 or SHA-3 are preferred hash functions. Besides, for the transformation function instantiated with encryption using block ciphers, we choose AES-CBC mode [37]. An IV is required to encrypt the plaintext using AES-CBC mode. To generate an IV required during encryption, we use the standard recommended method [37]. Apart from encryption using a block cipher, we can use a key-based hash function known as hash-based message authentication code (HMAC) that provides one-way transformation. The motivation behind choosing AES-CBC and HMAC is their practical deployment on large scale systems. Both these cryptographic primitives are well analyzed, secure and efficient in terms of performance.

B. Recognition Performance Evaluation

We plot the DET curve (Detection Error Trade-off) that demonstrates the false match rates (FMR) against the false non-match rates (FNMR) to evaluate the recognition performance. While considering various existing biometric template protection approaches, the following cases are constructed depending on the role of f_2 (shown in Fig. 1):

- Baseline-A- The fuzzy extractor and f_2 have no role. The comparisons for both the input templates are made using Hamming distance metric [50].
- Baseline-B- COTS systems used for face and iris' features extraction and comparisons.
- Bloom filter-based approach [39]- The first template is given to the fuzzy extractor to generate the key K with the comparisons done using the Hamming distance metric. f_2 transforms the second biometric template to bloom filter based arrays using K. The comparisons are made by matching corresponding bloom filter arrays [44] using the Hamming distance metric.
- Bloom filter based fusion approach [5]- Here, the bloom filter based templates generated from the two input biometric templates are fused using the simple OR operation. The comparisons are made on the fused bloom filter arrays. The fuzzy extractor and f_2 is not used.
- Proposed approach- The first template is given as input to the fuzzy extractor to generate the secret key. The comparisons are made using the Hamming distance metric. f_2 performs the bit-wise encryption on the second template where it is transformed by one of our proposed algorithms with comparisons done using Hamming distance metric.

Fusion of Scores: In the third case, multi-biometric bloom filter based fusion approach [5], the fused bloom filter arrays are directly compared using Hamming distance metric [5] to give the final scores without any fusion required on scores. In rest of the cases, the two individual scores are computed for each of the two input biometric templates, based on the underlying approach. The scores are first converted to a common range before they can be fused, known as normalization. It is done using the reduction of high-scores effect (RHE) normalization [66]. The normalized scores are then fused with the sum-rule based fusion approach, i.e. for two normalized scores x_1 and x_2, considering equal weights to both, the fused score is represented as $x_1 + x_2$.

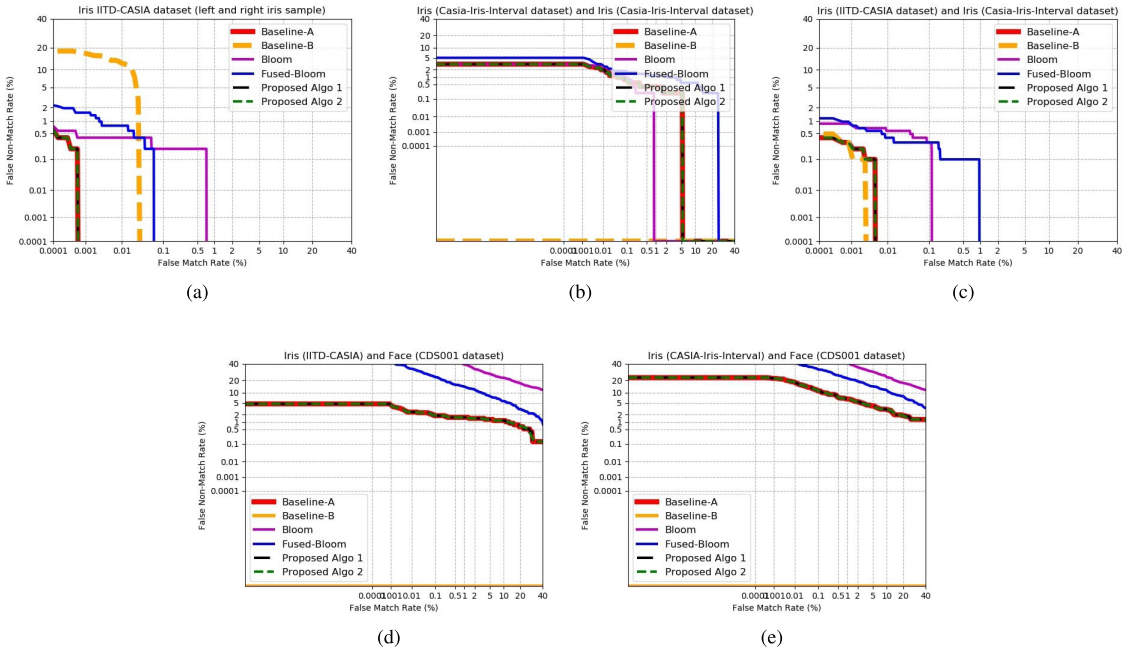


Fig. 6. Recognition performance evaluation for multi-biometric databases (a) IITD-CASIA (left & right samples) (b) CASIA-Iris-Interval (left & right samples) (c) IITD-CASIA & CASIA-Iris-Interval (d) IITD-CASIA & Face (e) CASIA-Iris-Interval & Face. Here, Baseline-A (open source) and Baseline-B (COTS), both represent the unprotected template approaches, bloom represents combining fuzzy extractor with bloom filter based approach [39], [44], fused-bloom represents the bloom filter based fusion approach [5]. Proposed algo 1 and proposed algo 2 represent combining fuzzy extractor with proposed Algorithm I and Algorithm II. (Refer web version to interpret colors in figure legends).

It can be observed from the DET curves plotted in Fig. 6 that the recognition performance of both our proposed algorithms is the same as that of the Baseline-A approach and the completely overlapped curves can depict the same for the Baseline-A and the proposed algorithms. Further, the performance of our proposed scheme outperforms the performance of the Bloom filter-based approach [39] and multi-biometric bloom filter based fusion approach [5] at 0.01% false match rate. It can be noticed that homomorphic encryption schemes [4] generally preserves the errors rate, producing a similar curve as the baseline approach. The Baseline-B (COTS systems) achieves the highest performance in most of the cases; though we can achieve the equivalent performance by optimizing the use of underlying feature extractor and comparison modules in our proposed algorithms.

C. Efficiency Evaluation for the Generation of Protected Template

We calculate the efficiency of our proposed work in terms of the time taken during enrolment and authentication.

1) Enrolment Time: the time taken to generate a secret key using fuzzy extractor (using BCH encoder) from the first template + time taken to transform the second template using bit-wise encryption.

2) Authentication Time: is given by the time taken to decode the secret key using fuzzy extractor from first template + time taken to transform the second template using bit-wise encryption + verification time taken to compare two templates.

Our implementation consists of two steps.

