INDEXING-FIRST-ONE HASHING BASED CANCELABLE IRIS TEMPLATE GENERATION

LAI YEN LUNG

MASTER OF ENGINEERING SCIENCE

LEE KONG CHIAN FACULTY OF
ENGINEERING AND SCIENCE
UNIVERSITY TUNKU ABDUL RAHMAN

SEPTEMBER 2017

INDEXING-FIRST-ONE HASHING BASED CANCELABLE IRIS TEMPLATE GENERATION

By

LAI YEN LUNG

A Master dissertation submitted to the Department of
Department of Mechatronics & Biomedical Engineering,
Lee Kong Chian Faculty of Engineering & Science,
University Tunku Abdul Rahman,
in partial fulfilment of the requirements for the degree of
Master of Engineering Science
September 2017

DECLARATION

I hereby declare that the dissertation is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTAR or other institutions.

Name _____

Date _____

APPROVAL SHEET

This dissertation entitled "**INDEXING-FIRST-ONE HASHING BASED CANCELABLE IRIS TEMPLATE GENERATION**" was prepared by LAI YEN LUNG and submitted as partial fulfilment of the requirements for the degree of Master of Engineering Science (M.Eng.Sc) at Universiti Tunku Abdul Rahman.

Approved by:

_____

(Prof. Dr. Goi bok Min) Date:……..

Supervisor

Department of Mechatronics & Biomedical Engineering

LKC Faculty of Engineering & Science

Universiti Tunku Abdul Rahman

_____

(Mr. Chai Tong Yuen) Date:……..

Co-supervisor

Department of Mechatronics & Biomedical Engineering

LKC Faculty of Engineering & Science

Universiti Tunku Abdul Rahman

Specially dedicated to

My beloved family and girl friend

ACKNOWLEDGEMENTS

In the name of Jesus for giving these acknowledgements, I would like to thank god for everything He has gave to me e.g. Knowledge, Wisdom, Courage etc. Besides, Thanks for everyone who had contributed to the successful completion of this project. I would like to express my gratitude to my research supervisor, Prof. Goi Bok Min and Co- supervisor Mr. Chai Tong Yuen for their invaluable advice, guidance and his enormous patience throughout the development of the research. Additionally, thanks to Dr. Jin Zhe and Prof. Andrew and Dr. Yap Wun She for their supports and advises on this research.

In addition, I would also like to express my gratitude to my loving parent and friends who had helped and given me encouragement.

Finally, I would like to acknowledge Malaysia eScience fund (01-02-11-SF0201) for financially sponsoring part of this research.

ABSTRACT

INDEXING-FIRST-ONE HASHING BASED CANCELABLE IRIS TEMPLATE GENERATION

Lai Yen Lung

Iris has been widely recognized as one of the strongest biometrics attributed to its high system performance. However, templates in conventional iris recognition systems are unprotected and highly vulnerable to numerous security and privacy attacks. Despite a number of iris template protection schemes have been proposed but at the expense of substantially decreased system performance. In this dissertation, we introduce a new iris template protection scheme, coined as "Indexing-First-One" (IFO) hashing. IFO hashing is inspired from Min-hashing, which is primarily used in text retrieval domain, but the scheme has been further strengthened by two novel mechanisms, namely P -order Hadamard multiplication and modulo threshold function. The IFO hashing scheme strikes the balance between system performance and privacy/security protection. Comprehensive experiments on CASIA-v3 iris benchmarking database and rigorous analysis demonstrated decent system performance i.e. 0.56% error rate able to achieve yet offer strong resilience against several major security and privacy attacks such as attack via record multiplicity, pre-image attack etc.

TABLE OF CONTENTS

DECLARATION ii

APPROVAL SHEET iii

ACKNOWLEDGEMENTS vi

ABSTRACT vii

TABLE OF CONTENTS viii

LIST OF TABLES xi

LIST OF FIGURES xii

LIST OF APPENDICES xiv

CHAPTER

1.0 INTRODUCTION 1

1.1 Backgrounds 1

1.2 Biometrics 4

1.3 Human Iris 5

1.4 Iris Recognition System 7

1.5 Iris Image Database 9

1.6 Metrics for Performance Evaluation 11

1.7 Security and Privacy Issues in Biometric Systems 13

1.8 Biometric Template Protection (BTP) 15

1.8.1 Feature Transformation 17

1.8.2 Salting 17

1.8.3 Non-Invertible Transform 19

1.8.4 Biometric Cryptosystem 20

1.8.5 Key Binding 20

1.8.6 Key Generation 21

1.9 Problem Statements 23

1.10 Objectives 25

1.11 Contributions 25
1.12 Dissertation Organization 27

2.0 LITERATURE REVIEW 28
2.1 Biometric Salting Approaches 29
2.2 Non-Invertible Transformation Approaches 31
2.3 Summary 38

3.0 PROPOSAL FOR INDEXING-FIRST-ONE HASHING 39
3.1 Preliminaries 39
3.1.1 Local Sensitive Hashing 39
3.1.2 Min-Hashing 40
3.1.3 IrisCode Generation 41
3.2 Indexing First-One Hashing 43
3.3 Matching 46
3.3.1 Pre-alignment 46
3.3.2 Relation to Jaccard Similarity 47
3.3.3 Definition and Theorem for IFO Matching 52
3.3.4 Matching in Practice 55
3.4 Alignment-Free IFO 56

4.0 EXPERIMENTS AND ANALYSIS 60
4.1 Test for IrisCode's Performance 60
4.2 Test for IFO & Alignment-Free IFO Performance 61
4.3 Test for IFO Template Performance 64
4.3.1 Effect of Parameter $\boldsymbol{m}$ 64
4.3.2 Effect of Parameters $\boldsymbol{K}$ and $\boldsymbol{P}$ 65
4.3.3 Effect of Parameter $\boldsymbol{\tau}$ 66
4.4 Comparison with Other Schemes 67
4.5 Non-Invertibility Analysis 68
4.5.1 Single Hash Attack (SHA) 69
4.5.2 Multi-Hash Attack (MHA) 78

4.5.3 Attack via Record Multiplicity (ARM) 79
4.6 Potential Security Attack 82
4.6.1 Pre-image Attack (PIA) 82
4.6.2 False Accept Attack (FAA) 84
4.7 Revocability Analysis 86
4.8 Unlinkability Analysis 88

5.0 CONCLUSION AND FUTURE WORKS 92
5.1 Conclusion 92
5.2 Future Works 93

REFERENCES 94

APPENDICES 101

LIST OF TABLES

TABLE	TITLE	PAGE
Table 1:	Summary of CASIA-Iris-Interval (CASIA-IrisV3)	11
Table 2:	Summarized existing BTP for iris recognition system	37
Table 3:	Matching result for applied pre-alignment IFO hashed codes ($m = 400, K = 400, \tau = 0, P = 1$) and alignment-free IFO hashed codes ($n = 32,\ l = 10, m = 400, K = 400, \tau = 0, P = 1$)	63
Table 4:	Average computation cost (sec) for applied pre-alignment Matching in IFO hashed codes ($m = 400, K = 400, \tau = 0, P = 1$), and alignment-free IFO hashed codes (Bloom filter generation + Matching) ($n = 32,\ l = 10,\ m = 400, K = 400, \tau = 0, P = 1$)	63
Table 5:	Performance result using $P = 3, \tau = 0$	65
Table 6:	Results for *EER* subject to different values of K and P with $m = 50$	66
Table 7:	Summarized results in EER of IFO hashing with the state of the arts (CASIA v3-Interval database).	68

LIST OF FIGURES

FIGURE	TITLE	PAGE
Figure 1.1:	Human Iris	7
Figure 1.2:	Example of iris images in CASIA-Iris-Interval (CASIA iris image database, Available: http://www.cbsr.ia.ac.cn/Database.htm.)	10
Figure 1.3:	Example of iris images in CASIA-Iris-Lamp (CASIA iris image database, Available: http://www.cbsr.ia.ac.cn/Database.htm.)	10
Figure 1.4:	Example of iris images in CASIA-Iris-Twins (CASIA iris image database, Available: http://www.cbsr.ia.ac.cn/Database.htm.)	11
Figure 1.5:	Genuine-imposter distribution	13
Figure 1.6:	Categorized four major attack points in biometric system	14
Figure 1.7:	Biometric template protection scheme	16
Figure 2.1:	Sectored Random Projection	31
Figure 2.2:	Bio-Encoding technique	34
Figure 2.3:	Example of Bloom filter technique	36
Figure 2.4:	Look-up mapping process	36
Figure 3.1:	The Min-hashing algorithm with two hash functions	42
Figure 3.2:	Toy example of IFO hashing based on three hash functions	45
Figure 3.3:	Example of pre-alignment process	47
Figure 3.4:	Possible permutation outputs under produce codes with same permutation	48
Figure 3.5:	$\mathbb{P}(z \geq z')$ vs $1 - D(\boldsymbol{X}, \boldsymbol{Y})$	54

Figure 3.6: Alignment-free IFO hashing 59

Figure 4.1: Original accuracy performance of IrisCode 61

Figure 4.2: Equal Error Rate versus Security Threshold 67

Figure 4.3: IrisCode recovery process using permutation token (best view in color) 72

Figure 4.4: Graph of the mean remaining bits with value '1' in the *K*-window (n) versus *P* 77

Figure 4.5: Estimated SHA complexity for IrisCode restoration. 77

Figure 4.6: The genuine, imposter and pseudo-imposter distributions; large overlap between imposter and genuine due to no pre-alignment. 88

Figure 4.7: Hamming score distribution of randomly permuted IrisCodes 91

Figure 4.8: Pseudo-Imposter & Pseudo-Genuine distribution: CASIA Database v3 91

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
APPENDIX A:	Example of IrisCode and IFO hashed Code	101

CHAPTER 1

INTRODUCTION

1.1 Backgrounds

Conventional deterministic identity and recognition mechanisms rely on memory-based credentials or possession tokens such as passwords, PIN numbers or access cards for system access. This model works efficiently only under the presumption that a legitimate user can always holds the consistency through constantly presenting unique cryptographic key, password, or PIN number. In practice, this model undermines the possibility and uncertainty that a user may fail to provide the exact password, or PIN number for every authentication. For instance, given a system where the input knowledge is exact, the transmission between the information may just be approximate, subjected to noise and perturbation. Moreover, long and complex cryptographic key is difficult to remember, therefore, users typically make typing error while entering the key/password using keyboard. Consequently, a deterministic model recognition system may not work when fuzziness is involved.

On the other hand, biometrics exploits the statistical analysis of individual's physiological and behavioural characteristics. The uses of biometric recognition system offer an alternative to perform personal recognition under non-deterministic condition. For example, under human biometric system, despite the input biometric data prone to slight distortion due

to noises factor during acquisition process, recognition or identification can still be able to success, provided the enrolled and query biometric data are similar up to certain extend.

