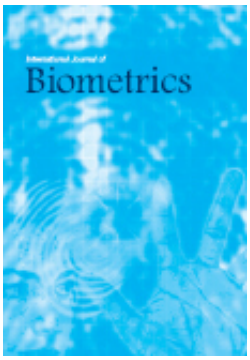


International Journal of Biometrics

ISSN online: 1755-831X - ISSN print: 1755-8301
https://www.inderscience.com/ijbm

Revocable iris templates using partial sort and randomised look-up table mapping

Mulagala Sandhya, Dilip Kumar Vallabhadas, Shubham Rathod

DOI: 10.1504/IJBM.2023.10051691

Article History:

Received:	05 August 2020
Last revised:	19 August 2021
Accepted:	23 August 2021
Published online:	15 December 2022

Revocable iris templates using partial sort and randomised look-up table mapping

Mulagala Sandhya*,
Dilip Kumar Vallabhadas and
Shubham Rathod

Department of Computer Science and Engineering,
National Institute of Technology Warangal,
Telangana, India
Email: msandhya@nitw.ac.in
Email: dilip.kumar218@gmail.com
Email: shubrathod44@gmail.com
*Corresponding author

Abstract: In the ongoing years, biometric systems end up helpless against the spillage of template information. If a biometric template is stolen, it is lost permanently and cannot be restored or reissued. Here, we use iris biometric because of its high accuracy. In this paper, we develop a new cancellable biometric scheme using the indexing-first-one (IFO) hashing coupled with a technique called partial sort. The IFO hashing uses new mechanisms called the P-order Hadamard product and modulo threshold function paired with the partial sort technique which has considerably strengthened it further. We used the very sophisticated CASIA-v3 database which provides us with a wide range of iris templates for our experiments. As compared to the previous cancellable schemes, the analysis of the results of these experiments provides us with good accuracy and strong resistance to various privacy and security attacks.

Keywords: iris; cancellable template; min-hashing; Jaccard similarity; security; privacy; partial sort; look-up table.

Reference to this paper should be made as follows: Sandhya, M., Vallabhadas, D.K. and Rathod, S. (2023) 'Revocable iris templates using partial sort and randomised look-up table mapping', *Int. J. Biometrics*, Vol. 15, No. 1, pp.21–39.

Biographical notes: Mulagala Sandhya received PhD from the School of Computer and Information Sciences, University of Hyderabad. She is currently working as an Assistant Professor in the Department of Computer Science Engineering at the National Institute of Technology Warangal. Her research interests include biometrics, image processing, pattern recognition, and machine learning.

Dilip Kumar Vallabhadas received MTech from National Institute of Technology Rourkela. He is currently a Research Scholar in the Department of Computer Science Engineering at the National Institute of Technology Warangal. His research interests include cancellable biometrics, biometric security, deep learning, and blockchain.

Shubham Rathod received MTech from the Department of Computer Science Engineering at the National Institute of Technology Warangal. His research interests include biometrics, and machine learning.

1 Introduction

In the Greek words bio (life) and metric (to measure), the expression biometric is derived. Biometric refers to our use to progress in the evaluation and examination of the psychological or social properties of a person. These attributes are novel to people, henceforth can be utilised to confirm or distinguish an individual (Lai et al., 2017).

1.1 Traditional biometric recognition

Traditional mechanisms for verification of identity made use of credentials that we have to remember like alphanumeric passwords, PINs, access cards. The problem with this approach is that these credentials can be easily forgotten by the individual or stolen or lost. Whereas biometrics of an individual like a fingerprint, palmprint, face, iris, etc. are inherently associated with the individual and prove very efficient for identity verification. There are various biometric traits that can be used for biometric recognition, but among all these the iris is the most effective because of its high performance and accuracy. Even though an individual cannot forget or lose his/her biometric, if we store the biometric templates in some database they are vulnerable to many security and privacy attacks (Sandhya and Prasad, 2017). To protect these templates stored in the database, we apply the mechanism of cancellable biometrics. A number of cancellable biometric schemes have been proposed but they have a very low degree of accuracy performance.

1.2 Need of cancellable biometrics

Biometrics is a survey on the perception, For example, surveillance, authentication, safety, and access control, depending on their psychological and behavioural characteristics in a variety of applications. Biometrics, for example are not exposed to blankness or misfortune in comparison to conventional authentication, for example, passwords or tokens. They are hard to fake. Regardless of their diverse preferences, and inadequate use of stored digital representations, such as fingerprint or iris, can present serious safety problems (Rafiq and Selwal, 2019). However, frameworks with raw unprotected biometric characteristics for recognition often suffer the negative results of issues such as spoof attacks, lack of characteristics, low accuracy of recognition, and biometrics information variation. A cancellable-biometric framework that uses a biometric template which is stored in the database using some cancellable biometrics scheme, such as fingerprint, palm print, face, iris and finger vein. It results in a more precise recognition, but to solve these problems is also harder to trick or assault. We cannot reset or replace a compromised biometric template (Ratha et al., 2007). Hence, an important protection template technique is a cancellable biometrics process that performs a one-way transform for verifying the original biometric data. Mathematically, this one-way transform can not be inverted, and only changing transformation parameters can effectively revoke and replace this trading template. But developing such a cancellable biometric scheme is tough as it should be truly one way and should be immune to various security attacks (Priya et al., 2015). The accuracy of verification and the speed of verification must be maintained in order to facilitate quick identification (Gupta and Sehgal, 2016). Biometric template protection technology is commonly classified as biometric cryptosystems and cancellable biometrics. Following are the four criteria that

are expected from a reliable and efficient cancellable biometric scheme (Ratha et al., 2007):

1 Unlinkability: It should be difficult for the adversary to make a difference that a single source (same users biometric) is used for the generation of one or more protected templates. Cross-matching across different applications can occur hence the unlinkability is necessary to avoid this.

2 Revocability: If the adversary gets his hands on multiple protected templates of the iris then it may be possible by using the computational power at his disposal to derive the original contents of the iris image which in a way invades the privacy of the iris template which is unacceptable. Revocability ensures that this does not happen. This also helps in the revocation or renewal of the new template so that we can replace the old template along with ensuring that the adversary does not get the original template.

3 Non-invertible: If any protected template is compromised, then it should not be in the computational power of the adversary to obtain the original contents of the iris image which in turn increases the security of the system and makes sure that the compromised biometric data is not abused (Sandhya and Prasad, 2018).

4 Performance: Cancellable template's performance should be preserved according to the original contents of the iris image.