- The fuzzy extractor implementation that majorly involves BCH encoding (during enrolment) and decoding module (during authentication). We use BCH encoding and decoding modules written in C language.[3] A server-grade processor; Intel Xeon 2.10 GHz, 8 core with hyper-threading for profiling BCH encoding and decoding subroutines are used in our experiments.
- Generate a cancelable template using proposed bit-wise encryption. To compute the time taken by f_2, we profiled our Java-based implementation on desktop-grade hardware (4 cores) of varying clock frequencies, including Intel i5 (Kaby Lake, 2.1 GHz), Intel i5 (Skylake 2.5 GHz), Intel i7 (Skylake, 2.6 GHz) and Intel i5 (Skylake, 3.3 GHz). We ran our experiments on various Android-based smartphones as well. An AES-NI (cryptographic hardware instructions) enabled processor can perform around 22952162 times *AES128* encryption in *CBC* mode. In our desktop implementation, we used Intel's AES-NI instruction set for fast *AES* computation inside a *C* library and interfaced the *C* library to our Java implementation using Java Native Interface (JNI).

In Table IV and V, we show a comparison of enrolment and authentication performance with 4 cases: (i) f_2 as existing Bloom filter based scheme [44], (ii) f_2 as homomorphic encryption-based scheme [4], (iii) f_2 as proposed Algorithm I and (iv) f_2 as proposed Algorithm II. All these 4 cases are protected template approaches. We also consider the baseline cases during authentication, where hamming distance is computed between the two unprotected iriscodes. The BCH code-encoding and decoding function consumes most of the time in the fuzzy extractor step. This time is the same for all the approaches (bloom filter-based, multi-biometric fusion and our two proposed algorithms).

The following inferences can be drawn.

[3]http://www.eccpage.com/bch3.c

TABLE IV

EFFICIENCY EVALUATION DURING ENROLMENT: TIME TAKEN (IN MILLISECONDS) TO GENERATE A CANCELABLE BIOMETRIC TEMPLATE BY f_2 (SHOWN IN FIG. 1). IT TAKES INPUTS AS A SECOND BIOMETRIC TEMPLATE AND THE KEY DERIVED FROM THE FUZZY EXTRACTOR USING THE FIRST BIOMETRIC TEMPLATE (IRISCODE ALWAYS, IN OUR EXPERIMENTS). WE HAVE NOT ADDED THE TIME TAKEN BY THE FUZZY EXTRACTOR TO DERIVE THE SECRET KEY FROM THE FIRST TEMPLATE GIVEN BY USER DURING ENROLMENT IN THE TABLE. EXPERIMENTALLY, THE FUZZY EXTRACTOR TAKES 4.37 MILLISECONDS TO DERIVE A KEY FROM IRISCODE TEMPLATE

Biometric characteristics	**Iris**								**Face**							
Platform/ Approaches	**Desktop (Processor clock freq. (GHz)**				**Android (smartphone model)**				**Desktop (Processor clock freq. (GHz)**				**Android (smartphone model)**			
	2.1	**2.5**	**2.6**	**3.3**	**Samsung Galaxy S6**	**One Plus 6**	**Google Pixel 2**	**Samsung Note 9**	**2.1**	**2.5**	**2.6**	**3.3**	**Samsung Galaxy S6**	**One Plus 6**	**Google Pixel 2**	**Samsung Note 9**
Bloom filter based [44]	7	6	4	2	24	7	12	9	20	18	17	14	24	9	12	9
Paillier cryptosystem based [4]	105	100	95	70	510	98	780	470	1240	1100	700	488	510	330	780	470
Proposed Algorithm I	17	16	14	12	800	460	580	600	220	190	130	105	800	460	580	600
Proposed Algorithm II	9	8	7	7	678	350	524	550	190	175	110	90	687	230	524	550

TABLE V

EFFICIENCY EVALUATION DURING AUTHENTICATION: TIME TAKEN (IN MILLISECONDS) TO PERFORM BIOMETRIC AUTHENTICATION (SPECIFICALLY THE TIME TAKEN FOR VERIFICATION BETWEEN ANY TWO TEMPLATES). IT INCLUDES THE TIME TAKEN BY f_2 TO TRANSFORM THE TEMPLATE AND THE COMPARISON TIME TO COMPARE THE TWO BIOMETRIC TEMPLATES. WE HAVE NOT ADDED THE TIME TAKEN BY THE FUZZY EXTRACTOR TO RECOVER THE SECRET KEY FROM THE FIRST TEMPLATE GIVEN BY USER DURING AUTHENTICATION IN THE TABLE. EXPERIMENTALLY, THE FUZZY EXTRACTOR TAKES AROUND 10.952 MILLISECONDS FOR KEY RECOVERY (BY BCH CODEWORD DECODING) DURING AUTHENTICATION

Biometric characteristics	**Iris**								**Face**							
Platform/ Approaches	**Desktop (Processor clock freq. (GHz)**				**Android (smartphone model)**				**Desktop (Processor clock freq. (GHz)**				**Android (smartphone model)**			
	2.1	**2.5**	**2.6**	**3.3**	**Samsung Galaxy S6**	**One Plus 6**	**Google Pixel 2**	**Samsung Note 9**	**2.1**	**2.5**	**2.6**	**3.3**	**Samsung Galaxy S6**	**One Plus 6**	**Google Pixel 2**	**Samsung Note 9**
Unprotected (Baseline-A)	0.042	0.040	0.031	0.027	0.243	0.224	0.221	0.213	1.609	1.365	1.600	1.597	3.141	2.994	3.119	3.064
Bloom filter based [44]	7.044	6.042	4.030	2.023	24.209	7.201	12.212	9.211	21.510	19.476	18.443	15.335	26.730	11.314	14.947	11.490
Paillier cryptosystem based [4]	2172	2142	2132	2112	2901	2241	2783	2466	2811	2744	2715	2634	3010	2811	2783	2791
Proposed Algorithm I	17.048	16.042	14.032	12.022	800.223	460.191	580.212	600.209	221.790	191.766	131.711	106.680	803.341	463.040	583.210	603.092
Proposed Algorithm II	9.043	8.042	7.031	7.022	678.213	350.201	524.211	550.220	191.712	176.680	111.600	91.581	690.203	233.003	527.129	553.010

- During enrolment and authentication, the proposed Algorithm II defined over block cipher modes of operation on desktop architecture on all the given clock frequencies is significantly comparable to the existing Bloom filter based scheme [44] in terms of time.
- In general, it can be seen on both the desktop as well as the Android-based implementation, the Algorithm I is slower than Algorithm II; This is because, in Algorithm I, we perform two cryptographic operations for each bit thereby increasing the computation and hence higher turnaround time.
- During authentication, the homomorphic encryption-based scheme [4] uses an additional call to the client to get the encrypted threshold, thereby adding a round trip time or RTT (we fixed it to 2 seconds). Due to this RTT, the homomorphic encryption-based scheme is comparatively slow.
- While the time taken for enrolment by homomorphic encryption-based scheme [4] on Android-based devices outperforms both the algorithms proposed, in the desktop-based architectures, both our proposed algorithms significantly outperform the homomorphic encryption-based scheme [4]. This contrariety is due to the use of AES-NI instruction set in desktop-based implementation. It provides multitude increase in encryption and decryption operations. We can get the same efficiency on Android as well (where we use only high-level application program interface) by implementing the hardware instructions. Further, we achieved faster implementation for encryption using block cipher modes of operation than keyed-hash functions (HMAC) since we use AES-NI instruction set for the implementation of encryption based on block ciphers.
- In the case when ± 4 bit-shifts are considered, the time taken to generate the protected template increases to about 2 to 4 times of the time taken by a single protected template.[4] In the case of Android, the time would be about 5 to 8 times of the single protected template creation time since Android manages memory more aggressively. For faster implementation, enrolment can be done offline with the storage of all the protected templates with bit-shifts applied for a particular instance. During authentication, the Hamming distance is computed with the protected templates stored for a user which takes almost negligible time for all the ± 4 bit-shift cases for a particular instance.
- Although the bloom filter based scheme [44] outperforms all the other protected approaches, the error rates in bloom filter based scheme are high as compared to our proposed algorithms. It limits the applicability of these schemes on a wider scale.