Typical biometric traits which can be used to construct a biometric system are human fingerprint, voice, iris face etc. The uniqueness of these biometric traits facilitated the rapid proliferation of biometric recognition system to be integrated into personal authentication mechanism especially for information forensics and criminal investigation. The foreseeable biometric applications trend in the future indirectly encouraged the large-scales deployment of a biometric database for biometric information storage.

However, the security of the database used to store individual biometric data is questionable. This is because it might be exposed to adversary's attack or compromise. Once the database compromised, severe influences of personal biological information will lead to permanent identity loss because the biological traits used are irrevocable and irreplaceable. Therefore, an effective and efficient solution for this particular damaging event is urgently needed for any biometric recognition systems.

Besides, due to the large variation of personal biometric data collected in every acquisition, the conventional mechanism in cryptology that relies on the encryption/decryption in securing the input data is no longer reliable. This is because, for every encryption/decryption used in cryptology protocol, the input data to be encrypted/decrypted has to be always the same (e.g

password/ID). This promoted the growth of a new research area namely biometric template protection, which provides security protection for the biometric data stored in a database as the alternative.

In order to address the security issues in a biometric database, a variety of template protection techniques have been proposed to protect the biometric data stored in a database. For instances, cancelable transformation e.g. non-invertible transformation, salting approaches and biometric cryptosystem e.g. key binding, key generation. However, most of the existing template protection schemes still suffer from its own vulnerabilities in term of security and system performance trade-off. Particularly, in order to achieve a higher system recognition performance, one requires high amount of distinctive information to be collected from a human iris. However, in term of security e.g. non-invertibility, information loss is required. For cancelable transformation approach, most of the current constructions are still suffer from serious security weakness with respect to their non-invertibility and security attacks such as attack via record multiplicity, pre-image attack, etc. More details about the security weakness of current construction will be covered in Chapter 2.

It is commonly understand that an efficient and reliable personal authentication system is the basic requirement for every security applications. For example, computer login system, immigration control application, unnamed surveillance, e-commerce transaction applications, etc. Traditionally, all of these security applications are categorized under the deterministic model that rely on the knowledge of a secret (e.g Password: 12345) to justify the

validity of what an individual possessing and then whether reject or accept the user who is trying to access the system. However, such authentication mechanism comes along with certain limitation, for instance, the use of secrets or passwords can be easily stolen or lost; with a simpler password (e.g 123), it is easy to collide, guessed or used by other users. Therefore, an intruder can easily guess the correct password and get unauthorized access to the system. Despite other complex secrets or passwords can be used, however, the complex password/secret can be hard to remember and easily forgotten by a user. The issue with "too many passwords" becomes worse and inconvenient when a certain application requires regular renewal of password.

1.2 Biometrics

Since, biometrics offers an alternative way for personal authentication, it able to compensate the issues mentioned in the previous section. Firstly, the uniqueness of biometric traits (e.g. iris, fingerprint) can be served as a personal identifier as a secret or password in a biometric recognition system. Secondly, biometric traits are always available and with the user. This implies that the user no longer needs to carry or memorize any password or secret, hence provide a larger degree of convenience for any authentication process. Thirdly, the biometric identifier cannot be shared and stolen compared to traditional secret or password. Besides, the biometric identifiers are inherited to every individual that is more difficult to be manipulated by others. This constituted a strong permanent link between the individual identities and biometric identifiers thus, offers another layer of security protection and convenience for

the user of biometric recognition system compare to the conventional deterministic recognition system

In fact, in the later nineteenth century, human beings have started to use biometrics as a mean for identification. For instant, Alphonse Bertillon (a French police officer) developed a set of tools to identify frequent offenders. These tools included the measurement of the head length/breadth, length of middle finger etc., collectively named as Bertillon. Later, the discoveries of fingerprint pattern, which is highly useful for individual identification by Faulds (1880), Herschel (1880), and Galton (1889) encouraged the replacement of Bertillon system. Until 1963, the first automatic fingerprint matching system was proposed by Mitchell Trauring (Trauring, 1963).

Follows the works on the automated fingerprint by Mitchell Trauring, different automated recognition system have been proposed by using different biometric traits such as, voice, face, signature, hand geometry and iris which are highlighted by Pruzansky (1963), Bledsoe (1966), Mauceri (1965), Ernst (1971), and Daugman (1933) respectively. All of their works have shown the reliability and capability of biometrics in personal identification.

1.3 Human Iris

Among all the biometric traits available today, eye iris is considered as one of the highly reliable biological traits attributed to its discriminability and stability. Human iris is a circular structure made up of two layers. The first layer is the pigmented Fibrovascular so-called stroma. Beneath the stroma is

the pigmented Epithelial cell. The stroma is connected to the sphincter and dilator muscles, which helps in changing the diameter of a pupil and controlling the amount of lights entering the eye. The high density of the stroma pigmentation restricts the lights that passes though the pupil and further gives the colour of the iris.

The iris itself can be further divided into two different zones, which refers as the inner pupillary zone and the outer ciliary zone (Wolff, 1967). These two zones are different in colours and separated by the collarette which is a zigzag pattern formed in between the outer pupil boundary and the inner iris regions.

The iris structure started to form in the third month of gestation (Daugman, 2004). The formation of the unique pattern on the surface of the iris almost completed during the first year of life. After this, the pigmentation of the stroma will begin in the first few years.

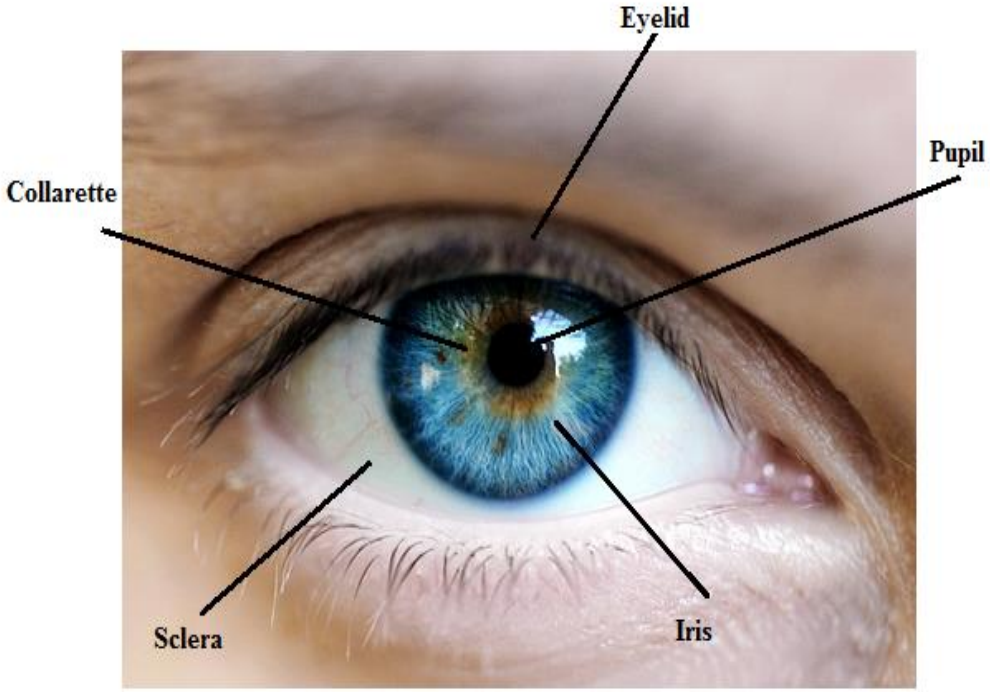

Figure 1.1: Human Iris

The human iris is stable yet protected under cornea. Besides, human iris is epigenetic which means the formation of the unique pattern of the iris is nothing related to genetic factors (Wildes, 1997). With this epigenetic nature, two eyes of an individual will possess very different iris pattern. For identical twins, the iris pattern are also completely different and uncorrelated (Dougman 2004, 2006).

1.4 Iris Recognition System

Among all the biometric identifiers, the human iris is a very reliable feature for personal identification and verification. There exist different proposed iris recognition systems, for example, the iris recognition system by Lim et al. (2001) and Boles et al. (1998). Until 2004 and 2006, the first

automated Iris recognition system using IrisCode was proposed by Daugman (2004, 2006). The IrisCode is a fix dimensionless binary array generated from a human iris. Matching between different Iris codes is conducted by calculating their bits difference (Hamming distance). Besides, it has been validated that the entropy of iris patterns is typically much higher than other biometric traits (Dougman 2004, 2006). This infers that the false matches between different IrisCodes are highly unlikely to be happened. Therefore, apart from verification, iris can be very useful for identification task.

As compared to the other well-known and commonly used biometric system (e.g Face and fingerprint), NIST report, IREX-III showed that iris recognition system contained 100000 times lower false match rate (FMR) than the best face recognition algorithm. On the other hand, even the most commonly used fingerprint recognition system has also shown its failure during matching. For example, a Strathclyde police became the primary suspect in Kilmarnock case due to a false match in thumbprint in the year 1999; an Oregon lawyer was held in for two weeks as a suspect in train bombing due to the false matching in fingerprint data found at the crime scene 100% matching.

Iris recognition system showed a promising result in reducing FMR based on the *statically independence theory*. In UAE database, 200 billion of IrisCodes have cross-matched and the result showed that it is impossible for two different IrisCodes to be matched with a Hamming distance lower than $\frac{1}{3}$ and the mean of the score distribution is a binomial distribution along with a mean $= 0.5$. With a Hamming distance of 0.285 between different IrisCodes,

the false match rate probability is 2.0×10^{-8}, when increased into 1000 bits, the false match probability has decreased to 2.0×10^{-11} (Daugman, 2004, 2006). This showed that iris recognition system can be considered as a much more promising approach in doing personal identification.

1.5 Iris Image Database

The most commonly used database for human iris study is the iris database created from Chinese Academy of Sciences (CAS) Institute of Automation (IA), in abbreviation CASIA. CASIA-iris database current has total four versions, which are CASIA-IrisV1, CASIA-IrisV2, CASIA-IrisV3, and CASIA-IrisV4.

In this dissertation, the **CASIA-IrisV3** database is used. CASIA-IrisV3 can further subdivide into three subsets, which are CASIA-Iris-Interval, CASIA-Iris-Lamp, and CASIA-Iris-Twins.

For **CASIA-Iris-Interval**, the iris images were captured by using a close-up near-infrared ray (NIR) camera. The captured iris images are very clear and the iris texture can easily see with naked eye. This subset is well suited for the study of the detailed iris texture and features.