In this paper, we present a technique for creating a cancellable biometric scheme using the partial sort which is combined with a technique called indexing-first-one (IFO) hashing. In partial sorting, we divide the iris code into several fixed-size windows which are in turn divide into small windows and sorted partially. IFO is a technique that uses the concept of min-hashing, i.e., it takes the first occurrence of 1 from the random permuted iris code. This IFO technique is applied to the partially sorted iris code. This code is linked to a look-up table from which we are going to generate a binary cancellable iris code that is non-inevitable. Our results show that our proposed technique is more accurate when compared to Lai et al. (2017) which gives an equal error rate (EER) of 0.54.

In Lai et al. (2017) the EER of the proposed model is sufficiently low and the authors concentrated on revocability and unlinkability of the method. Our proposed method has sufficiently less EER and satisfies all the requirements of cancellable biometrics, i.e., revocability, irreversibility, diversity, and efficiency.

The dataset that we are going to use is the CASIA-v3 (CASIA Iris Database, http://www.cbsr.ia.ac.cn/english/IrisDatabase.asp) database that contains a plethora of iris images all taken in different lighting conditions, different types of subjects, etc. For example, the database contains the iris images of left and right eyes classified in different folders, the iris templates of twins in order to check the performance of the cancellable iris scheme with all types of subjects.

The rest of the paper is organised as follows: Section 2 gives a brief overview of the work done on template protection. The background knowledge required for our method is discussed in Section 3. Section 4 presents our proposed method. Experimental results and analysis is explained in Section 5 which is followed by conclusions in Section 6.

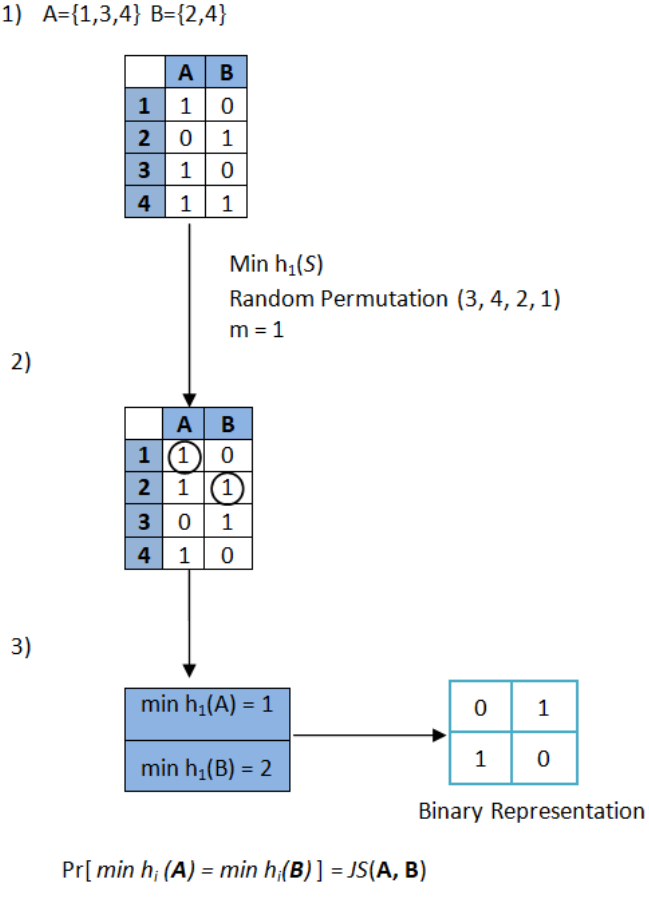
2 Related work

This section covers the previous work that is done on template protection, the concept of the cancellable template generation was first introduced by Ratha et al. (2007). Adamović et al. (2020) applied machine learning techniques to recognise Iris by omitting Gabor wavelets. Iris templates are represented in 1D space followed by stylometric feature extraction. Gupta and Sehgal (2019) given a method to protect the Iris recognition system against the template and replay attacks. Dwivedi et al. (2017) used look-up table mapping for Iris encoding. Zuo et al. (2008), etc. by applying various different operations like convolution on the feature vector generated by the Gabor filter. The salting approach provided a good accuracy performance but the major problem with the salting approach was that it had weak non-invertibility. The weak non-invertibility was found in the case of the stolen token situation where if the unique password of the user is stolen the adversary can easily invert the cancellable template and find out the original iris information. Hence, it is required that the auxiliary data be kept safe forever which is not practical. As the salting approach lacks non-invertibility, there was a need for a more non-invertible approach. The next subsection explains the approach that was proposed to increase the non-invertibility of the salting approach. To overcome the weak non-invertibility of the salting approach the non-invertible transformation technique was developed. In this technique, the iris template is converted into a template that is non-invertible with a transformation function which is one-way so that the new transformed template can be stored in the database securely. BIN-COMBO and GREY-COMBO were the two transformation methods proposed by Zuo et al. (2008) for non-invertible iris template transformation. In the GREY-COMBO method what they did was a shift in a row-wise manner the iris image via a parameter called the random offset. Then they randomly selected two rows and performed arithmetic operations on them like multiplication or addition. While in the BIN-COMBO method the procedure applied on the iris code was the same but the operations were logical rather than arithmetic which was XNOR or XOR. Hence, the above two techniques can produce cancellable iris templates which fulfil the criteria of non-invertibility, as the original iris data is distorted because of the arithmetic and logical operations applied on it. Still, the techniques were not perfect because they had flaws like the first technique gave a degraded performance if the iris images used were of bad quality and as the techniques used the password which is unique to the user, the technique got vulnerable to the stolen token situation like the salting approach. Later, Rathgeb et al. (2014) applied bloom filters to protect Iris templates. Lai et al. (2017) used index first one hashing to protect Iris templates. As Jenisch and Uhl (2011) stated in his security analysis on IrisCode, when we match an IrisCode with any sparse binary code we will get nearly a 50% match between the IrisCode and the sparse binary code. This means that half of the bits of two uncorrelated IrisCode will be matched. In the biometric salting, approach (Zuo et al., 2008) accuracy performance discrepancy was the problem in the scenarios of stolen and genuine tokens (Chong et al., 2006). Achieving revocability should be possible in any template protection scheme in which there is a requirement of auxiliary data which is independent.