VIII. SECURITY AND PRIVACY ANALYSIS

Our proposed bit-wise encryption scheme is secure in terms of secrecy and irreversibility. Please refer Appendix A for the mathematical proof that justifies the security of the scheme.

[4]Most of the data like a key, tweak etc. remains in memory for all bit shifts.

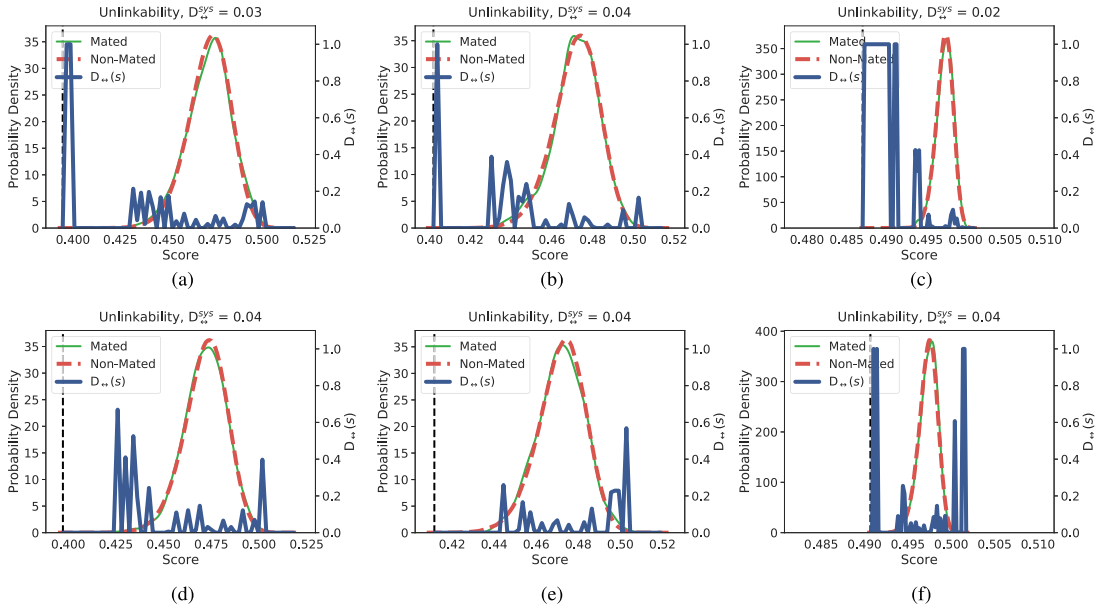


Fig. 7. Unlinkability evaluation [67]: (a) IITD-CASIA Algorithm I (b) CASIA-Iris-Interval Algorithm I (c) Face Algorithm I (d) IITD-CASIA Algorithm II (e) CASIA-Iris-Interval Algorithm II (f) Face Algorithm II.

Assumption: We assume that the system is robust enough to handle spoofing using a spoof detection module (beyond the scope of this paper). Even though the face is easily prone to the spoof attacks as compared to iris, much research has been done on the detection of presentation attacks (PAD) in [68]–[70]. Though we suggest to design the system with biometric characteristics which are difficult to spoof, such as iris which is more robust to spoofing [71]. In such a scenario, the attack complexity of an attacker to get access to one biometric template of a genuine user is equivalent to the attack complexity to get both the biometric templates of the particular user.

As stated in ISO/IEC IS24745:2011 [14], the protected biometric template satisfies 3 main privacy properties:

A. Irreversibility

Irreversibility or non-invertibility states that for a given protected biometric template, it is computationally infeasible for an attacker to recover the original biometric template with or without the knowledge of secret parameters involved. The original biometric templates can be guessed in the following ways:

1) **Predict one or both the original templates from the given protected biometric template:** Given that key is not known, for each bit of the protected template, the attacker needs to reverse the Extraction and the Transformation functions of the bit-wise encryption scheme to get the corresponding one-bit of the original template. The mathematical proof in Appendix A shows that the bit-wise encryption scheme preserves irreversibility for both the algorithms, given the underlying transformation function F with a secret parameter, key K. Further, following the suggestions provided in Section VIII B, the helper data generated from the first biometric template using fuzzy extractor would not reveal any information about the original biometric template.
2) **Chosen Plaintext Attack:** Chosen plaintext attack presumes that an attacker can choose some arbitrary plaintext and can obtain corresponding ciphertexts. Let the client's device acts as a random encryption oracle. Assume that attacker knows the key K and has access to the cancelable templates stored on the database. The CPA can be carried out by the targeted user attacker, as explained in Section VI. In our case, if the first biometric template B_1 is not known to the attacker, the attacker would not be able to derive the correct key K from it. Without the correct key attacker cannot perform chosen-plaintext attack on the second biometric template B_2.

B. Unlinkability

Unlinkability states that given different samples of the same instance of a particular user, an attacker should not be able to determine whether two biometric templates are derived from the same or the different instances.

1) Unlinkability Analysis of Fuzzy Extractor: The helper data generated as output from the fuzzy extractor is stored as public on the client's device. It often leaks information about linkability of biometric templates [17], [20] if accessed by an attacker during the offline attack mode. To prevent the linkability information leakage, we suggest the following procedures:

- The helper data can be encrypted using a system's specific secret key stored in the trusted platform module (TPM) [72] on the client's device. Since it is not possible to break the TPM, the key is secure, and hence, during the offline mode, the attacker will get only encrypted helper data.
- Recently proposed reusable fuzzy extractors [34]–[36] remain secure even when the attacker knows the helper data generated from correlated values of a user multiple times.