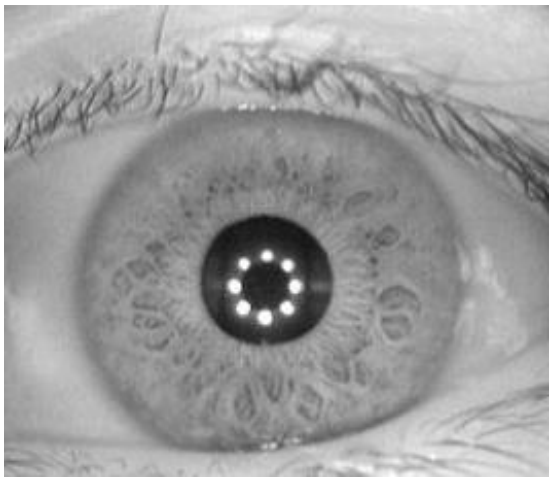

Figure 1.2: Example of iris images in CASIA-Iris-Interval (CASIA iris image database, Available: http://www.cbsr.ia.ac.cn/Database.htm.)

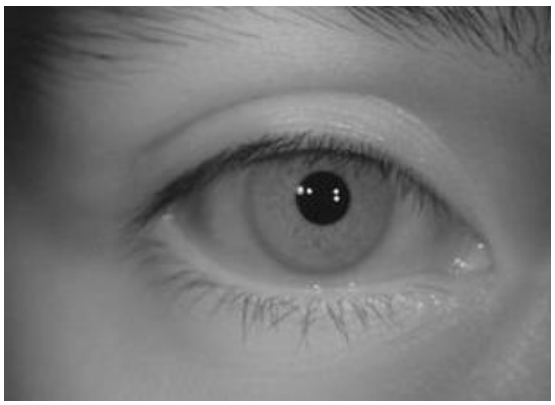

Figure 1.3: Example of iris images in CASIA-Iris-Lamp (CASIA iris image database, Available: http://www.cbsr.ia.ac.cn/Database.htm.)

For **CASIA-Iris-Twins,** 100 pairs of twin's iris images were capture. This subset is suitable for the study of dissimilarity and similarity between irises in twins.

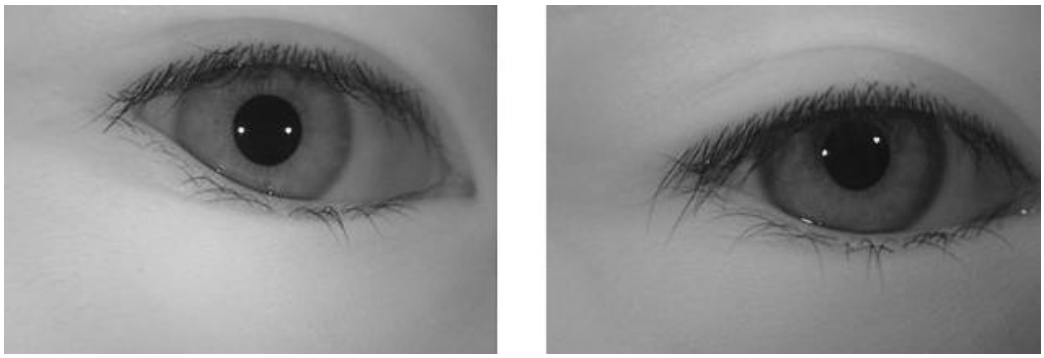

Figure 1.4: Example of iris images in CASIA-Iris-Twins (CASIA iris image database, Available: http://www.cbsr.ia.ac.cn/Database.htm.)

	Characteristics						
Subset	*Sensor*	*Environment*	*Session*	*No. of subjects*	*No. of classes*	*No. of images*	*Resolution*
CASIA-Iris-Interval	CASIA close-up iris camera	Indoor	Two different session	249	395	2639	320*280
CASIA-Iris-Lamp	OKI IRISPASS-h	Indoor with lamp on/off	One session	411	819	16212	640*480
CASIA-Iris-Twins	OKI IRISPASS-h	Outdoor	One session	200	400	3183	640*480

Table 1: Summary of CASIA-Iris-Interval (CASIA-IrisV3)

In this dissertation, since the study of template protection for iris features is our focus, hence CASIA-Iris-Interval has been chosen.

1.6 Metrics for Performance Evaluation

In a generic biometric system, the performance evaluation devoted the recognition performance after matching among different biometric templates. The most commonly agreed performance indicators included the False

Acceptance Rate (*FAR*), False Rejection Rate (*FRR*) and Equal Error Rate (*EER*).

FAR refers to the error rate when the system accepted an unauthorized user, while FRR refers to the error rate when the system rejected the legitimate user. The *FAR* and *FRR* can be calculated by using the following equations:

$$FAR = \frac{\text{number of accepted unauthorized user}}{\text{total number of unauthorized access}} \times 100 \quad (1.1)$$

$$FRR = \frac{\text{number of rejected legitimate user}}{\text{total number of legitimate access}} \times 100 \quad (1.2)$$

For *EER*, it is another indicators commonly used to compare the performance for different biometric systems. *EER* refers to the point at which *FAR* and *FRR* are equal. In general, *EER* can be approximated by $EER = \frac{FAR+FRR}{2}$, and lower *EER* implies lower *FAR* and *FRR*, thus, higher system performance (Jain et al., 2011).

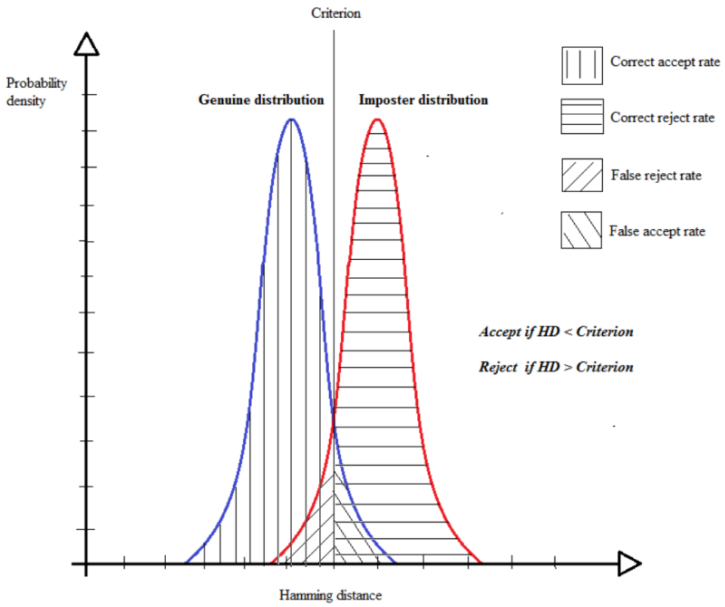

Figure 1.5: Genuine-imposter distribution

Besides, the genuine-imposter distribution is also used to evaluate the performance of a biometric system. For a given criterion, the genuine-imposter distribution graph shows the distribution of the correct accept rate, correct reject rate, *FRR* and *FAR* as depicted in Figure 1.5. Strong overlapping between the genuine and imposter distribution indicates poor performance and vice versa.

1.7 Security and Privacy Issues in Biometric Systems

Ratha et al. (2001) have highlighted eight levels of biometric system attacks which can be potentially launched by an attacker. As shown in Figure 1.6, the eight levels of attacks have been categorized into four main attack points.

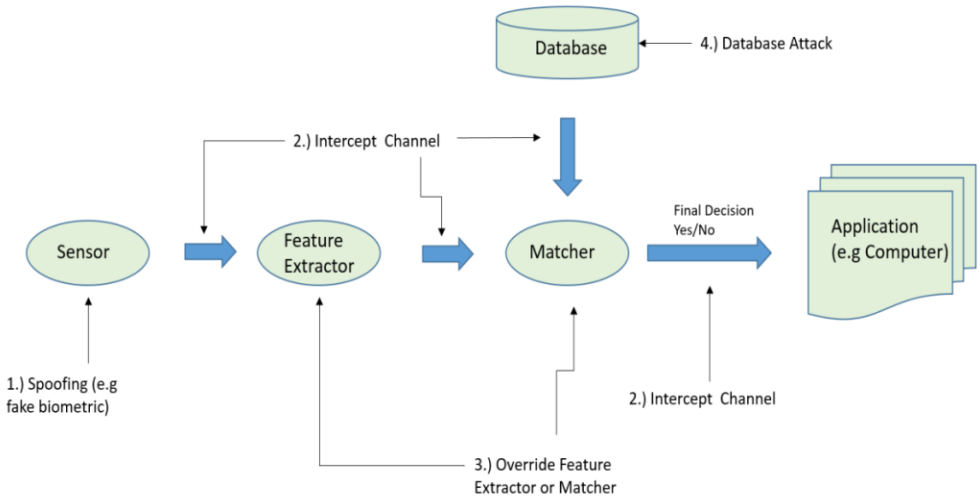

Figure 1.6: Categorized four major attack points in biometric system

1. Spoofing: Fake biometric being use during the acquisition of personal biometric data.
2. Intercept Channel: illegitimate biometric data being injected into the genuine biometric data during data transmission in the channel, final decision also may be overridden before reaching the application.
3. Override Attack: The feature extractor or matcher is potentially being attacked by Trojan horse to perform an illegitimate action (e.g generate pre-defined biometric feature), a decision from the matcher module also potentially being overridden.
4. Database Attack: Original stored biometric data being removed and replaced by illegitimate data.

Jain et al. (2008) labelled the most damaging attack on these four attack points as the database attack. Once the database is being attacked, it will lead to severe security and privacy issues. They also highlighted three major vulnerabilities on the consequences of this damaging attack, which can be described as follow:

1. The genuine user biometric data can be replaced by other illegitimate data to gain unauthorized access.
2. Spoofing/fake biometric traits can be generated from the genuine user biometric data stored in the database.
3. The stolen genuine biometric data might be used for other unintended purposes (e.g fingerprint data stored for crime search is being used in healthcare data search, cross matching).

1.8 Biometric Template Protection

In this dissertation, the database attack mentioned in the previous Section 1.7 is indeed our focus and the solution for this kind of attack is to design a biometric template protection (BTP) scheme to protect the template stored in the database i.e. generate *cancelable* iris template (protected template) to store inside the database as the replacement of the original IrisCode. For an ideal biometric template protection scheme, it must satisfy four main criteria, which have been highlighted by (Jain et al., 2008; Teoh et al., 2006) as follow:

1. **Unlinkability**: It should not be able to differentiate whether one or more cancelable templates are generated from the same source (same

user's biometric). This is to prevent cross-matching in different applications.

2. **Revocability**: It should be computationally infeasible to derive its original counterparts from *multiple* cancelable templates. This enables new templates to be revoked or renewed in order to replace the old one meanwhile preventing the adversary from obtaining the original template.
3. **Non-invertible**: It should be computationally infeasible to derive its original counterparts from the cancelable template and/or the helper data; hence, it prevents the abuse of the compromised biometric data and enhances the security of the system.
4. **Performance**: The system performance of cancelable template must be approximately preserved with respect to its original counterparts.