3 Background

The main function of *min hashing* technique is to determine how similar two sets are quickly (Broder, 1997). The min-hashing is used on large binary vectors to locate the position where the bit '1' occurred first and record for a large number of permutations of these binary vectors. The varied seeds of permutation on the binary vectors can be used to encode different index vectors into binary form (Broder et al., 2000). 1 explains the min hashing algorithm where we consider two set A = 1, 3, 4 and B = 2, 4. First, we form a matrix by placing 1 at elements specified in the set and remaining places as 0, i.e., for set A place 1, 3, 4 we assign 1, and for place 2 we assign 0. In the next step, we perform random permutation in our example it is (3, 4, 2, 1). Based on the permutations the values in the matrix will also change. The changed matrix is shown in step 2. From this matrix, we consider the occurrence of the first 1 in each set which is the min hash value for set A it is 1 and for set B it is 2. These min hash values are represented in binary form.

Figure 1 A working example of the min-hashing algorithm (see online version for colours)



Consider a binary vector, generate two index vectors from it, say A and B. Let the members of the vectors A and B be mapped to different indexes by a hash function, say h. Suppose S is a set then the minimal member of this set wrt h is given by min$h(S)$, for any set like S. Let A and B both be applied by min$h()$. After doing so only if $A \cap B$ contains elements of $A \cup B$ we will get min$h(A)$ = min$h(B)$. It is a matter of probability of this being true which constitutes another term called the hash collision rate. The hash collision rate is the ratio of $|A \cup B|$ and $|A \cap B|$. Hence, the formula can be generated as,

$$\Pr\left[\min h_i(A) = \min h_i(B)\right] = JS(A, B) \pm \varepsilon \tag{1}$$

where ε is called as an estimation error for $i = 1\ldots\ldots m$ where the number of $h(.)$ is m. $JS(A, B)$ is called as the Jaccard similarity which can be expressed as,

$$JS(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{2}$$

The value of JS varies as $0 \leq JS \leq 1$ and $JS = 1$. If we increase the size of the hashed code storage in order to increase the value of m, we can considerably reduce the error. Figure 1 demonstrates the working of the min-hashing algorithm. The iris code generation is a multi-step process. Firstly, we use the weighted adaptive Hough transformation to detect the iris region. The iris and the pupil boundaries are then segmented using a two-stage segmentation process (Uhl and Wild, 2012). The iris region is unwrapped using a normalisation process. It is unwrapped into a fixed dimension array which has a size of 64 * 512. The normalisation process used here is the rubber sheet model. From this array of size 64 * 512, the last 14 rows are dropped to form an array iris texture of size 50 * 512 pixels. Then a one-dimensional vector is formed by averaging pixels of every five rows into one. Hence, we get an iris texture of a size of 10 * 512. Then we get a complex iris Gabor-features, by convoluting each vector with a 1D log Gabor filter, of size 10 * 512. Finally, we get the IrisCode $X \varepsilon \{0, 1\}^{n1*n2}$, with $n_1 = 20$ and $n_2 = 512$ by phase quantising into 2 bits every value that is complex in the iris Gabor features.

4 Proposed method

Figure 2 shows the proposed model. Each block in this diagram represents a step in the cancellable iris template generation process. The first block in Figure 2 represents the iris CASIA-v3 database. The next block is the segmentation and normalisation which are the initial stages of iris image processing.

4.1 Segmentation

In the iris recognition system, there are various stages of which the first stage is segmentation. Segmentation means extracting the iris image from the given eye image. This is done by recognising the locations of the various eye components like the pupil, eyelids, and eyelashes. Moreover, in the eye image, there may be unnecessary interference like the occlusion due to eyelids, segmentation removes the occlusion. The iris region in the eye is the portion between the outer boundary of the pupil and the inner boundary of the sclera (Uhl and Wild, 2012). To extract the iris, we need to approximately mark two circles with the centre of the pupil as the centre of the circles

and the circumference of the first circle as the outer boundary of the pupil, and the circumference of the second circle as the inner sclera boundary. In marking the circles there might be a possibility that the upper and lower eyelids and eyelashes might come in the picture. So these are needed to be eliminated carefully. The segmentation stage is very important in the whole iris recognition system because if the results of this stage are not correct, then the iris template that is generated using these results will be corrupted which will, in turn, affect the biometric system as a whole. The quality of the images of the eye that are considered for research also affects the effectiveness of the segmentation process. We used the weighted adaptive Hough algorithm and ellipsopolar transform technique for the segmentation of iris using the USIT Iris toolkitv2 (Rathgeb et al., 2016). The results of the segmentation using the adaptive Hough algorithm and ellipsopolar transform technique are shown in Table 1.

Figure 2 Proposed model for cancellable iris template generation

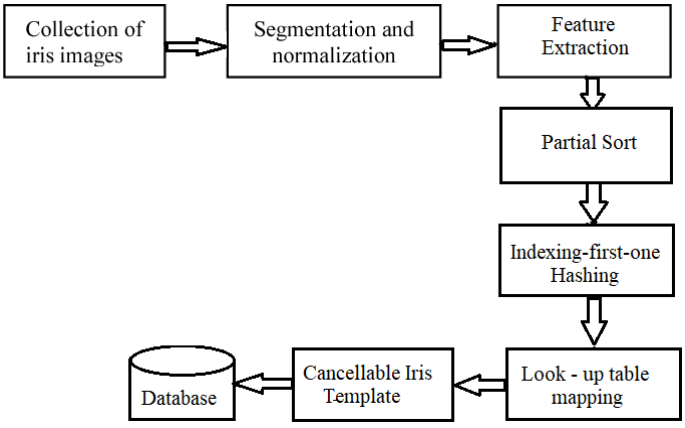


Table 1 Resultant EER for adaptive Hough algorithm and ellipsopolar transform technique (AHAET)

Equal error rate (EER)			
Technique	CASIA-v1	CASIA-v2	CASIA-v3
AHAET	2.85	3.10	2.80

4.2 Normalisation

Once the iris image is segmented, we need to prepare it for feature extraction. This process is called normalisation. The distance and the angle of the iris wrt camera highly affects images taken of the iris especially in the Cartesian coordinates. Also, the intensity of the light falling on the eye causes variations in the patterns of the iris which are nonlinear, as the pupil expands or contracts depending on it (Uhl and Wild, 2012). We need an efficient normalisation method so that we can tackle these variations and transform the iris image properly. Here, for ease of comparison, we align an image that is already in the database with an image that is acquired newly. Here, we use an approach for normalisation which is different from the method of Daugman's rubber sheet model (Johar and Kaushik, 2015).