2) Unlinkability Analysis of Bit-Wise Encryption Scheme: Unlinkability is measured based on the mated (genuine and enroled samples across different applications) and

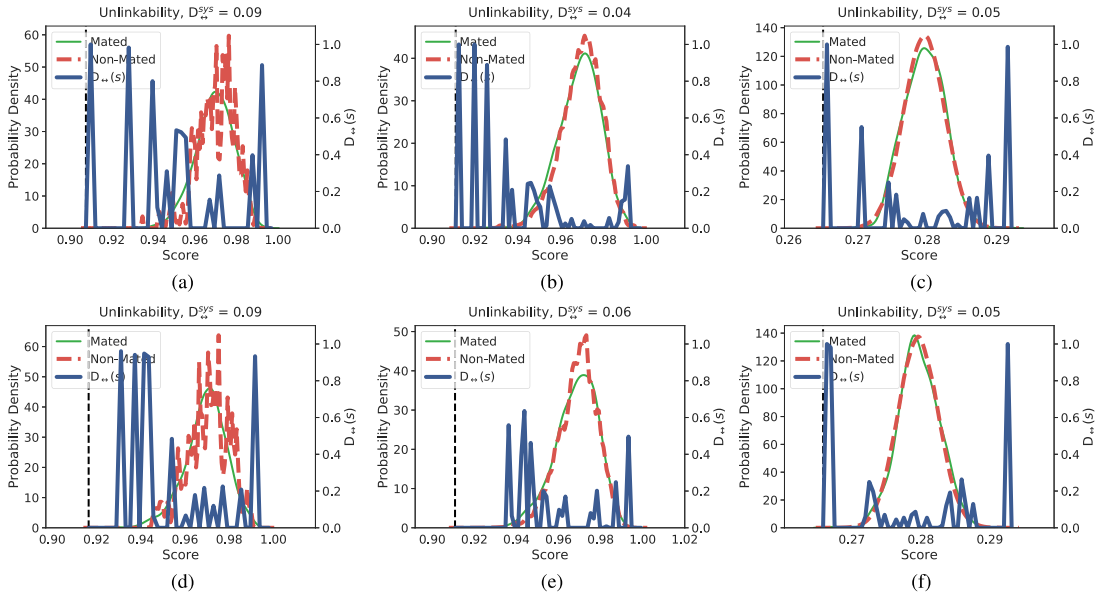


Fig. 8. Further linkage functions [39]. Unlinkability evaluation using Hamming weight function: (a) IITD-CASIA Algorithm I (b) CASIA-Iris-Interval Algorithm I (c) Face Algorithm I (d) IITD-CASIA Algorithm II (e) CASIA-Iris-Interval Algorithm II (f) Face Algorithm II. The black dashed line represents likelihood ratio $LR(s)$ as denoted in [39].

non-mated (impostors and enroled samples across different applications) samples distribution. As proposed by Gomez-Barrero *et al.* [67], we compute two different measures for the linkability of iriscodes. Local measure $D_{\leftrightarrow}(s)$ evaluates the linkability in a score-wise basis. Further, Global measure D_{sys} provides the linkability of the whole system and is independent of the score domain of the system. Fig. 7 shows the distribution of the mated and non-mated sample calculated from the dissimilarity scores. It is clear from the graph that the distributions are significantly overlapped with global linkability D_{sys} close to 0. Since the samples of biometric template for a particular instance are similar but not identical, the key generated from these samples would be completely different for each sample. Thus, the unlinkability is preserved due to the use of different secret key across multiple applications.

3) Further Linkage Functions: Hamming Weights Function Applied on Bloom Template: The two bloom filter based biometric samples of the same instance protected using the different keys have similar Hamming weights [39], [73]. The Hamming weight difference $diff$ between two samples, bf_1, bf_2 is evaluated as $diff = |HW(bf_1) - HW(bf_2)|$, where HW is the number of ones in the bloom filter based template. In this case, knowledge of protected templates is required by an attacker to find the linkage between them. From the protected template generated using our proposed approach, the attacker can generate the bloom filter based templates, and Hamming weights function can be applied to detect linkability. The plots for unlinkability measure are shown in Fig. 8.

Considering two linkage functions (Hamming distance based comparison on protected iriscodes using proposed Algorithms and Hamming weights function on bloom filter based templates), for Algorithm I, the global linkability value of system for IITD-CASIA database is given as $D_{sys} = max\{0.03, 0.09\} = 0.09$, for CASIA-Iris-Interval database, $D_{sys} = max\{0.04, 0.04\} = 0.04$ and for face database, $D_{sys} = max\{0.02, 0.05\} = 0.05$. For Algorithm II, the global linkability value of system for IITD-CASIA database is given as $D_{sys} = max\{0.04, 0.09\} = 0.09$, for CASIA-Iris-Interval database, $D_{sys} = max\{0.04, 0.06\} = 0.06$ and for face database, $D_{sys} = max\{0.04, 0.05\} = 0.05$.

C. Renewability

Renewability states that when an existing protected template is compromised, it should be possible to revoke or cancel the compromised protected template and re-generate a new protected biometric template using a different security parameter. This can be ensured by having a large key space, preferable with a key of length equal to or greater than 128. If a key is compromised, a new key can be issued by changing the underlying error correcting codeword for a particular user.

IX. Conclusions

We propose a cancelable multi-biometric approach for biometric authentication system by combining the fuzzy extractor with a novel bit-wise encryption scheme to generate cancelable biometric templates. The proposed work fulfils the ISO/IEC IS 24745:2011 recommended prerequisites for biometric template protection schemes stated as irreversibility, unlinkability, renewability and high biometric recognition performance. The novel bit-wise encryption scheme ensures that no additional noise in terms of bit errors is generated in the protected template which makes the performance of our proposed scheme equivalent to the unprotected systems' performance (Baseline-A). Further, the experimental results show that the proposed system outperforms the existing cancelable biometric scheme in terms of recognition performance. The empirical results for measuring efficiency in time units show that for desktop with clock frequency 3.3 GHz, our Algorithm II based approach takes around 12 milliseconds during enrolment and 18 milliseconds during authentication phase on an average of over 200 runs. The mathematical proof provided in the paper justifies the secrecy and irreversibility of the bit-wise encryption scheme.

TABLE VI

EXAMPLE FOR ALGORITHM I. HERE, B_1 AND B_1' ARE THE TWO SAMPLES OF THE SAME INSTANCE, WHEREAS B_2 DENOTES THE SAMPLE FROM ANOTHER INSTANCE. $b_{(x,y)}$ DENOTES A PARTICULAR BIT IN THE SAMPLE AT COORDINATES (x, y). T AND K DENOTES THE TWEAK AND KEY RESPECTIVELY. REFER SECTION II FOR NOTATIONS AND ALGORITHM I FOR DETAILS. SEE WEB VERSION TO INTERPRET COLORS

We conclude that the proposed scheme is not only theoretically sound but can also be practically deployable on large scale systems such as India's Aadhaar project, particularly for Aadhaar-based verification [74]; while achieving the high recognition performance and efficiency as well as strong security. In addition to the applications in biometric authentication, the proposed bit-wise encryption scheme can be applied to a wide range of scenarios where the input that needs to be secured is in the binary format. For instance, for any crucial security data (binarized) that needs to be stored on cloud servers. Another application could be the digital forensics where fuzzy hashing [75] is used; the one-bit encryption could be used as a part of fuzzy hashing approach. We provide a generalized, modular approach where the bit-wise encryption scheme is used with a feature extractor, a matcher and a fuzzy extractor to provide secure storage of biometric templates on the database servers. The bit-wise encryption scheme preserves the errors completely, independent of the dataset or the underlying modules used. Thus to improve the performance and efficiency of our approach, it can be built over the commercial-off-the-shelf systems (with optimized feature extractor and matchers) to achieve the equivalent performance with high security and hence, wider scalability.

As a part of our future work, we would like to implement an efficient fuzzy extractor for face templates. Further, the reusable fuzzy extractor takes huge storage. We would like to work on its implementation feasibility in our proposed approach.