Generally, BTP can be divided into two main categories, by using the feature transformation and biometric cryptosystem (Jain et al., 2008).

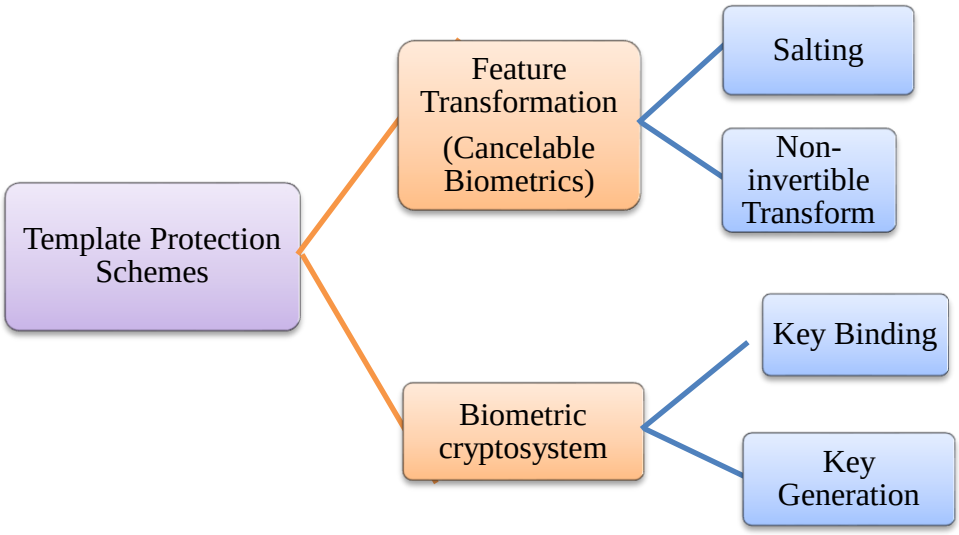

Figure 1.7: Biometric template protection scheme

1.8.1 Feature Transformation

In feature transformation, the original biometric template T will be transformed by using certain transformation function $\mathcal{F}$. The transformed template $\mathcal{F}(T;K)$ is cancelable and will be stored in a database instead of the original biometric template. The transformation parameters can be derived randomly using a key, K or random number (password). Same procedure when comes to matching, the query template Q has to be processed through the same transformation $\mathcal{F}(Q;K)$ and allowed the matching process to be carried out in the transformed domain. During the matching process, only when $\mathcal{F}(T;K)$ and $\mathcal{F}(Q;K)$ are close enough (e.g. low Hamming distance: the number of position at which the corresponding symbols/bits values are different is low), meet the threshold T set, then it will be considered as a successful match. Feature transformation can be further classified into **salting** and **non-invertible transformation** depending on the transformation function $\mathcal{F}$.

1.8.2 Salting

In biometric salting, independent auxiliary data such as user-specific password or token is combined with biometric data to render a distorted version of the biometric template. Salting approach is invertible by using the same transformation key, K. The incorporated user-specific key, K increased the entropy (randomness) of the original biometric. This made an adversary difficult to guess the template, thus, non-invertibility is satisfied. When compromised case happens, one can simply replace the user-specific key for new cancelable template generation, thus, revocability is able to achieve.

Besides, the use of user-specific key under salting approach shows promising recognition performance through the increase in the inter-variation of different users' biometric information. However, salting approach usually suffered from accuracy or performance discrepancy problem under genuine-token and stolen-token scenarios (Kong et al. 2006). The genuine-token scenario refers to a secure individual storage case, whereby individual biometric data is mix with a user-specific password or token. This manifested a condition where the user specific password or token will never reveal or known by the third party. On the other hand, the stolen-token scenario refers to the case when same password or token is used to combine different users' biometric data. This manifested a condition similar to a given token is being stolen and used by the adversary to carry out any impersonation. The impersonation can be done by using he/she own biometric data with the stolen token to generate the cancelable template for matching with the formerly enrolled genuine user's cancelable template. Kong et al. (2006) have demonstrated that the impersonation success probability (also known as false accept rate, *FAR*) is indeed higher and the performance is significantly degraded under the stolen-token scenario. Therefore, the security of salting approach does not merely rely on the computational hardness in reverting the cancelable template itself but also relies on how the key (K) is being stored. Consequently, despite salting approach able to achieve non-invertibility, high recognition performance and suitable to be used in BTP, it also resulted in a token storage issues which requires to be solved.

1.8.3 Non-Invertible Transform

On the other hand, the non-invertible transformation mainly refers to a general way to generate the cancelable templates though the usage of a one-way transformation on the original biometric data. In principle, given a non-invertible transformation applied on a biometric data, a cancelable template can be easily generated (in a polynomial time $O(n)$), however, it is computationally hard to invert back to its original form (Nagar et al., 2010). Conventionally, given a non-invertible transformation used for cancelable template generation, a public random key, K is used to induce randomness for a given biometric data, hence new cancelable template is able to generate (revocability) and non-invertibility is satisfied. In contrast to salting approach, no user-specific key or token is involved under non-invertible transform. Besides, since K is publicly known, the security evaluation of a particular non-invertible transformation function always carry out under the scenario when K is being revealed or known by an adversary e.g. covered the lost key or token scenario. Therefore, the security of non-invertible transform approach does not rely on how the key or token is being stored. Consequently, the security of non-invertible transformation only relies on the computational hardness (usually in terms of brute force complexity) while an adversary trying to regenerate the original template in an inverting ways of the transformation process.

1.8.4 Biometric Cryptosystem

The biometric cryptosystem is initially proposed as a meant to secure a cryptographic key with biometric data (Jain et al, 2008). In a biometric cryptosystem, a helper data is usually stored in the database instead of the original template. The helper data refers to the public information, which is safe to be uncovered even though in a compromised database event. This helper data will be used to facilitate the extraction of the cryptographic key from a query biometric feature during the matching process. The validity of the cryptographic key extracted from the query biometric features is confirmed with a successful matching result. Depending on how the helper data is being obtained, the biometric cryptosystem can be further divided into two groups which are the **key binding** and **key generation**.

1.8.5 Key Binding

In key binding process, the given cryptographic key, K is bind with the enrolled template, T to extract the helper data, which will be stored in the database. In this case, the helper data, H is a function of the enrolled template with the bind cryptographic key ($H = \mathcal{F}(T, K)$). When comes to matching, the cryptographic key needs to be extracted from the query template, Q. This can be done by matching Q with the formerly enrolled template T, if Q is similar to T , the correctness properties of the key binding scheme guarantee the extraction of the exact bind cryptographic key, K with the help of the helper data. One notable example of key binding approach refers to the *Fuzzy Commitment* (FC) scheme proposed by Juels and Watternberg (1999). In a FC scheme, a person with a biometric data $x \in \mathcal{F}^n$, is used to generate a offset data

denoted as $\delta \in \mathcal{F}^n$, where $\mathcal{F}$ is a field. In principle, this off data δ can be generated by computing $\delta = c + K$ with a random code word $c \in \mathcal{F}^n$. After that, a one-way hash function $h(.)$ will be used to generate the hashed code word through one-way hashing described as $h(c)$. The hashed code word will be stored together with the offset data as a commitment denoted as $(h(c), \delta)$. In future authentication, given a biometric data x' which is close to x i.e. $d(x', x) < t$, with $d(.,.)$ refers to the hamming distance lower than certain threshold t, x' can be used to de-commit $(h(c), \delta)$ and recover the code word through reverse computation described as $x' - \delta = c'$. With the incorporated error correction code ECC e.g. BCH, Reed-Solomon code, c can be fully recovered from c'. In what follows, the condition $h(c) = h(c')$ can be satisfied, and consequently, yielding a successful De-commitment. It shows that the construction of FC scheme is straightforward and simplistic, with the possibility to be implemented by using a different kind of ECC. The FC scheme is provable secure under random oracle model (ROM) in such a way that, it is difficulty to reconstruct the code word c given an adversary has zero knowledge about c or x from the hashed output $h(c)$.

1.8.6 Key Generation

For key generation approach, a cryptographic key is extracted directly from the helper data instead of binding with the enrol template. Bodo (1994) has first presented the patented idea of key generation without actual implementation. Ideally, the extracted cryptographic key will always remain the same due to its correctness properties given the enrolled template is similar to the query template for a genuine matching. However, because of intrinsic

failure (e.g noises in a sensor or incomplete capture of biometric information), the resulted query template might not be exactly the same as the enrolled template even though in a genuine matching case. This issue will prone the system to reject the correct user due to the failure in cryptographic key identification. In order to tackle this problem, error correction code is normally used in a biometric cryptosystem to tolerate this kind of errors hence offers its correctness properties. Dodis et al. proposed the first provable secure key generation scheme named *Fuzzy Extractor* (FE) in year 2004. FE is mean to generate a usable random key from noisy data with a pair of extraction and reconstruction process denoted as $\langle \mathrm{Ext}, \mathrm{Rec} \rangle$. In principle, under the extraction process (Ext), it allows the extraction a uniform and random string $\boldsymbol{R}$ from $\boldsymbol{w}$ with securely high entropy to be used for personal authentication. In the same time, a help of helper string $\boldsymbol{P}$ is also generated that is publicly stored without compromising the security of $\boldsymbol{R}$. Given an input $\boldsymbol{w}'$ that is affected by noise and changes. Suppose $\boldsymbol{w}'$ is still close to $\boldsymbol{w}$ within a small threshold t under hamming distance measurement, i.e. $d(\boldsymbol{w}', \boldsymbol{w}) < t$, the reconstruction process (Rec) able to tolerance the noise added in an input biometric data $\boldsymbol{w}$. Thereafter, the string $\boldsymbol{R}$ can be re-constructed with helper string $\boldsymbol{P}$ to be used for any secure authentication of key encryption. The first actual implementation of a fuzzy extractor construction that come with flexible input size called Pinsketch is implemented by Kevin Harmon and Leonid Reyzin. It works with the principle by computing the symmetric difference of the input $\boldsymbol{w}$ and $\boldsymbol{w}'$ through generating a syndrome sketch $SS(\boldsymbol{w}) = syn(\boldsymbol{w}) = s$ and $SS(\boldsymbol{w}') = syn(\boldsymbol{w}') = s'$ by using error correction code i.e. BCH encoder. The input $\boldsymbol{w}$ can then be recovered through recovery process denoted as $\mathrm{Rec}(s, \boldsymbol{w}') = \boldsymbol{w}$

with BCH decoder provided the Hamming distant $d(s, s')) \leq t$ smaller than a certain threshold t.