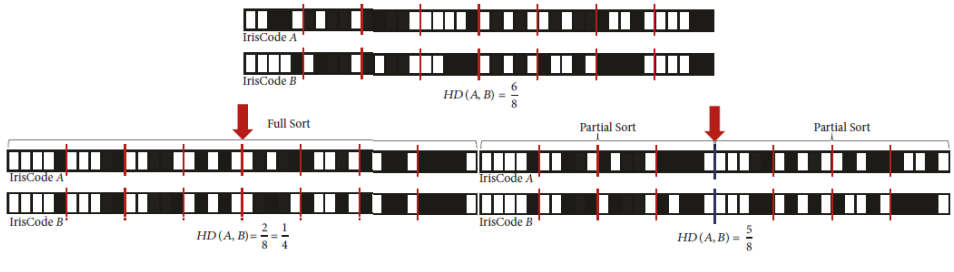
4.3 Feature extraction

The Iris features to be extracted are supposed to be in the form of iris code, which eases the further implementation of the algorithm. We used USIT toolkitv2 to apply scale invariant feature transform (Rathgeb et al., 2016) to generate Iris code from iris texture. Finally, we need to extract the binary information from the image. This gives us an iris code of dimensions 1 * 10,240 for each image.

4.4 Partial sort

Partial sort is sorting the iris code vector based on some parameters. There are two types of sorts: the partial sort and the full sort. In full sorting, we sort the iris code fully while in partial sort we divide the iris code and then sort the divided parts, hence called the partial sort (Jeong and Jeong, 2019). The difference between the partial and full sort can be explained in the following example. In the shown example the Hamming distance between the two iris codes A and B is 6/8 without sorting. Now we perform fully as well as partial sorting on them. After performing the full sorting we get a Hamming distance of 1/4 which is because the blocks which are different between the two iris codes have decreased considerably in number. Whereas, after performing the partial sort we get a Hamming distance of 5/8 because the blocks which are different between the two iris codes have reduced slightly in number. A pictorial depiction of the above example is shown in Figure 3. There are steps in the partial sort technique. First, we divide the iris code which is in the form of a row vector into blocks called the s-blocks each of size p. Then we further divide the s-blocks into smaller blocks called the 'u-blocks' each of size q. Then in every s-block, we sort $p = q$ number of u-blocks. Hence, instead of sorting the full iris code, we sort the iris code in parts which is exactly the partial sorting that has been discussed.

Figure 3 Difference between partial sort and full sort (see online version for colours)



4.5 Partial sort algorithm

In Algorithm 1, we will depict the partial sort algorithm for row vectors along with proper explanation and suitable examples.

H: *Partially sorted form of S, i.e., S'*
Input: s-block of size p, u-block of size q, row vector S
k = size of S / p
1 = *p / q*
1 *In every s-block u-block must be sorted*
for j = 0 *to k* – 1 *do*
2 *Take q-bit arrays and update them with integers variables*
for n = 0 *to l* – 1 *do*
B[n] = 0
C[n] = 0
End for
3 *Convert the bits in each u-block to an integer*
for n = 0 *to l* – 1 *do*
copy q bits from S(1 + (*l* * *q*) + (*j* * *p*)) *into B[n]*
End for
*sort (B[*0*], …, B[l* – 1*]) into (C[*0*], …, C[l* – 1*])*
4 *Update the s-block that is not sorted with the sorted one*
for n = 0 *to l* – 1 *do*
put C[n] into S'(1 + (*n* * *q*) + (*j* * *p*))
End for
End for

4.5.1 Partial sort

The steps of Algorithm 1 are as follows: In Algorithm 1, we input the iris code as a row vector *S*, *s*-block of size *p*, *u*-block of size *q*. Then we calculate some parameters as *r* = size of (*S*) / *p*, *c* = *p* = *q*. Now,

1 For each *s*-block we need to sort the *u*-blocks in it. Hence, we run a loop for each *s*-block and perform the following steps.

2 Take two temporary arrays of size *q* and initialise them with 0.

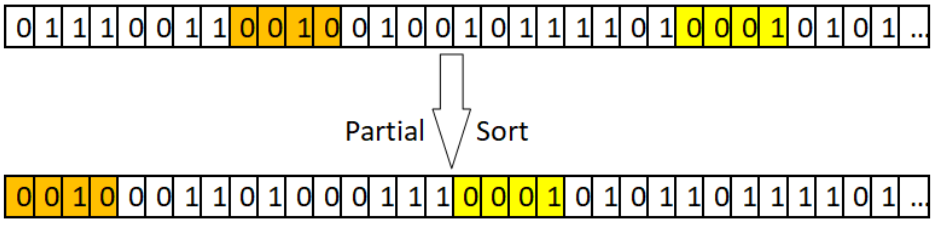
3 Then for each *u*-block in the *s*-block copy the *q*-bits into any one of the temporary arrays. Then convert this array of *q*-bits into its decimal form.

4 Store this converted decimal numbers into the second temporary array, hence for each *s*-block we will get an array with integers that are the decimal from of each *u*-block.

5 Sort this temporary array of decimal numbers and convert the decimal into its binary form. Simultaneously, replace the original array with these binary bits one by one according to the decimal numbers. Hence, we will get a sorted *s*-block, and eventually, after applying the same process to each sort block we will get a partially sorted array.

The working example of the algorithm is given in Figure 4 with the initial parameters $p = 16$, $q = 4$ that gives us $c = 4$. We can see that the row vector is divided into s-blocks of size 16 and then the s-blocks are further divided into u-blocks of size 4. The u-blocks are then sorted for each s-block. Hence, we get a partially sorted row vector from the original vector.

Figure 4 An example of partial sort (see online version for colours)



4.6 IFO hashing

The IFO hashing is an update to the min-hashing algorithm which is formed by adding three operations to the original algorithm called as the partial sort, P-Hadamard product and the modulo threshold function. Just like the min-hashing, the algorithm makes use of m hash functions which are independent of each other. These hash functions are developed from P number permuted IrisCode which are tokenised. The range of both m and P is [1, ∞]. The algorithm is shown in Algorithm 2.