APPENDIX A
MATHEMATICAL PROOF OF BIT-WISE ENCRYPTION SCHEME

We establish the following results. Our bit-wise encryption scheme S is secure and irreversible for a given input sample space $\mathcal{M} \in \{0,1\}$, a transformed output sample space $\mathcal{C} \in \{0,1\}$ and a random key space $\mathcal{K} \in \{0,1\}^{|K|}$. Following are the definitions that are used in the proofs.

- Secrecy: Given an input sample $m \in \mathcal{M}$ and output sample $c \in \mathcal{C}$, $Pr[S(m) = c] = Pr[S(m') = c]$, $\forall\{m, m'\} \in \mathcal{M}$

TABLE VII

EXAMPLES FOR ALGORITHM II. REFER THE CAPTION OF TABLE VI FOR DETAILS. SEE WEB VERSION TO INTERPRET COLORS

Input: $B_1 = 111111110.....100000111$, $T_1 = 01000.....0110$, K_1 **Output:**$C_1 = 010111000.....011100010$ for each bit $b_{(x,y)}$ of B_1: taking 0th position bit, $(0,0)$ = position 0 $b_{(x,y)} = 1_{(0,0)}$ **EXPANSION-2** $((0,0))$: $M_{1_{(0,0)}} \longleftarrow 0000000000000000$ **TRANSFORMATION** $(K_1, M_{1_{(0,0)}}, T_1)$: $D_{1_{(0,0)}} \longleftarrow AES(K_1, M_{1_{(0,0)}}, T_1)$ $D_{1_{(0,0)}} = 1111100.....1001111$ **EXTRACTION-XOR** $(1_{(0,0)}, D_{1_{(0,0)}})$: XORing together each bit in $D_{1_{(0,0)}}$ to get $1 \oplus 1 \oplus \ldots \oplus 1 \oplus 1 \oplus 1 = 1$ $c_{(0,0)} = 1 \oplus 1$ $c_{(0,0)} = 0$ taking 5th position bit, $(0,5)$ = position 5 $b_{(x,y)} = 1_{(0,5)}$ **EXPANSION-2** $((0,5))$: $M_{1_{(0,5)}} \longleftarrow 0000000000000101$ **TRANSFORMATION** $(K_1, M_{1_{(0,5)}}, T_1)$: $D_{1_{(0,5)}} \longleftarrow AES(K_1, M_{1_{(0,5)}}, T_1)$ $D_{1_{(0,5)}} = 1100000.....1111000$ **EXTRACTION-XOR** $(1_{(0,5)}, D_{1_{(0,5)}})$: XORing together each bit in $D_{1_{(0,5)}}$ to get $1 \oplus 1 \oplus \ldots \oplus 0 \oplus 0 \oplus 0 = 0$ $c_{(0,5)} = 1 \oplus 0$ $c_{(0,5)} = 1$ Similarly, we obtain $c_{(x,y)}$ for all input bits to get $C_1 = c_{(0,0)}c_{(0,1)}\ldots c_{(0,10239)}$.	**Input:** $B'_1 = 111110000.....100000111$, $T'_1 = T_1 = 01000.....0110$, K_1 **Output:**$C'_1 = 010110110.....011100010$ for each bit $b_{(x,y)}$ of B'_1: taking 0th position bit, $(0,0)$ = position 0 $b_{(x,y)} = 1_{(0,0)}$ **EXPANSION-2** $((0,0))$: $M_{1_{(0,0)}} \longleftarrow 0000000000000000$ **TRANSFORMATION** $(K_1, M_{1_{(0,0)}}, T_1)$: $D_{1_{(0,0)}} \longleftarrow AES(K_1, M_{1_{(0,0)}}, T_1)$ $D_{1_{(0,0)}} = 1111100.....1001111$ **EXTRACTION-XOR** $(1_{(0,0)}, D_{1_{(0,0)}})$: XORing together each bit in $D_{1_{(0,0)}}$ to get $1 \oplus 1 \oplus \ldots \oplus 1 \oplus 1 \oplus 1 = 1$ $c_{(0,0)} = 1 \oplus 1$ $c_{(0,0)} = 0$ taking 5th position bit, $(0,5)$ = position 5 $b_{(x,y)} = 0_{(0,5)}$ **EXPANSION-2** $((0,5))$: $M_{0_{(0,5)}} \longleftarrow 0000000000000101$ **TRANSFORMATION** $(K_1, M_{0_{(0,5)}}, T_1)$: $D_{0_{(0,5)}} \longleftarrow AES(K_1, M_{0_{(0,5)}}, T_1)$ $D_{0_{(0,5)}} = 1100000.....1111000$ **EXTRACTION-XOR** $(0_{(0,5)}, D_{0_{(0,5)}})$: XORing together each bit in $D_{0_{(0,5)}}$ to get $1 \oplus 1 \oplus \ldots \oplus 0 \oplus 0 \oplus 0 = 0$ $c_{(0,5)} = 0 \oplus 0$ $c_{(0,5)} = 0$ Similarly, we obtain $c_{(x,y)}$ for all input bits to get $C'_1 = c_{(0,0)}c_{(0,1)}\ldots c_{(0,10239)}$.	**Input:** $B_2 = 111111110.....100000111$, $T_2 = 01010.....0001$, K_2 **Output:**$C_2 = 000110011.....110110101$ for each bit $b_{(x,y)}$ of B_2: taking 0th position bit, $(0,0)$ = position 0 $b_{(x,y)} = 1_{(0,0)}$ **EXPANSION-2** $((0,0))$: $M_{1_{(0,0)}} \longleftarrow 0000000000000000$ **TRANSFORMATION** $(K_2, M_{1_{(0,0)}}, T_2)$: $D_{1_{(0,0)}} \longleftarrow AES(K_2, M_{1_{(0,0)}}, T_2)$ $D_{1_{(0,0)}} = 1110111.....0010110$ **EXTRACTION-XOR** $(1_{(0,0)}, D_{1_{(0,0)}})$: XORing together each bit in $D_{1_{(0,0)}}$ to get $1 \oplus 1 \oplus \ldots \oplus 1 \oplus 1 \oplus 0 = 1$ $c_{(0,0)} = 1 \oplus 1$ $c_{(0,0)} = 0$ taking 5th position bit, $(0,5)$ = position 5 $b_{(x,y)} = 1_{(0,5)}$ **EXPANSION-2** $((0,5))$: $M_{1_{(0,5)}} \longleftarrow 0000000000000101$ **TRANSFORMATION** $(K_2, M_{1_{(0,5)}}, T_2)$: $D_{1_{(0,5)}} \longleftarrow AES(K_2, M_{1_{(0,5)}}, T_2)$ $D_{1_{(0,5)}} = 1100001.....1010111$ **EXTRACTION-XOR** $(1_{(0,5)}, D_{1_{(0,5)}})$: XORing together each bit in $D_{1_{(0,5)}}$ to get $1 \oplus 1 \oplus \ldots \oplus 1 \oplus 1 \oplus 1 = 1$ $c_{(0,5)} = 1 \oplus 1$ $c_{(0,5)} = 0$ Similarly, we obtain $c_{(x,y)}$ for all input bits to get $C_2 = c_{(0,0)}c_{(0,1)}\ldots c_{(0,10239)}$.