1.9 Problem Statements

Although human iris recognition system comes along with certain benefits (e.g reliable, universality, convenient), its vulnerabilities in terms of security and privacy have drawn great attention. A few major problems have been outlined as follow:

1. **Privacy issues** – The Iris template stored in a database potentially to be attack or compromise. Venugopalan et al. (2011) have stated the method to generate spoofed iris images from IrisCode. Iris images can reveal the personal diseases such as free-floating iris cyst and diffuse iris melanoma (Zhou, 2012). When the database compromised, the adversaries can obtain all kind of private and sensitive information. This causes an inevitable privacy invasion. Besides, the adversaries can also gain illegitimate access to the systems, which results in an unauthorized access of privacy information (Jain et al., 2008).

2. **Irrevocability issues**- The information of human Iris is limited due to every person only possess one pair of eyes. The compromised templates cannot be easily revoked and further restricted the usage of human Iris in recognition purpose.

3. **Tradeoff between recognition performance and non-invertibility**- template protection methods has demonstrated a trade-off between non-invertibility and recognition (Nagar et al., 2010). This is due to the contradiction where non-invertibility requires to throw away as much information about the original template as possible while high system performance is achieved only when more discriminative information from the original templates are retained.

4. **Cross-matching issues**- Same biometric traits registered into multiple biometric applications are potentially linked. This leads to serious security and privacy problems when either one of the enrolled templates being attack or compromise.

5. **Security issues**- There existing the security issues in conventional Iris recognition system due to the potential security attacks, such as pre-image attack, and false accept attack. In fact, Jernish et al., (2011) have demonstrated a pre-image attack with only 60% of the IrisCode information is exploited. This security issues majorly inherited from the robust matching between different IrisCode that is at least 50% between different IrisCode will be matched (Dougman 2004, 2006). Therefore, this kind of security attack should be taken into consideration when designing a biometric template protection scheme.

1.10 Objectives

Based on our problem statement discussed above, we have formulated three objectives in this research.

1. To study various existing iris's template protection schemes.
2. To propose an iris's template protection scheme that is able to satisfy the four criterions of biometric template protection method, namely unlinkability, revocability, non-invertibility, and performance.
3. To analyze the generated cancelable templates in terms of security and performance.

1.11 Contributions

The contributions of this dissertation described in details as follow:

1. **New robust hashing technique for BTP**: A new iris's template protection scheme named Indexing First-One hashing is proposed to protect the iris template stored in a database. The proposed scheme has extended the well-known "Min-hashing" (Broder et al., 1998), by introducing a Hadamard multiplication and a modulo threshold function to achieve stronger non-invertibility of the newly generated iris template. Experiment results showed the effectiveness of the proposed scheme in generating cancelable iris template, which satisfied the four ideal cancelable biometric criteria as highlighted in the previous section.

2. **Efficient transformation and matching algorithm**: The proposed scheme has inherited the useful properties of Min-hashing. In this context, it offers an alternative to generating discriminative representation from IrisCode based on its implicit ordering by choosing the location of the first binary '1'. By doing so, the absolute binary information of IrisCode can be eliminated as much as possible hence higher security in terms of non-invertibility. Apart from this, the robustness of the proposed protection scheme contributed an alternative measurement of similarity between IrisCodes in a transformed real index feature domain, which equals to computing their Jaccard similarity. Moreover, we also show that IFO is able to extended into alignment-free IFO with Bloom-filter integration. The alignment-free IFO not only reduces the computational cost in the IFO template matching steps, it also preserve the recognition performance.

3. **Resolve token storage issues in BTP**: In order to further justify the performance and security of our proposed scheme, extensive experiments and analysis have been done under the assumption that the token/permutations used in our proposed scheme are being published. Results showed that even under token stolen condition (token being published), our proposed scheme experienced insignificant deterioration of the Equal Error Rate (*EER*) with just -0.16%. In this point of view, no user-specific token is required for performance preservation. Hence, the token/permutation storage issues as discussed in salting approach are resolved.

4. **Resistance to pre-image attack and false-accept attack**: Besides carryout the analysis for the four major BTP requirements, we also focus on the security analysis on the potential security attacks such as the pre-image attack and false accept attack. We demonstrated with in-deep analysis on these two attacks and show that IFO offers high resistance against them.

1.12 Dissertation Organization

The dissertation is organized as follow: In Chapter 2, a literature review, which includes the overview of biometric salting and non-invertible transformation, is discussed. Followed by Chapter 3, new iris's template protection framework is proposed and discussed in details. Thereafter, Chapter 4 discuss our experimental results and other analysis in revocability, diversity, non-invertibility and recognition performance of the proposed framework. In Chapter 5, a concluding remark about this research and a good description of our future focus are outlined.

CHAPTER 2

LITERATURE REVIEW

In this section, previous works for iris template protection have been revisited and summarized.

Ratha et al., (2007) was the first one to introduce the notion of cancelable biometric. They proposed to generate cancelable fingerprint template by applying a certain transformation to the original fingerprint template. In their work, they used random permutation to permute the original fingerprint template randomly in a Cartesian and polar domain to achieve revocability. Besides this, they also proposed the used of surface folding concept for minutia points remapping to generate cancelable fingerprint template. Their results showed high performance accuracy but the non-invertibility was not strong (Maltoni et al., 2009). However, their concept and works on cancelable biometric are later being used in iris template protection related researches.

In compiling the all previous works on iris templates protection, we have categorized them into salting and non-invertible transformation as discussed in Section 1.8.

2.1 Biometric Salting Approaches

An instance of iris salting scheme was first proposed by Chong et al. (2006) namely S-IrisCode encoding. To be specific, the iris Gabor-feature vector $\boldsymbol{\omega} \in \mathbb{C}^n$ was first generated by convoluting the 1-D log-Gabor filter with the normalized iris image which would be reshaped into a *n*-dimensional feature vector later. Then, the magnitude of $\boldsymbol{\omega}$, denoted as $\boldsymbol{w}$ was projected into a lower dimensional feature space through iterated inner products with a set of user-specific orthonormal random vectors $\{\boldsymbol{r}_{\perp i} \in \Re^n | i = 1, \dots, m\}$ where $m \leq n$. A quantization process was carried out to compute s_i from $\{\alpha = \langle \boldsymbol{w} | \boldsymbol{r}_{\perp i} \rangle | i = 1, \dots, m\}$ with $s_i = 0$ when $\alpha \leq 0$; $s_i = 1$ when $\alpha > 0$. Once compromised, a new cancelable template can be regenerated by issuing a new set of random vectors from the user-specific token. To improve the system performance, a noise mask $\{s_{iN} | i = 1, \dots, m\}$ was utilized with $s_{iN} = 0$ when $\alpha < -\sigma \; and \; \alpha > \sigma$ otherwise. The noise mask acts as the control bit to determine the validity of the s_i bits by eliminating the weak inner product, thus, improved the correctness in hamming distance matching.

Zuo et al. (2008) proposed a salting method which could be applied to either real-valued or binary iris patterns, namely GREY-SALT and BIN- SALT. In GREY-SALT, an artificial pattern was either added or multiplied to the iris pattern. For BIN- SALT, XOR operation was applied to the IrisCode to output a random binary key pattern. For both GREY-SALT and BIN-SALT, the iris information was concealed with the auxiliary data. Thus, cancelable iris template refreshment could be realized by replacing new auxiliary data.

However, the system performance of this method might significantly deteriorate without pre-alignment process.

Instead of using the whole iris image as in Chong et al. (2006), Pillai et al. (2010) used sectored random projections for cancelable iris template generation. They remarked that, by projecting the iris image directly via a user-specific random matrix may inevitably lead to performance deterioration due to noises such as eyelashes, specular reflection, and eyelid as well as inhomogeneous quality in different regions of the iris image. Thus, a linear transformation of iris regions with the noises corrupted the data. In their work, the iris region was partitioned into several sectors and then the Gabor features of each sector were projected into a lower dimensional space via a user-specific random Gaussian matrix. Lastly, the cancelable template was generated by concatenating the projected outputs from different sectors followed by a feature encoding process as in conventional iris recognition system (Daugman, 2004, 2006), (Masek, 2003). Their work compressed the original template while preserving the system performance. New templates can be generated by using different random projection matrices. Figure 2.1 shows the proposed sectored random projection technique.

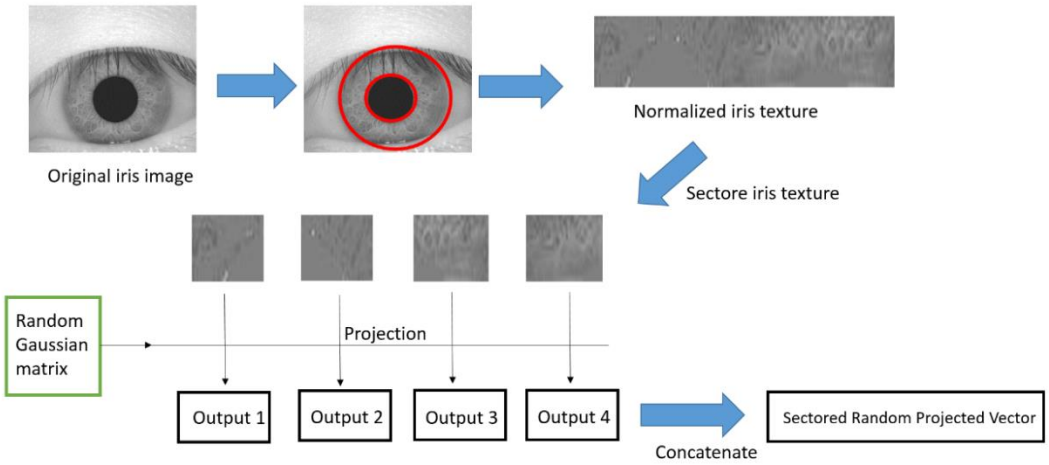

Figure 2.1: Sectored Random Projection

However, in salting approach, Kong et al. (2006) and Lacharme et al. (2012) showed that if the same random matrix was applied to different users, the system performance would degrade significantly and it was possible that the cancelable template could be inverted when the user-specific random matrices are disclosed to the attacker (stolen-token scenario). This implies that the biometric salting (e.g biohashing) can achieve promising results in recognition performance only under an undesired assumption that the user-specific secrets or token will never be stolen or shared.

2.2 Non-Invertible Transformation Approaches

For non-invertible transformation approaches, Zuo et al. (2008) proposed two non-invertible transformation methods, namely GREY-COMBO and BIN-COMBO for iris templates. In GREY-COMBO, they shifted the iris image in a row-wise manner via the random offset (random key), then followed by an operation (either addition or multiplication) on two randomly selected

rows. In BIN_COMBO, the same procedure was performed to IrisCode but with XOR or XNOR operation. In this manner, the original iris data was distorted attributed to the addition/multiplication operation between the two randomly selected row features, hence, fulfilled non-invertibility criterion. In both GREY-COMBO and BIN_COMBO, the shifted rows of the iris template were always in the same orientation regardless rotation, hence it is 'registration free', which implies that no alignment is needed for matching. However, the first method suffered from performance degradation when poor quality iris images were used. Nevertheless, since they used a user-specific key, this exposed to the risk of stolen-token as salting approach.