H: *IFO hashed code,* R_I'', $R_{Ij}'' = \{R_{Ij}'' \mid j = 1, \ldots\ldots, m\}$, $R_X'' \varepsilon [0, K - \tau - 1]$ *Input: Size of window K, token of permutation* $\theta_{(i,1)}$, *no. of permutations m, IrisCode* $X \varepsilon \{0, 1\}^{n1*n2}$, *security threshold* τ

For every row of iris code I:

for j = 1 *to m*

Make initial value of i^{th} *code* R_i *to* 0 *which is hashed*

1 *Perform partial sort on the iris code using the partial sort algorithm*

2 *Generate permutations of I wrt* $\theta(i, 1)$; $l\varepsilon[1, P]$

3 *Compute Hadamard product of the permutations as* $(X^P) = \prod_{l=1}^{p} (X_l')$

4 *Form a window of length k*
for l = 1 *to k*

5 *Note the location where bit '*1*' occurs first*
if $X^P(j) > R_i(j)$, *then* $R_i = j$
End if

6 *Perform Modulo thresholding as* $R'i = R_i \bmod (K - \tau)$

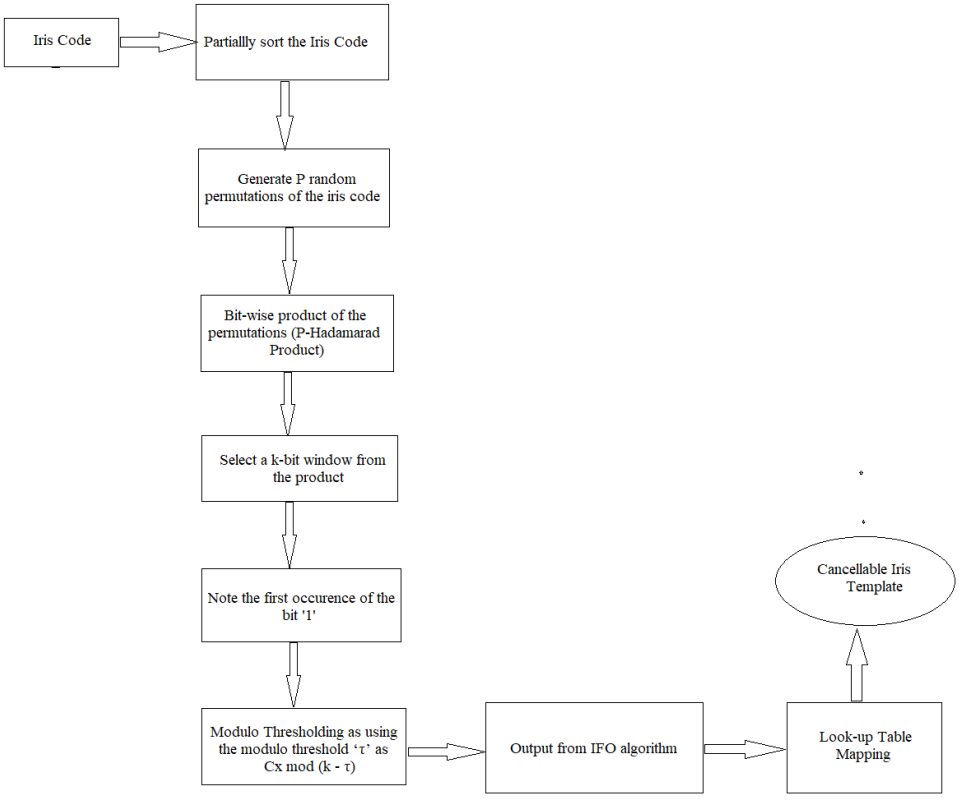
7 *Generate an entry from the Look-up table for R'i as* R_i''

End for
End for

4.6.1 Modified IFO hashing algorithm

A detailed flowchart of the above listed algorithm is in Figure 5 An example of the algorithm is given in Figure 6 with initial parameters values as $k = 3$, $m = 3$, $P = 2$ and $\tau = 2$ where k is the window size, m is the number of permutations of the iris code, P is the degree of the Hadamard product and τ is the security threshold.

Figure 5 Modified IFO hashing algorithm



4.7 Mapping using look-up table

In this paper, we construct a look-up table incorporated from Jeong and Jeong (2019) that is randomly generated to avoid any vulnerabilities in case of an attack. The look-up table is a matrix of bits 0 and 1 with dimensions as 2^q x q. So for example we have $q = 4$, we will get a look-up table of dimensions 16×4. All the bits in the look-up table are generated randomly using some random function. The main function of the look-up table is that the template of decimal numbers obtained as output from the IFO algorithm is mapped to the entries of the table. For the mapping we select a parameter say d where $d \leq q$. Then depending on the value of d, we map the template to the entries of the look-up table. This can be understood more properly with an example. In Figure 7, we have a template of decimal numbers mapped to the entries of the look-up table and a cancellable iris template is generated also we have considered the value of d as 2. The

cancellable iris template will be *d* times longer in size than the template generated from the IFO hashing algorithm.

Figure 6 Example of modified IFO hashing algorithm

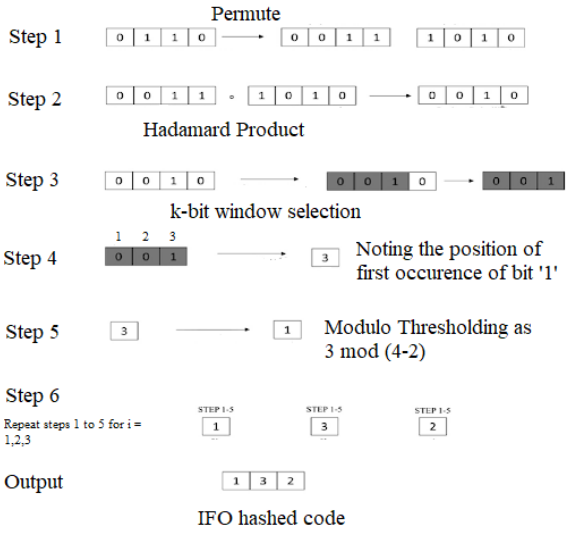
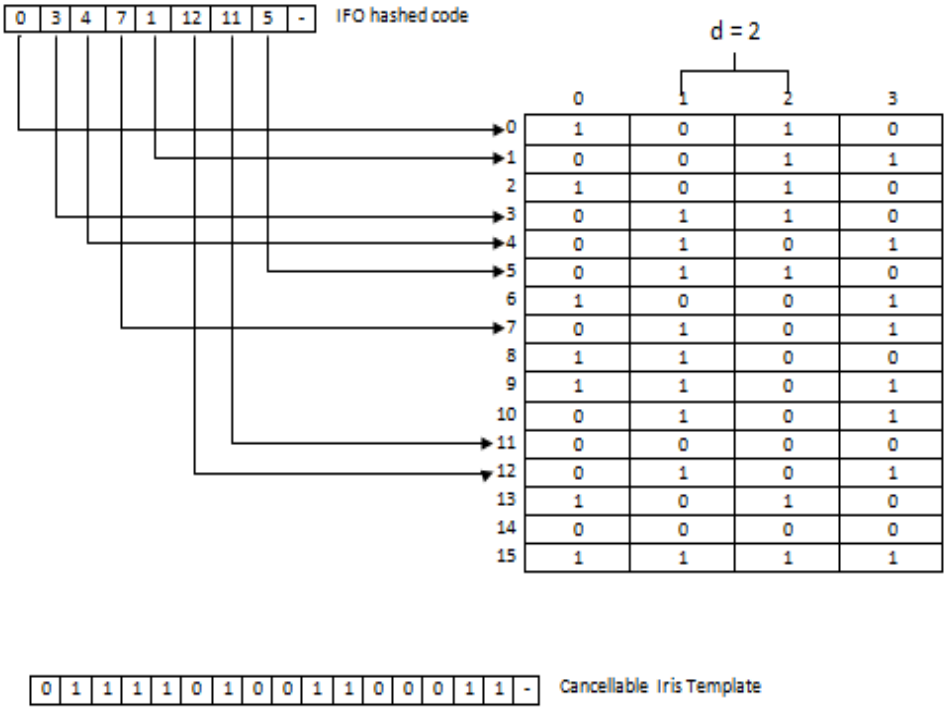