- Irreversibility: Given an output sample $c \in \mathcal{C}$, $Pr[\exists m \in \mathcal{M} | S(m) = c]$ is computationally bounded by $O(2^{|K|})$ where $K \in \mathcal{K}$ is a security parameter.

Statement 1*:* Bit-wise security implies the overall security of the scheme S.

Proof: The proposed bit-wise encryption scheme follows two conditions:

- Each input bit taken from the original biometric template is mapped to its corresponding output bit in the protected template as shown in (1). Clearly, this condition ensures one-to-one mapping between the input bit and the corresponding output bit, thereby ensuring that bit errors are fully preserved in the protected biometric template.
- The output bit in the protected template corresponding to given input bit is completely random, i.e. the probability of guessing each output bit is no more than 2^{-1}.

A function F with a secret key K transforms a fixed length input to a fixed length random output. In real life, such F can be instantiated with various cryptographic primitives. Two such choices could be deterministic encryption function F_E and keyed hash function F_H. In general,

$$F_E : \{0,1\}^n \times \{0,1\}^{|K|} \rightarrow \{0,1\}^n \quad \text{or}$$
$$F_H : \{0,1\}^n \times \{0,1\}^{|K|} \rightarrow \{0,1\}^\lambda$$

where $|K|$ is the size of key and λ is the size of hash output.

Thus, from the property of transformation function (computationally bounded) given as $D_{b_{(x,y)}} \longleftarrow F(K, M_{b_{(x,y)}}, T)$, the obtained output string $D_{b_{(x,y)}}$ is random, where input $M_{b_{(x,y)}}$ is generated by considering the coordinates (x, y) at positions of each input bit $b_{(x,y)}$ (in the expansion step). Tweak is a public parameter. Hence for a user, different inputs are processed through the same secret key to provide different random outputs.

Each bit after transformation produces an output $D_{b_{(x,y)}}$ of size n or λ that is random. Further, from $D_{b_{(x,y)}}$, a random output bit $c_{(x,y)}$ is extracted (in Statement 2). Cryptographic properties of F ensure that it is computationally infeasible to find the corresponding input bit $b_{(x,y)}$, given the output $D_{b_{(x,y)}}$. Hence, bit security is achieved. □

Statement 2*:* The proposed bit-wise scheme S is irreversible, given that K is a secret and is unique for each user.

Proof: A random string $D_{b_{(x,y)}}$ is obtained using function F as shown in Statement 1. From this random string, an output bit $c_{(x,y)}$ corresponding to each input bit $b_{(x,y)}$ is obtained in the extraction step.

In the Extraction-Bit step (Algorithm I), a random output bit $c_{(x,y)}$ is extracted from $D_{b_{(x,y)}}$ and $D_{(1-b_{(x,y)})}$, with the help of a random index in obtained from (2). Since the key K involved in underlying function F is secret, the attacker cannot obtain the input bit $b_{(x,y)}$ from the corresponding output bit $c_{(x,y)}$. Hence, the security bound can be given as $2^{|K|}$.

In the Extraction-XOR step (Algorithm II), $D_{b_{(x,y)}}$ of length $n = \lambda = d$ is random represented as

$$D_{b_{(x,y)}} = D_{b_{(x,y)_0}} D_{b_{(x,y)_1}} \ldots D_{b_{(x,y)_{(d-1)}}}$$
$$\Longrightarrow D_{b_{(x,y)_0}} \oplus D_{b_{(x,y)_1}} \ldots \oplus D_{b_{(x,y)_{(d-1)}}}$$

is random [from the property of XOR operation]

So, given an unknown input bit $b_{(x,y)}$,

$c_{(x,y)} \longleftarrow b_{(x,y)} \oplus (D_{b_{(x,y)_0}} \oplus D_{b_{(x,y)_1}} \ldots \oplus D_{b_{(x,y)_{(d-1)}}})$ is random [from the property of XOR operation]

Hence, the security is similar to the security of one-time pad. Given the secret parameter K, it is impossible for an attacker to get $b_{(x,y)}$ from the random output bit $c_{(x,y)}$. Hence, irreversibility is achieved. □

From statements 1 and 2, it can be inferred that the proposed encryption scheme S provides both the properties, secrecy and irreversibility over the bound of the secret parameter. Further, each input bit is independent of every other bit and is random. Hence, it never disregards loop invariant condition of the computation theory and also becomes the reason behind the preserving of error rate across the entire iris code.

APPENDIX B
EXAMPLES FOR ALGORITHMS, ALGORITHM I AND ALGORITHM II

In Tables VI and VII, we provide the examples depicting the working of our proposed algorithms, Algorithm I and Algorithm II respectively. Iriscode samples are used in the examples. During the implementation, the input biometric template (with 10239 bits) is traversed in row-major order to give 10239 bit positions starting from 0. For easy understanding and due to space constraints, we took only the 0th and 5th bit positions in both the examples while executing the steps for both the algorithms. Please refer Section II for the notations used in tables.

For Algorithm I in Table VI, the expanded message $M_{b_{(x,y)}}$ is denoted by 2 bytes where the first two bits represent the input bit (including the appended 0) and the remaining 14 bits denotes the bit position of the particular input bit. Whereas in the Table VII for Algorithm II, the bit positions are denoted by 16 bits to give the 2 bytes expanded message $M_{b_{(x,y)}}$. Further, in the implementation, the tweak of length 128 bits is generated out of which 112 bits are appended during the transformation step. In Section VI, we provide the design rationale for each step for the algorithms. From both the examples shown in tables, it can be clearly inferred that bit-wise encryption scheme preserves one-to-one mapping (to preserve the number of bit-errors) while the output bit for a particular input bit remains unpredictable given the key is secret.

REFERENCES

[1] J. Daugman, *600 Million Citizens of India are Now Enrolled With Biometric ID*, vol. 7. Bellingham, WA, USA: SPIE, 2014.

[2] S. Mahajan and A. Deshpande, "Multibiometric template security using fuzzy vault," *Int. J. Comput. Appl.*, vol. 154, no. 3, pp. 21–26, 2008.

[3] C. Li, J. Hu, J. Pieprzyk, and W. Susilo, "A new biocryptosystem-oriented security analysis framework and implementation of multibiometric cryptosystems based on decision level fusion," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 6, pp. 1193–1206, Jun. 2015.

[4] M. Gomez-Barrero, E. Maiorana, J. Galbally, P. Campisi, and J. Fierrez, "Multi-biometric template protection based on homomorphic encryption," *Pattern Recognit.*, vol. 67, pp. 149–163, Jul. 2017.

[5] M. Gomez-Barrero, C. Rathgeb, G. Li, R. Ramachandra, J. Galbally, and C. Busch, "Multi-biometric template protection based on Bloom filters," *Inf. Fusion*, vol. 42, pp. 37–50, Jul. 2018.

[6] A. K. Jain and A. Ross, "Multibiometric systems," *Commun. ACM*, vol. 47, no. 1, pp. 34–40, 2004.

[7] P. Basak, S. De, M. Agarwal, A. Malhotra, M. Vatsa, and R. Singh, "Multimodal biometric recognition for toddlers and pre-school children," in *Proc. IEEE Int. Joint Conf. Biometrics (IJCB)*, Oct. 2017, pp. 627–633.