Hämmerle-Uhl et al. (2009) used block-remapping method to perform a non-invertible transformation. The normalized iris image was first partitioned into several image blocks and then randomly permuted by a key. An image block remapping technique was applied to generate a cancelable template. In this process, a target image, which is the same size as the source iris image was initialized. Then, different image blocks from the source image were mapped into the target image. Same image blocks were allowed for multiple times of remapping. The lossy remapping process prevents the reconstruction of original iris image and satisfies the non-invertibility criterion. Despite the scheme did not jeopardize the system performance, it does not meet the desirable system security level. For example, Jenisch et al. (2011) demonstrated that 60% of the original iris image can be restored from the stolen template and this potentially enable an adversary to get into the system with an approximately regeneration of the IrisCode (pre-image attack).

Ouda et al., (2010, 2011) proposed a tokenless IrisCode template protection scheme, namely Bio-encoding. They first determined the "consistent bits" from several IrisCodes of each user. The consistent bits refer to the bits that have a lower probability to be flipped among several samples collected. The consistent bits, $C \in \{1,0\}^n$ where n denotes the length of the consistent bit vector, where $n \in [332, 3737]$. In addition, a position vector $P \in \{1,0\}^n$ that recorded the position of the consistent bit in C was created and stored. Finally, the C was split into m binary codewords and each codeword was encoded to a randomly generated binary sequence $S \in \{1,0\}^l$ where $l = \frac{n}{m}$ to render BioCode, $B \in \{1,0\}^l$. For example, for $m = 5$, if the addressed codeword is 10011_m (decimal 19), then the corresponding BioCode bit will output either '0' or '1' according to the 19th-bit value in S. In this instant, same codewords will output the same bit resulted in a lossy many-to-one mapping, hence, non-invertibility is satisfied. To protect the scheme from correlation attack, a second random sequence was XORed or permutated with the IrisCode before Bio-encoding process. Their work also showed system performance preservation with respect to its unprotected counterpart. However, Lacharme (2012) pointed out that the non-invertible property of BioCode was invalid. The restoration is highly possible when the Boolean function, which used to generate the random sequence was exposed. Figure 2.2 shows the proposed Bio-Encoding technique.

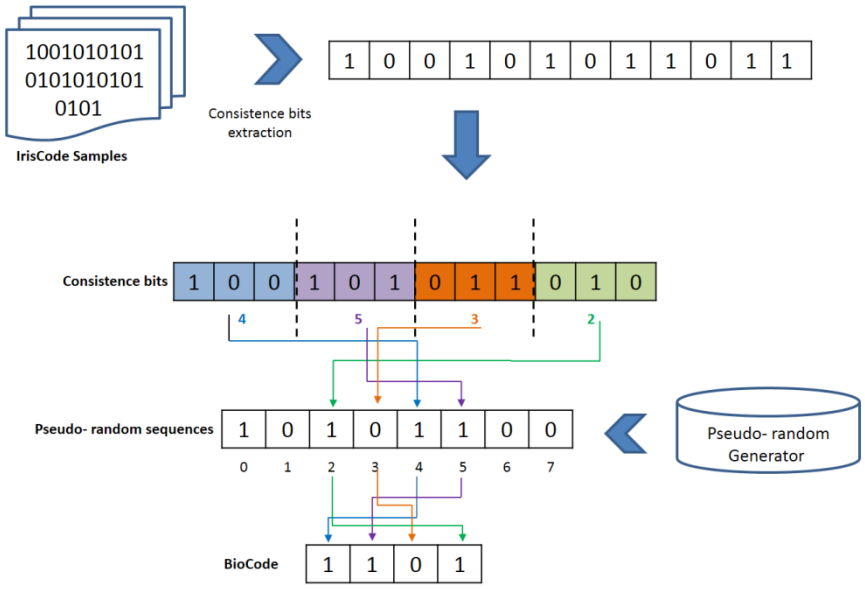

Figure 2.2: Bio-Encoding technique

Rathgeb et al. (2013) proposed a cancelable IrisCode method using Bloom filter. A Bloom filter b is a bit array of length n. The Bloom filter was initialized with zeros and formed by adding elements '1' into it using K independent hash function $h_1, h_2, h_3, \ldots, \ldots, h_K$ with range $h \in [0, n-1]$. In practice, instead of using K independent hash functions, they proposed using a binary to decimal mapping. The IrisCode with dimension $H \times W$ was first split into K blocks with size $l = \frac{W}{K}$, where l is the number of columns of each block. Each block B_i, $i \in [1, K]$ constituted to the formation of the Bloom filter b_i. This could be done by adding element '1' to b_i based on the position that manifested by each column codeword $x_j \in \{1,0\}^m$ inside B_i, where $j \in [1, l]$ and $1 \leq m \leq H$. Same x_j would map to the same element in the Bloom filter resulted a many-to-one mapping hence non-invertibility criterion was satisfied.

For cancelable template refreshment, they applied an application-specific secret key T like how other cancelable biometric schemes did.

The system performance of Bloom filter was comparable to its original counterparts. However, Hermas et al. (2014) pointed out that the template could be restored with low complexity of 2^{25}. They also presented an attack where two Bloom filters b_1 and b_2 generated from the same IrisCode could be identified with high probability at around 96%. Bringer et al. (2015) also stated that the un-linkability attack is highly possible due to the small key space, which was intended to preserve the system performance. Recent work done by M. Gomez-Barrero et al. (2016) showed how to prevent the cross-matching attacks in Bloom filter-based template protection schemes. Figure 2.3 shows that example of the Bloom-filter technique.

Dwivedi et al. (2015) proposed a cancelable iris template based on the look-up table. They first generated the rotation invariant iris template by shifting different samples of IrisCode left and right with respect to a reference template generated from the same user. Then, a single row vector $C \in \{1,0\}^{1\times N}$ was formed by appending every row of the rotation invariant code where N represents the length of the row vector. Then, C was further divided into l binary codewords, each codeword consists of m bits thus $l = \frac{C}{m}$. The decimal values represented by each codeword denoted as $d = \{d_i \in [0, 2^m - 1] | i = 1, \dots, l\}$ was recorded. A look-up table $M \in \{1,0\}^{R\times m}$ was generated randomly and $R \geq 2^m - 1$. By look-up mapping, d was encoded and yield cancelable

template. However, since the look-up table should be stored along with the cancelable templates, in the event of compromise and parameter *m* is disclosed, IrisCode can be easily restored. Figure 2.4 shows that example of look-up mapping process.

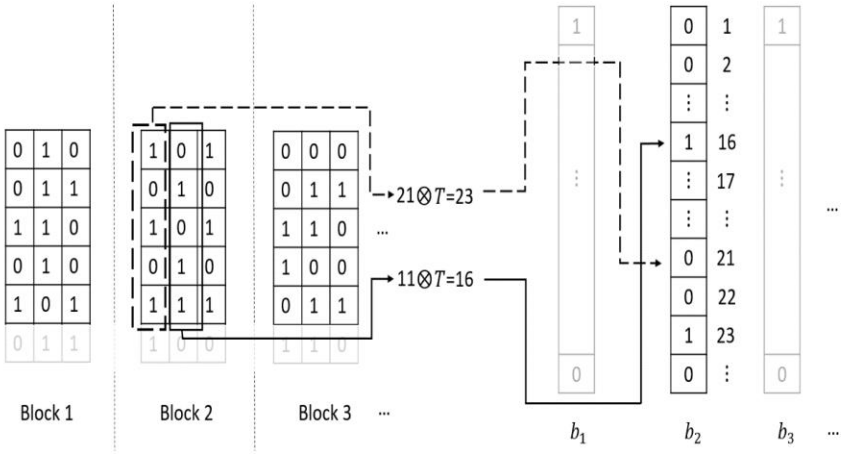

Figure 2.3: Example of Bloom filter technique

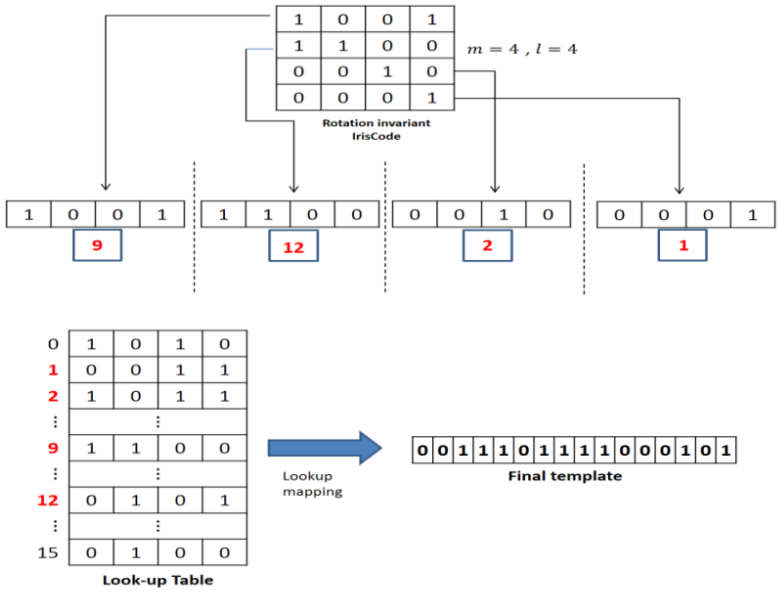

Figure 2.4: Look-up mapping process

Table 2 tabulated the summarized version of the well-known existing BTP schemes for iris recognition system. Their key techniques together with the main drawbacks are highlighted:

Proposed Method	**Input Iris Features**	**Key Technique**	**Main Drawbacks**
Chong et al. (2006)	Iris Gabor features	Generate S-iris code based on projection using random matrices	-User-Specific token storage issues
Zuo et al. (2008)	Iris image/ IrisCode	GREY Salt/ BIN salt GREY Combo/ BIN Combo	-User-Specific token storage issues -Performance degrades when input iris images are not in good quality
Hämmerle-Uhl et al. (2009)	Iris image	Image block permutation and remapping	-Vulnerable to False-match attack (Jenisch et al 2011)
Pillai et al. (2010)	Iris Gabor features	Sectored iris image random projection using random Gaussian matrices	-User-Specific token Storage issues
Osama et al. (2010)	Iris code	Consistent bits to random sequences mapping (Bio-Encoding)	-Spoofed Iris can be generated (Lacharme, 2012)
Rathgeb et al. (2013)	Iris code	Adaptive Bloom filter	-Key size is too small in preserving the recognition performance (J. Hermas et al. 2014) -Vulnerable to False-match attack (Bringer et al., 2015)
Dwivedi et al., (2015)	Iris code	Look-up table mapping	-Performance degraded -Look-up table storage issues

Table 2: Summarized existing BTP for iris recognition system

2.3 Summary

The vast majority of existing iris template protection schemes are able to maintain the recognition performance but still contain certain vulnerabilities. For example, the protection scheme based on Biometric salting technique (Chong et al., 2006), (Zuo et al., 2008), (Pillai et al., 2010), showed significant degradation in recognition performance when same user token was used (Kong et al., 2006). This implies the user-specific token used in salting technique requires to be securely stored in order to achieve better recognition performance. Although the revocability can be satisfied, this created another token storage problem based on security concern. On the other hand, although the BTP schemes proposed by (Hammerle-Uhl et al., 2009), (Rathgeb et al., 2013) are able to achieve non-invertibility of new iris template, false positive matching (False match attack) can be successfully launched (Jernish et al., 2011), (Hermes et al., 2014). This puts a question on the usability of the proposed scheme and the biometric system for personal authentication. To summarize the literature, new iris template protection scheme is required to compensate the current weakness and vulnerabilities of the existing template protection schemes. There is still room for improvement in this research.