Figure 7 An example of look-up table mapping



4.8 Implementation of the proposed method

We take an iris image as input and perform the pre-processing on that image of segmentation, normalisation, and feature extraction. After the pre-processing of the iris image is done we get the Iris code for that image which is 10,240 bits long. We perform the partial sort on the obtained iris code using the partial sort algorithm. Post to this step we predefined the values of k which is the window size depending upon which we will throw the bits from the Iris code out in order to introduce more distortion, and 'τ' which is the modulo threshold function. Also, we decide the number of pairs of permutations that we are going to perform on the iris code which will subsequently decide the length of the cancellable Iris template of the particular Iris image. As the size of the Iris code is big, i.e., 10,240 bits, we manually permute the Iris code by swapping two bits anywhere from the Iris code randomly, for, e.g., one permutation is done by swapping the first and 200th bit of the Iris code and the other is done by swapping the first and 300th bit of the Iris code. After the calculation of these permutations, we perform a bitwise product on these two permutations. This operation is the first addition that is introduced in the min-hashing algorithm, is called the P-Hadamard product where P is the number of permutations of the Iris code that we multiply to form the product vector. This result obtained after the bitwise product of the permutations is in the form of a matrix with the dimensions of 50 × 512. Hence we need to convert this Matrix into a vector of size 10,240 for which we have used the operator for straightening out the matrix. We then select the first k bits of this product vector and throw away the remaining bits in the result. This introduces an additional distortion in the original Iris code which helps us in making the cancellable Iris template more secure. This operation is done we proceed to the next additional operation in the min hashing algorithm which is called the modulo threshold function for which the operator required called as the modulo threshold 'τ' was declared earlier. We note down the first occurrence of the bit '1' in this reduced product of the permutations and then modulo it with the difference of k and the modulo threshold 'τ'. This will give us one by one each value in the template of the Iris image depending upon the length of the template that we have chosen earlier. We have to store these values obtained one by one in a vector. In this vector, we have the output of the IFO hashing algorithm in decimal form. To obtain the cancellable iris template we have to perform the look-up table mapping. Once the look-up table is formed we map the template to its entries and finally obtain the cancellable iris template for an iris image. The length of this cancellable iris template is 'd' times the size of the template generated from the IFO algorithm. Hence, by implementing the modified index first one hashing algorithm we have obtained the cancellable Iris template for the Iris image.

4.9 Matching

This process is done in the verification phase of the biometric recognition system. In this process, as we get the query iris code we sort it partially then we need to do the pre-alignment of the query Iris code. In this pre-alignment process, we shift the Iris code 1 bit to the left 16 times and 1 bit to the right 16 times which gives us a total of 33 queries Iris codes including the original Iris code. The main purpose of shifting the iris code and generating new templates is to achieve rotational invariance. As there are 512 columns in the entire circular iris pattern. As a result, shifting one column is equal to 360 / 512 = 0.703125 degrees, if we shift 16 times means it will result in an 11.25-degree rotation.

After the pre-alignment process, we apply the index first one hashing algorithm on each of these 33 queries Iris codes and map the output from this algorithm to the look-up table which will give us corresponding 33 cancellable Iris templates. We will then match these cancellable templates against the template that is stored in the database. Amongst all these comparisons the comparison which will give us the highest accuracy will be considered as valid and the corresponding query Iris code will be accepted.

5 Experimental results and analysis

In this section, we are going to discuss the experimental setup: which consists of information about the iris template database used and the various software's and tools used, analysis of the results obtained from the proposed system, and their comparison with the existing system's results.

5.1 Experimental setup

The iris images used in this study are from the CASIA-v3 database that contains three classes of images namely: iris interval, iris lamp, iris twins. Among all these classes of the iris templates, we are going to use the first one, i.e., the CASIA iris interval for the evaluation of accuracy performance. This class has 2,639 iris templates obtained from 396 distinct people. We are only considering the images of the left eye. To ensure that the matching is standard we take only those subsets of the class who have at least seven samples. There are 124 such subsets, hence we get a total of $124 \times 7 = 868$ iris templates. If we need to perform comparisons within the class then we match the iris template with other iris images of the same person. By doing so on our selected dataset we got 2,604 comparisons that are genuine scores. And for comparisons with other class iris templates, we compare the iris template with all the other images of iris from different classes. After doing so we got a sum of 373,674 comparisons that are imposter scores. Here for the evaluation of performance, we are using the EER, i.e., the equal error rate. EER is a point where the false acceptance rate (FAR) and false rejection rate (FRR) are equal. For the pre-processing of the iris images, we have used the USIT toolkit which contains different tools for various operations performed on the iris image. For segmentation and normalisation, we have used the weighted adaptive Hough and ellipsopolar transform tool. After executing this tool we get iris texture as an output. For converting the iris texture into iris code we used scale invariant feature transform tool. We get the iris code as the output of this stage which is in the form of an image.

5.2 Results and analysis

The metrics used to analyse the proposed fusion method performance are:

- Genuine acceptance rate (GAR): It is the measure by which the system accepts genuine iris templates in the total number of iris templates tested.
- False rejection rate (FRR): It is the measure by which a genuine iris template on the total number of iris templates tested is falsely refused. FRR can also be represented using GAR, i.e., GAR = 1 – FRR. GAR stands for GAR.

- False acceptance rate (FAR): It is the measure by which a false iris template on the total number of iris templates tested is wrongly accepted.
- Equal error rate (EER): It is the error value obtained when the values of FRR and FAR are equal. Using genuine score distribution as well as imposter score distribution, the performance measures are also calculated.