[8] A. K. Jain and A. Kumar, "Biometric recognition: An overview," in *Proc. 2nd Gener. Biometrics: Ethical, Legal Social Context*. Dordrecht, The Netherlands: Springer, 2012, pp. 49–79.

[9] N. K. Ratha, J. H. Connell, and R. M. Bolle, "Enhancing security and privacy in biometrics-based authentication systems," *IBM Syst. J.*, vol. 40, no. 3, pp. 614–634, 2001.

[10] R. Cappelli, D. Maio, A. Lumini, and D. Maltoni, "Fingerprint image reconstruction from standard templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 9, pp. 1489–1503, Sep. 2007.

[11] J. Galbally, A. Ross, M. Gomez-Barrero, J. Fierrez, and J. Ortega-Garcia, "Iris image reconstruction from binary templates: An efficient probabilistic approach based on genetic algorithms," *Comput. Vis. Image Understand.*, vol. 117, no. 10, pp. 1512–1525, Oct. 2013.

[12] K. Simoens *et al.*, "Criteria towards metrics for benchmarking template protection algorithms," in *Proc. 5th IAPR Int. Conf. Biometrics (ICB)*, Mar. 2012, pp. 498–505.

[13] 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," *Off. J. Eur. Union (OJ)*, vol. 59, nos. 1–88, p. 294, 2016.

[14] *Information Technology–Security Techniques-Biometric Information Protection*, International Organization for Standardization, ISO ISO/IEC Standard 24745:2011(en), 2011.

[15] M. Sandhya and M. V. Prasad, "Biometric template protection: A systematic literature review of approaches and modalities," in *Biometric Security and Privacy*. Cham, Switzerland: Springer, 2017, pp. 323–370.

[16] C. Rathgeb and A. Uhl, "A survey on biometric cryptosystems and cancelable biometrics," *EURASIP J. Inf. Secur.*, vol. 2011, no. 1, p. 3, Dec. 2011.

[17] K. Nandakumar and A. K. Jain, "Biometric template protection: Bridging the performance gap between theory and practice," *IEEE Signal Process. Mag.*, vol. 32, no. 5, pp. 88–100, Sep. 2015.

[18] I. Natgunanathan, A. Mehmood, Y. Xiang, G. Beliakov, and J. Yearwood, "Protection of privacy in biometric data," *IEEE Access*, vol. 4, pp. 880–892, 2016.

[19] A. Juels and M. Sudan, "A fuzzy vault scheme," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun./Jul. 2002, p. 408.

[20] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 2004, pp. 523–540.

[21] A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in *Proc. 6th ACM Conf. Comput. Commun. Secur. (CCS)*, 1999, pp. 28–36.

[22] K. Xi and J. Hu, "Bio-cryptography," in *Handbook of Information and Communication Security*. 2010, pp. 129–157.

[23] C. Rathgeb and A. Uhl, "The state-of-the-art in iris biometric cryptosystems," in *State of the art in Biometrics*. London, U.K.: IntechOpen, 2011, pp. 179–202.

[24] J. Zuo, N. K. Ratha, and J. H. Connell, "Cancelable iris biometric," in *Proc. 19th Int. Conf. Pattern Recognit.* Tampa, FL, USA: IEEE, Dec. 2008, pp. 1–4.

[25] V. M. Patel, N. K. Ratha, and R. Chellappa, "Cancelable biometrics: A review," *IEEE Signal Process. Mag.*, vol. 32, no. 5, pp. 54–65, Sep. 2015.

[26] J. H. Cheon, H. Chung, M. Kim, and K.-W. Lee, "Ghostshell: Secure biometric authentication using integrity-based homomorphic evaluations," IACR Cryptol. ePrint Arch., Tech. Rep. 2016/484, 2016, p. 484.

[27] M. Barni, G. Droandi, and R. Lazzeretti, "Privacy protection in biometric-based recognition systems: A marriage between cryptography and signal processing," *IEEE Signal Process. Mag.*, vol. 32, no. 5, pp. 66–76, Sep. 2015.

[28] S. Kanade, D. Petrovska-Delacretaz, and B. Dorizzi, "Obtaining cryptographic keys using feature level fusion of iris and face biometrics for secure user authentication," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (Workshops)*, Jun. 2010, pp. 138–145.

[29] A. Nagar, K. Nandakumar, and A. K. Jain, "Multibiometric cryptosystems based on feature-level fusion," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 1, pp. 255–268, Feb. 2012.

[30] Y. Sutcu, Q. Li, and N. Memon, "Secure biometric templates from fingerprint-face features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–6.

[31] S. C. Dass, K. Nandakumar, and A. K. Jain, "A principled approach to score level fusion in multimodal biometric systems," in *Audio- and Video-Based Biometric Person Authentication*, T. Kanade, A. Jain, and N. K. Ratha, Eds. Berlin, Germany: Springer, 2005, pp. 1049–1058.

[32] R. Dwivedi and S. Dey, "Score-level fusion for cancelable multi-biometric verification," *Pattern Recognit. Lett.*, vol. 126, pp. 58–67, Sep. 2019.

[33] H. Wallace, "Error detection and correction using the BCH code," *EBook UNDUH*, 2001.

[34] X. Boyen, "Reusable cryptographic fuzzy extractors," in *Proc. 11th ACM Conf. Comput. Commun. Secur. (CCS)*, 2004, pp. 82–91.

[35] R. Canetti, B. Fuller, O. Paneth, L. Reyzin, and A. Smith, "Reusable fuzzy extractors for low-entropy distributions," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 2016, pp. 117–146.

[36] J. H. Cheon *et al.*, "A reusable fuzzy extractor with practical storage size: Modifying Canetti et al.'s construction," in *Proc. Australas. Conf. Inf. Secur. Privacy*. Cham, Switzerland: Springer, 2018, pp. 28–44.

[37] M. Dworkin, "Recommendation for block cipher modes of operation. methods and techniques," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. 800-38A, 2001.

[38] C. Rathgeb and C. Busch, "Cancelable multi-biometrics: Mixing iris-codes based on adaptive Bloom filters," *Comput. Secur.*, vol. 42, pp. 1–12, May 2014.

[39] M. Gomez-Barrero, C. Rathgeb, J. Galbally, C. Busch, and J. Fierrez, "Unlinkable and irreversible biometric template protection based on Bloom filters," *Inf. Sci.*, vols. 370–371, pp. 18–32, Nov. 2016.

[40] S. Cimato, M. Gamassi, V. Piuri, R. Sassi, and F. Scotti, "Privacy-aware biometrics: Design and implementation of a multimodal verification system," in *Proc. Annu. Comput. Secur. Appl. Conf. (ACSAC)*, Dec. 2008, pp. 130–139.

[41] C. Fang, Q. Li, and E.-C. Chang, "Secure sketch for multiple secrets," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.* Berlin, Germany: Springer, 2010, pp. 367–383.

[42] J. Bringer, H. Chabanne, and B. Kindarji, "The best of both worlds: Applying secure sketches to cancelable biometrics," *Sci. Comput. Program.*, vol. 74, nos. 1–2, pp. 43–51, Dec. 2008.