CHAPTER 3

PROPOSAL FOR INDEXING-FIRST-ONE HASHING

3.1 Preliminaries

In this preliminaries section, we briefly introduce the background of local sensitive hashing and Min-hashing which are important concepts used in our proposed iris template protection scheme.

3.1.1 Local Sensitive Hashing

Indyk et al. (1997, 1978) have introduced the usage of local sensitive hashing (LSH) in similarity search. The main idea of LSH is to hash the given data points using multiple hash function, h, which derived from a local sensitive hashing function family, H. The use of multiple hash functions enable calculation of the distance between two points in term of collision probability. LSH ensures the points which are closer always having higher probability of collision in the hashed domain, but the points which are far apart will have lower probability of hash collision. The LSH family H can be defined as follows:

$$\begin{aligned} &\Pr_H\big(h_i(X) = h_i(Y)\big) \leq P_1\,, \qquad \text{if } |X - Y| > R_1, \\ &\Pr_H\big(h_i(X) = h_i(Y)\big) \geq P_2\,, \qquad \text{if } |X - Y| < R_2. \end{aligned} \tag{3.1}$$

Given that the probability $P_2 > P_1$, while $X, Y \in \mathbb{R}^d$, and $H = \{h: \mathbb{R}^d \rightarrow U\}$, where U is the hashed matrix space. While $h_i(.)$ refers to the ith hash function used. Eq. (3.1) described a LSH family that is (R_1, R_2, P_1, P_2) sensitive. The probability $\Pr_H\big(h_i(X) = h_i(Y)\big)$ is high when the distance between X and Y is small e.g. hamming distance between X and Y is small.

3.1.2 Min-Hashing

Min-hashing is first proposed by Broder et al. (1997, 1998) for the purpose of fast searching of similar documents or webpages. Typically, Min-hashing consider as an example of LSH (Indyk et al., 1999). A simple view of Min-hashing is basically encoded the first '1' appears for every permutation of the original input binary vectors which relates the collision rate of two different binary vectors $\boldsymbol{A}$ and $\boldsymbol{B}$ corresponding to the Jaccard similarity. Different binary vectors can be formed by using different permutation vectors. Let $i = 1, 2, \dots, \dots, m,$ while m denotes the number of permutation vectors used, the probability of vectors $\boldsymbol{A}$ and $\boldsymbol{B}$ to be equal after performing Min-hashing $(\min h_i(\boldsymbol{A}), \min h_i(\boldsymbol{B}))$ can be expressed as follows:

$$\Pr[\min h_i(\boldsymbol{A}) = \min h_i(\boldsymbol{B})] = JS(\boldsymbol{A}, \boldsymbol{B}). \tag{3.2}$$

The Jaccard similarity is given by $JS(\boldsymbol{A}, \boldsymbol{B}) = \frac{|\boldsymbol{A} \cap \boldsymbol{B}|}{|\boldsymbol{A} \cup \boldsymbol{B}|}$ and $0 \leq JS \leq 1$. When $JS = 1$, it indicates a perfect match. This Jaccard similarity estimation is bounded within an error ε when $m = \left(\frac{2}{\varepsilon^2}\right) \ln\left(\frac{2}{\delta}\right)$ which is described as below:

$$\Pr[|M - JS(\boldsymbol{A}, \boldsymbol{B})| > \varepsilon] < \delta, \qquad \text{where } M = \sum_{i=1}^{m} X_i \tag{3.3}$$

$$X_i = \begin{cases} 1, & \min h_i(\boldsymbol{A}) = \min h_i(\boldsymbol{B}), \\ 0, & \text{otherwise}. \end{cases} \tag{3.4}$$

Based on Eq. (3.3) and Eq. (3.4), the Jaccard similarity estimated with at least ε error with probability less than δ when the number of permutations used is $m = \left(\frac{2}{\varepsilon^2}\right) \ln\left(\frac{2}{\delta}\right)$. For better Jaccard estimation, the number of permutations (m) needs to be increased. Figure 3.1 shows an example of Min-hashing process with two ($m = 2$) hash functions $h_1(.)$ and $h_2(.)$.

3.1.3 IrisCode Generation

Our research focuses on the protection of IrisCode which is stored inside the database. Hereby, to generate the IrisCode from every user, we adopted the technique proposed by Rathgeb et al., (2013) and Uhl et al., (2012) to generate the IrisCode. In this context, the iris region is first detected by applying the weighted adaptive Hough transform. After that, a 2-stages segmentation process is used to segment the iris and pupil boundaries (Uhl et al., 2012). Followed by a normalization process to unwrap the iris region into a fixed dimension array namely rubber sheet model (Daugman, 2006). The normalized iris texture is enhanced as a rectangular texture with 64×512 pixels. The lower fourteen rows are eliminated during the feature extraction process due to the Log-Gabor feature extractor used in USIT tool only extracts and processed the upper 50×512 pixels (Rathgeb et al., 2013). This lead to the

formation of the new iris texture with size of 50×512 pixel; the pixels of every five rows are averaged and resulted a new one-dimensional signal. Each one-dimensional signal is convoluted with 1-D log Gabor filter to output a complex iris Gabor-features with size of 10×512. Finally each complex value of the iris Gabor-features are phase-quantized into 2 binary bits to generate the IrisCode $\boldsymbol{X} \in \{0,1\}^{n_1 \times n_2}$, with $n_1 = 20$ and $n_2 = 512$, which resulting a total of 10240 bits.

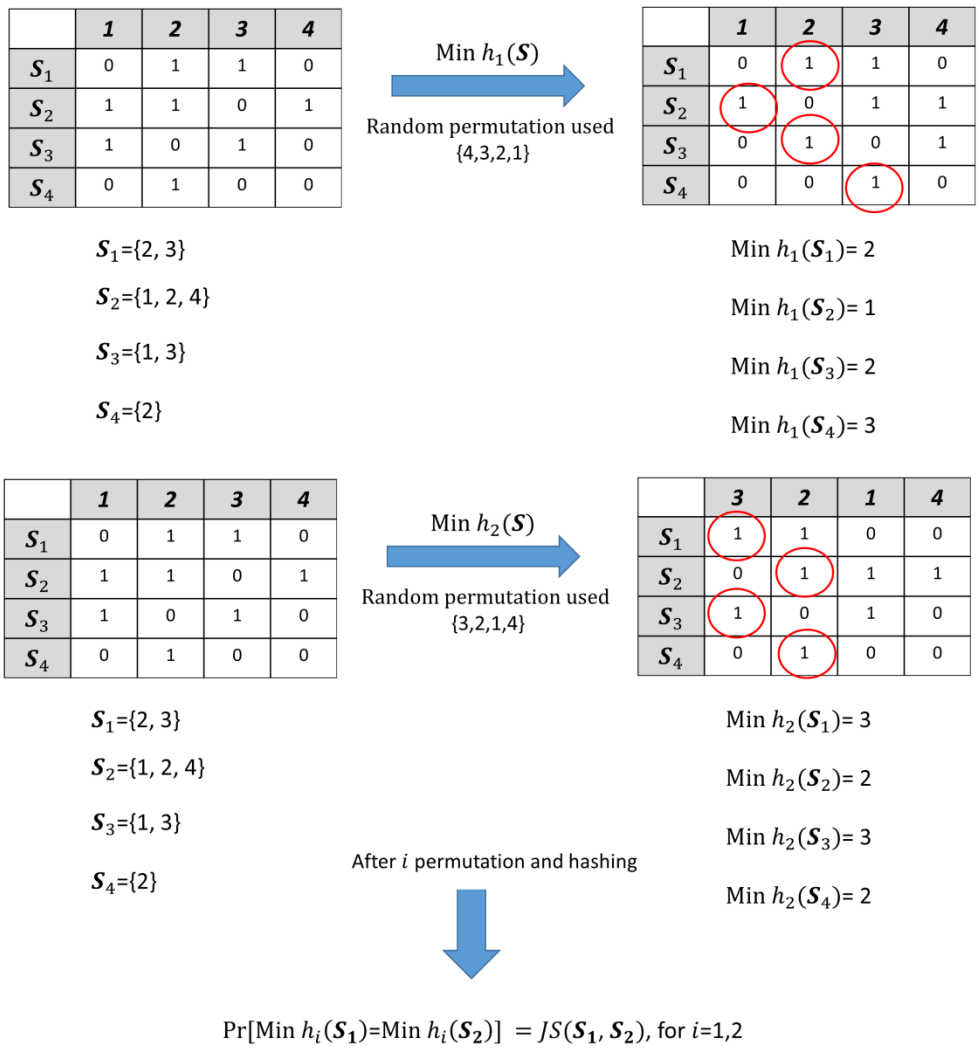

Figure 3.1: The Min-hashing algorithm with two hash functions

3.2 Indexing First-One Hashing

In order to protect the IrisCode inside the database, a new robust hashing technique is proposed, namely Indexing First-One (IFO) hashing to generate non-invertible iris template from the IrisCode.

IFO hashing is essentially an extension of Min-hashing coupled with Hadamard multiplication and modulo thresholding. Similar to Min-hashing, the IFO hashing utilizes m independent hash functions $h_1, \dots, h_m$ where each independent hash function is derived from P number of tokenized permuted IrisCode, $\boldsymbol{X}$ in column-wise manner. Both m and P can be set within the range $[1, \infty)$. The procedure of deriving IFO hashing function, $H(\boldsymbol{X}) = \{h_i(\boldsymbol{X}) | i = 1, \dots, m\}$ is described as follows:

1. Random Permutation: Generate a permutation set θ contains P number of random generated permutation vectors. Permute the input IrisCode X column-wise yields $X' = \{X'_l | l = 1, \dots, P\}$.