5.2.1 Accuracy performance

The performance of the proposed method is measured in one of the evaluation metrics stated above which is the EER. The EER in turn is dependent upon the various parameters used in the proposed method. The fluctuation in the EER values is shown in Table 2 based on the changes in different parameters. By varying the value of m and keeping constant P as 3 and τ as 0. The window size k is also increased gradually. As a result of these experiments, we can see from the table that with m at 100 and the window size as 10 we get the first instance of the minimum EER as 0.52%. Beyond m = 100 the EER remains nearly the same with constant k whereas for k > 10 the performance accuracy degrades or the EER rises.

Table 2 Resultant EER with varying m

Equal error rate (EER)								
Window size (k)	*Number of hash function (m)*							
	10	*20*	*40*	*60*	*80*	*100*	*200*	*400*
10	6.5	3.0	1.6	1.12	0.87	0.52	0.52	0.53
100	3.12	1.32	1.05	0.75	0.54	0.52	0.53	0.54
200	3.22	1.35	0.89	0.82	0.65	0.52	0.57	0.53
300	2.85	1.10	0.80	0.75	0.62	0.52	0.52	0.52

We performed experiments with constant m as 50 and as 0 and varying k and P. The results of these are listed in Table 3. We can interpret from the following that as the value of k increases the EER decreases as it becomes easy to locate the bit '1' in the window of larger size. Whereas the EER is significantly larger at higher values of P. We can conclude that the higher the k will give better EER if the value of P is also high as the higher k will compensate for the effect of the Hadamard product. Further, the EER remains nearly constant for higher values of k. The tabular representation of these variations in Table 4.

Table 3 Resultant EER with varying k and P

Equal error rate (EER)								
Hadamard product (P)	*Window size (k)*							
	5	*10*	*15*	*20*	*30*	*40*	*50*	*60*
2	0.88	0.58	0.53	0.52	0.52	0.53	0.52	0.52
3	1.98	0.89	0.78	0.76	0.68	0.69	0.60	0.65
4	5.20	1.98	1.06	0.98	0.89	1.06	0.82	0.72
5	21.10	5.12	2.65	1.78	1.65	1.35	1.96	1.08

Table 4 Comparison of the proposed method with the state of art techniques

Methods	*Number of iris images used*	*EER*
Partial sort and look-up table	868 (left eye)	0.52
IFO hashing (Lai et al., 2017)	868 (left eye)	0.54
Block remapping (Jenisch and Uhl, 2011)	2,653	1.30
Bio-encoding (Zuo et al., 2008)	740	6.27
Adaptive bloom filter (Chong et al., 2006)	1,332 (left eye)	1.14
Bin-combo (Pillai et al., 2010)	1,332 (left eye)	4.41

We performed the partial sort on the iris codes with a vast range in the values of p, q, and d. In doing so we got the best EER as 0.52 with the values of these parameters as $p = 240$, $q = 4$ and $d = 2$.

5.2.2 Irreversible

As we are using the indexed first one hashing technique which is combined with the partial sort which creates a cancellable template. Even if the template is compromised it is not possible to get the data of the user as the template we are creating is irreversible.

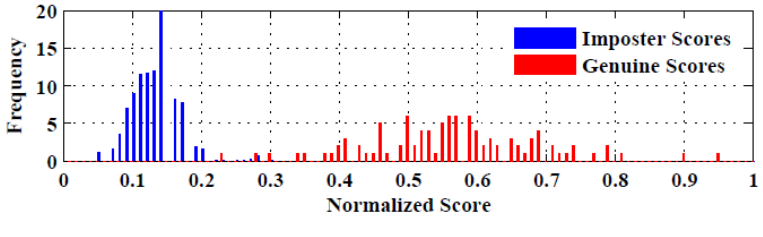
5.2.3 Diversity

The d-prime value computes the change in mean in terms of standard deviation units between genuine and imposter distributions. It is calculated as shown in equation (3).

$$d' = \frac{|\mu_1 - \mu_0|}{\sqrt{(\sigma_1^2 + \sigma_0^2)/2}} \tag{3}$$

We have plotted the genuine and imposter score distribution of the proposed method, as shown in Figure 8, and calculated the d-prime value using equation (3). The resultant d-prime value for our method is 3.812 that show that the genuine and imposter scores are clearly separated and proves the validity of the proposed method.

Figure 8 Genuine vs. imposter scores distribution (see online version for colours)



5.2.4 Revocability

The template's revocability condition states that if a transformed template is compromised, the proposed method should be capable of generating a new transformed

sample with no cross-matching between them. In our proposed method we can generate *n*-different templates for the existing template by performing shift operations.

5.3 Security analysis

We can determine the proposed method is secure if it preserves certain security properties using the various steps introduced in the method. The system is said to produce cancellable iris templates that are non-invertible if they are resilient to various invertibility attacks. These attacks are carried out by the attacker who is equipped with all the algorithm parameters and cancellable iris templates which are stolen by him. Record Multiplicity attack is an attack on privacy which is done using multiple cancellable iris templates that are stolen with or without the parameters available (Scheirer and Boult, 2007). But to achieve success in this attack the attacker needs to map all the cancellable iris templates generated to their original iris codes which are computationally impossible thereby averting the attack. Pre-image attack focuses on just retrieving an iris template that is just close to the original template and not the original iris template completely which considerably reduces the attack complexity (Nandakumar and Jain, 2015). This attack is averted by the look-up table mapping that is done to the output of the IFO algorithm as the look-up table is highly randomised and can be further done by increasing the range set in the implementation. As we know that using the proposed algorithm we can generate a large number of IFO hashed codes from a single iris code as we permute the iris code and the number of permutations for such long iris codes can be numerically huge (Bringer et al., 2015). Hence, it is impossible to obtain the original iris code from these hashed codes which make the cancellable template irrevocable. Unlinkability implies that it should be difficult for the adversary to make a difference that a single source (same users biometric) is used for the generation of one or more protected templates (Daugman, 2005). This is ensured as the permutations of the same iris code are independent of each other which thereby prove the independence of the cancellable templates generated from the same iris code. Hence, the unlikability is preserved.

5.4 Comparison with existing techniques

We compared our proposed method with five state-of-the-art cancellable iris template generation schemes. The comparison will be based on the EER given by each of the techniques. In all the techniques that are listed below, the number of iris images used in each technique is different. The number of iris images used is also a factor on which the EER of the technique depends. Hence, we will include the number of iris images used in each of these techniques. These comparisons are summarised in a tabular form in Table 4.

Hence, the proposed cancellable iris template generation method proves more efficient than the other methods listed in the table with an EER of 0.52%. The modification added to the IFO hashing scheme (Lai et al., 2017) by including partial sort and randomised look-up table improved its performance further.