[43] O. Ouda, N. Tsumura, and T. Nakaguchi, "Tokenless cancelable biometrics scheme for protecting iris codes," in *Proc. 20th Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 882–885.

[44] C. Rathgeb, H. Baier, F. Breitinger, and C. Busch, "On application of Bloom filters to iris biometrics," *IET Biometrics*, vol. 3, no. 4, pp. 207–218, Dec. 2014.

[45] P. Drozdowski, C. Rathgeb, and C. Busch, "Bloom filter-based search structures for indexing and retrieving iris-codes," *IET Biometrics*, vol. 7, no. 3, pp. 260–268, May 2018.

[46] C. Rathgeb, F. Breitinger, H. Baier, and C. Busch, "Towards Bloom filter-based indexing of iris biometric data," in *Proc. Int. Conf. Biometrics (ICB)*, May 2015, pp. 422–429.

[47] P. Drozdowski, S. Garg, C. Rathgeb, M. Gomez-Barrcro, D. Chang, and C. Busch, "Privacy-preserving indexing of iris-codes with cancelable Bloom filter-based search structures," in *Proc. 26th Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2018, pp. 2360–2364.

[48] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On data banks and privacy homomorphisms," *Found. Secure Comput.*, vol. 4, no. 11, pp. 169–180, 1978.

[49] K. Zhou and J. Ren, "PassBio: Privacy-preserving user-centric biometric authentication," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 12, pp. 3050–3063, Dec. 2018.

[50] J. Daugman, "The importance of being random: Statistical principles of iris recognition," *Pattern Recognit.*, vol. 36, no. 2, pp. 279–291, Feb. 2003.

[51] M. Inuma, A. Otsuka, and H. Imai, "Theoretical framework for constructing matching algorithms in biometric authentication systems," in *Proc. Int. Conf. Biometrics*. Berlin, Germany: Springer, 2009, pp. 806–815.

[52] A. Rattani and N. Poh, "Biometric system design under zero and non-zero effort attacks," in *Proc. Int. Conf. Biometrics (ICB)*, Jun. 2013, pp. 1–8.

[53] A. Kumar and A. Passi, "Comparison and combination of iris matchers for reliable personal authentication," *Pattern Recognit.*, vol. 43, no. 3, pp. 1016–1026, Mar. 2010.

[54] J. Ortega-Garcia *et al.*, "The multiscenario multienvironment BioSecure multimodal database (BMDB)," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 6, pp. 1097–1111, Jun. 2010.

[55] N. Othman, B. Dorizzi, and S. Garcia-Salicetti, "OSIRIS: An open source iris recognition software," *Pattern Recognit. Lett.*, vol. 82, pp. 124–131, Oct. 2016.

[56] C. Rathgeb, A. Uhl, P. Wild, and H. Hofbauer, "Design decisions for an iris recognition SDK," in *Handbook of Iris Recognition*. London, U.K.: Springer, 2016, pp. 359–396.

[57] L. Masek, "Recognition of human iris patterns for biometric identification," M.S. thesis, Univ. Western Australia, Perth, WA, Australia, 2003.

[58] A. Anjos, M. Günther, T. de Freitas Pereira, P. Korshunov, A. Mohammadi, and S. Marcel, "Continuously reproducing toolchains in pattern recognition and machine learning experiments," in *Int. Conf. Mach. Learn. (ICML)*, Aug. 2017, pp. 1–8. [Online]. Available: http://publications.idiap.ch/downloads/papers/2017/Anjos_ICML2017-2_2017.pdf

[59] A. Anjos, L. E. Shafey, R. Wallace, M. Günther, C. McCool, and S. Marcel, "Bob: A free signal processing and machine learning toolbox for researchers," in *Proc. 20th ACM Conf. Multimedia Syst. (ACMMM)*, Oct. 2012, pp. 1449–1452. [Online]. Available: https://publications.idiap.ch/downloads/papers/2012/Anjos_Bob_ACMMM12.pdf

[60] *Neurotechnology Verieye SDK, version: 11.2*. Accessed: Feb. 1, 2020. [Online]. Available: https//:www.neurotechnology.com/verieye.html

[61] *Neurotechnology Verilook SDK, version: 11.2*. Accessed: Feb. 1, 2020. [Online]. Available: https://www.neurotechnology.com/verilook.html

[62] S. Kanade, D. Camara, E. Krichen, D. Petrovska-Delacretaz, and B. Dorizzi, "Three factor scheme for biometric-based cryptographic key regeneration using iris," in *Proc. Biometrics Symp.*, Sep. 2008, pp. 59–64.

[63] F. Hao, R. Anderson, and J. Daugman, "Combining crypto with biometrics effectively," *IEEE Trans. Comput.*, vol. 55, no. 9, pp. 1081–1088, Sep. 2006.

[64] R. C. Bose and D. K. Ray-Chaudhuri, "On a class of error correcting binary group codes," *Inf. Control*, vol. 3, no. 1, pp. 68–79, Mar. 1960.

[65] F. J. MacWilliams and N. J. A. Sloane, *The Theory Error-Correcting Codes*, vol. 16. Amsterdam, The Netherlands: Elsevier, 1977.

[66] M. He *et al.*, "Performance evaluation of score level fusion in multimodal biometric systems," *Pattern Recognit.*, vol. 43, no. 5, pp. 1789–1800, May 2010.

[67] M. Gomez-Barrero, J. Galbally, C. Rathgeb, and C. Busch, "General framework to evaluate unlinkability in biometric template protection systems," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 6, pp. 1406–1420, Jun. 2018.

[68] R. Raghavendra, K. B. Raja, and C. Busch, "Presentation attack detection for face recognition using light field camera," *IEEE Trans. Image Process.*, vol. 24, no. 3, pp. 1060–1075, Mar. 2015.

[69] R. Raghavendra and C. Busch, "Robust scheme for iris presentation attack detection using multiscale binarized statistical image features," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 4, pp. 703–715, Apr. 2015.

[70] J. Hernandez-Ortega, J. Fierrez, A. Morales, and J. Galbally, "Introduction to face presentation attack detection," in *Handbook of Biometric Anti-Spoofing*. Cham, Switzerland: Springer, 2019, pp. 187–206.

[71] B. Toth, *Liveness Detection: Iris*. Boston, MA, USA: Springer, 2009, pp. 931–938. [Online]. Available: https://doi.org/10.1007/978-0-387-73003-5_179

[72] *Information Technology-Trusted Platform Module Library Part 1: Architecture*, Int. Org. Standardization, Geneva, Switzerland, 2015.

[73] J. Hermans, B. Mennink, and R. Peeters, "When a Bloom filter is a doom filter: Security assessment of a novel iris biometric template protection system," in *Proc. Int. Conf. Biometrics Special Interest Group (BIOSIG)*, Sep. 2014, pp. 1–6.

[74] *Aadhaar Authentication API Specification, Version: 2.0*. Accessed: Mar. 5, 2020. [Online]. Available: https://uidai.gov.in/images/FrontPageUpdates/aadhaar_authentication_api_2_0.pdf

[75] J. Kornblum, "Identifying almost identical files using context triggered piecewise hashing," *Digit. Invest.*, vol. 3, pp. 91–97, Sep. 2006.