6 Conclusions

In this study, we proposed a new modification to the existing IFO hashing algorithm by adding the partial sort technique and store the cancellable iris template obtained from the IFO algorithm as the mapped entries to the look-up table which ensures more security. Due to these modifications to the IFO algorithm, we could improve the EER of the existing system from 0.54% to 0.52%. The security properties like irreversibility, revocability, and unlinkability are also preserved and the cancellable iris templates generated are also immune to various privacy attacks. In future work, we want to change the *d* value so that the length of the cancellable template can be decreased without affecting the accuracy and efficiency of the system.

References

Adamović, S., Miškovic, V., Maček, N., Milosavljević, M., Šarac, M., Saračević, M. and Gnjatović, M. (2020) 'An efficient novel approach for iris recognition based on stylometric features and machine learning techniques', *Future Generation Computer Systems*, Vol. 107, No. C, pp.144–157.

Bringer, J., Morel, C. and Rathgeb, C. (2015) 'Security analysis of bloom filter-based iris biometric template protection', in *Proceedings of the 2015 International Conference on Biometrics (ICB)*.

Broder, A. (1997) 'On the resemblance and containment of documents', in *Proceedings of the Compression and Complexity of Sequences*.

Broder, A., Charikar, M. and Frieze, A. (2000) 'Min-wise independent permutations', *J. Comput. Syst. Sci.*, Vol. 60, No. 3, pp.630–659.

CASIA Iris Database, version 3 [online] http://www.cbsr.ia.ac.cn/english/IrisDatabase.asp (accessed 20 February 2021).

Chong, S.C., Jin, A.T.B. and Ling, D.N.C. (2006) 'High security iris verification system based on random secret integration', *Comput. Vis. Image Underst.*, Vol. 102, No. 2, pp.169–177.

Daugman, J. (2005) 'How iris recognition works', *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 14, No. 1, pp.21–30.

Dwivedi, R., Dey, S., Singh, R. and Prasad, A. (2017) 'A privacy-preserving cancelable iris template generation scheme using decimal encoding and look-up table mapping', *Computers & Security*, Vol. 65, pp.373–386, ISSN 0167-4048.

Gupta, R. and Sehgal, P. (2016) 'A review on iris recognition system for person identification', *International Journal of Biometrics*, Vol. 8, No. 2, pp.145–178.

Gupta, R. and Sehgal, P. (2019) 'A complete end-to-end system for iris recognition to mitigate replay and template attack', in Wang, J., Reddy, G., Prasad, V. and Reddy, V. (Eds.): *Soft Computing and Signal Processing, Advances in Intelligent Systems and Computing*, Vol. 900, Springer, Singapore.

Jenisch, S. and Uhl, A. (2011) 'Security analysis of a cancelable iris recognition system based on block remapping', in *Proceedings of the IEEE International Conference on Image Processing.*

Jeong, J.Y. and Jeong, I.R. (2019) *Efficient Cancelable Iris Template Generation for Wearable Sensors*, Wiley, Republic of Korea.

Johar, T. and Kaushik, P. (2015) 'Iris segmentation and normalization using Daugman's rubber sheet model', *International Journal of Scientific and Technical Advancements*, Vol. 1, No. 1, pp.11–14.

Lai, Y-L., Jin, Z., Teoh, A.B.J., Goi, B-M., Yap, W-S., Chai, T-Y. and Rathgeb, C. (2017) 'Cancellable iris template generation based on indexing-first-one hashing', *Pattern Recognition*, Vol. 64, No. C, pp.105–117.

Nandakumar, K. and Jain, A.K. (2015) 'Biometric template protection: bridging the performance gap between theory and practice', *IEEE Signal Process. Mag.*, Vol. 32, No. 5, pp.88–100.

Pillai, J.K., Patel, V.M., Chellappa, R. and Ratha, N.K. (2010) 'Sectored random projections for cancelable iris biometrics', *Proceeding of the IEEE International Conference on Acoustics Speech and Signal Processing*.

Priya, S.S.S., Karthigaikumar, P., Mangai, N.M.S. and Sandhya, R. (2015) 'A survey of attacks on iris biometric systems', *International Journal of Information and Communication Technology*, Vol. 7, Nos. 4–5, pp.437–454.

Rafiq, S. and Selwal, A. (2019) 'Template security in iris recognition systems: research challenges and opportunities', in Singh, P., Kar, A., Singh, Y., Kolekar, M. and Tanwar, S. (Eds.): *Proceedings of ICRIC 2019. Lecture Notes in Electrical Engineering*, Vol. 597, Springer, Cham.

Ratha, N.K., Chikkerur, S., Connell, J.H. and Bolle, R.M. (2007) 'Generating cancelable fingerprint templates', *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 29, No. 4, pp.561–572.

Rathgeb, C., Breitinger, F., Busch, C. and Baier, H. (2014) 'On application of bloom filters to iris biometrics', *IET Biometrics*, Vol. 3, No. 4, pp.207–218.

Rathgeb, C., Uhl, A., Wild, P. and Hofbauer, H. (2016) 'Design decisions for an iris recognition SDK', in Bowyer, K. and Burge, M.J. (Eds.): *Handbook of Iris Recognition*, 2nd ed., *Advances in Computer Vision and Pattern Recognition*, Springer, London.

Sandhya, M. and Prasad, M.V.N.K. (2017) 'Biometric template protection: a systematic literature review of approaches and modalities', in Jiang, R., Al-maadeed, S., Bouridane, A., Crookes, P. and Beghdadi, A. (Eds.): *Biometric Security and Privacy. Signal Processing for Security Technologies*, Springer, Cham.

Sandhya, M. and Prasad, M.V.N.K. (2018) 'Multi-algorithmic cancelable fingerprint template generation based on weighted sum rule and T-operators', *Pattern Analysis and Applications*, Vol. 21, No. 2, pp.397–412.

Scheirer, W.J. and Boult, T.E. (2007) 'Cracking fuzzy vaults and biometric encryption', in *Proceedings of the Biometrics Symposium*.

Uhl, A. and Wild, P. (2012) 'Weighted adaptive Hough and ellipsopolar transforms for realtime iris segmentation', *5th International Conference on Biometrics (ICB'12)*, New Delhi, India, 29 March–1 April, pp.283–290.

Zuo, J., Ratha, N.K. and Connel, J.H. (2008) 'Cancelable iris biometric', in *Proceedings of the 19th International Conference on Pattern Recognition (ICPR 2008)*.