2. *Hadamard product code generation*: Generate Pth-ordered Hadamard multiplication product code $\boldsymbol{X}^P$ by multiplying (conjunction) all the $\boldsymbol{X}$'s, ie. $\boldsymbol{X}^P = \prod_{l=1}^{p} (\boldsymbol{X}'_l)$. Huge amount of binary information lost during this process to prevent IrisCode restoration. The permutation process and Hadamard multiplication also enable the exclusion of certain "fragile" bits (Hollingsworth et al. 2009).

3. *Construct the K-window*: For each row in the product code $\boldsymbol{X}^P$, select the first K elements, where $1 \leq K \leq n_2$. This step again throws away the binary information beyond the K-window.

4. Among the selected first K elements, record the index value, denoted as C_X corresponding to the first occurrence bit '1'.

5. *Modulo thresholding*: A modular threshold function is imposed to alleviate the leakage of $\boldsymbol{X}$ by means of a security threshold value τ, $1 \leq \tau < K$. That is, for every $C_X \geq K - \tau$, compute $C'_X = C_X \bmod (K - \tau)$. The imposed modulo threshold induced a many-to-one mapping for the output C_X hence strengthen the non-invertibility properties.

6. Repeat Step 1 to 5 with different permutations set $\theta_{(i,l)}$, while $i \in [1, m]$, $l \in [1, P]$ to form $n_1 \times m$ IFO hashed code, $\boldsymbol{C}'_X = \{ C'_{Xi} | i = 1, \ldots, m\}$, where $C'_{Xi} \in [0, K - \tau - 1]$. When $m < n_2$, it results a dimension reduction of the IFO hashed code generated as compared to the original input IrisCode $\boldsymbol{X} \in \{0,1\}^{n_1 \times n_2}$. Each round from Step 1 to Step 5 refers to the independent hashing function h_i for $i \in [1, m]$.

Figure 3.2 shows a toy example of the IFO hashing for a single row of IrisCode by using three hash function ($m = 3$) and parameters setting of $K = 3$, $P = 2$, and $\tau = 1$. Besides, **Algorithm** below depicts the pseudo-code of IFO hashing for IrisCode. The permutation token $\theta_{(i,l)}$, for $i \in [1, m]$, $l \in [1, P]$ used in IFO hashing is composed of m permutation sets, and each

permutation set contains P permutation vectors. Each permutation set is used for each hash function $h_i(\boldsymbol{X})$ to output $\boldsymbol{C'_{Xi}}$. In the event of IFO hashed code is compromised, m number of hash functions $h(.)$ are regenerated with different random permutation tokens to replace the compromised hash functions. A new template can be re-issued by following **Algorithm**.

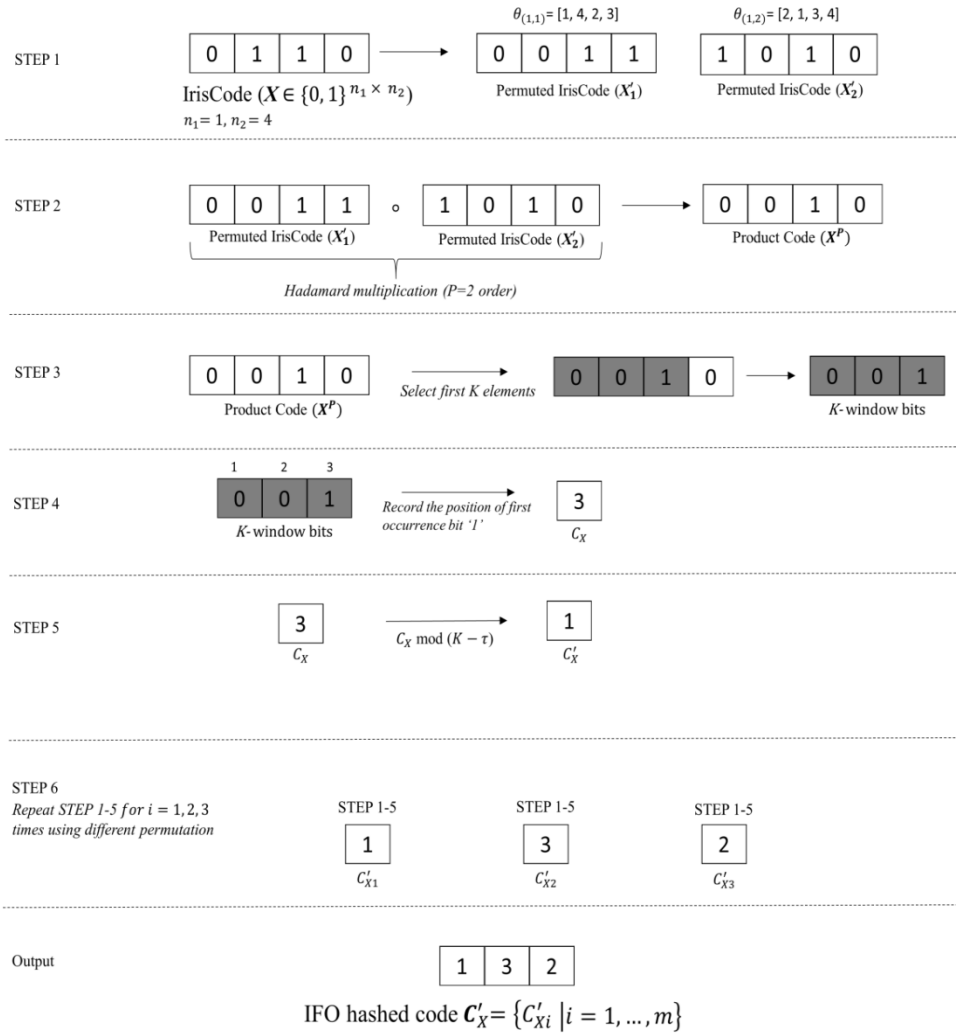

Figure 3.2: Toy example of IFO hashing based on three hash functions

Input Window size K, Permutation token $\theta_{(i,l)}$, number of permutations m, IrisCode $\boldsymbol{X} \in \{0,1\}^{n_1 \times n_2}$, security threshold τ
For each row of iris code $\boldsymbol{X}$: *for* i=1 to m Initialize i^{th} hashed code C_i to 0. 1. Permute elements of $\boldsymbol{X}$ according to $\theta_{(i,l)}$, $l \in [1, P]$ 2. Hadamard multiplication: Set $\boldsymbol{X}^P = \prod_{l=1}^{p}(\boldsymbol{X}'_l)$ 3. Construct K-window: *for* $j = 1$ to K 4. Locate the first occurrence bit '1' $if\ X^P(j) > C_i(j), then\ C_i = j$ $End\ if$ 5. Modulo thresholding: Compute $C'_i = C_i\ mod\ (K - \tau)$ *End for* *End for*
Output IFO hashed code, $\boldsymbol{C'_X}$, $\boldsymbol{C'_X} = \{\boldsymbol{C'_{Xi}} \mid i = 1, \dots, m\}$, $\boldsymbol{C'_X} \in [0, K - \tau - 1]$

Algorithm: IFO hashing for IrisCode

3.3 Matching

The matching of IFO hashed codes is a two-step process, which comprises of pre-alignment step and similarity matching.

3.3.1 Pre-alignment

Before matching takes place, the rotational inconsistency issue of the iris images due to head tilt of a person during the acquisition need to be addressed. Hence, a pre-alignment step is required prior to IFO hashing.

The pre-alignment is carried out by shifting the query IrisCode $\boldsymbol{Y}$ to left and right with ± 16 bits. Then, the IFO hashing is applied to each shifted IrisCode along with the original non-shifted IrisCode, yielding 33 shifted query instances. The matching is carries out between the enrolled hashed code and each of the query instances. Among all the matching results, only the highest

score will be recorded, which indicates the finalised matching result between the queries hashed code ($\boldsymbol{C'_Y}$) and the enrolled hashed code ($\boldsymbol{C'_X}$). Figure 3.3 shows an example of the pre-alignment and matching processes with left and right shifted (± 1 bit) with three shifted query hashed code ($\boldsymbol{C'_Y}$) and the enrolled hashed code ($\boldsymbol{C'_X}$).

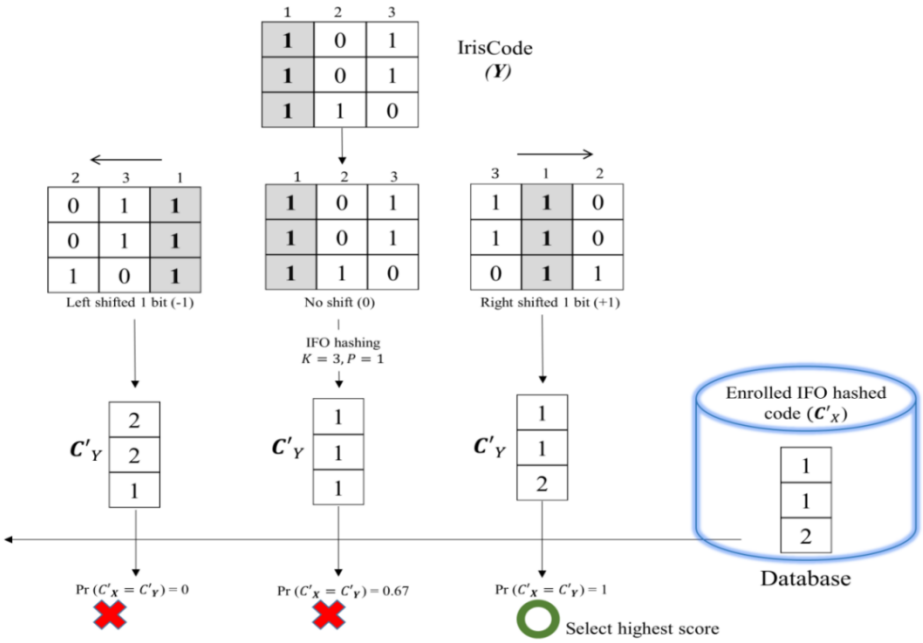

Figure 3.3: Example of pre-alignment process

3.3.2 Relation to Jaccard Similarity

IFO hashing scheme inherits useful properties of Min-hashing in which the similarity of $\boldsymbol{C'_X}$ and $\boldsymbol{C'_Y}$ can be estimated based on the probability of their hashed codes to become identical, i.e. $\Pr[h_i(\boldsymbol{X}) = h_i(\boldsymbol{Y})]$. This section will discuss how this probability is highly related to the Jaccard similarity.

The IFO hashing is applied to IrisCode in row-wise manner, to measure $\Pr[h_i(\boldsymbol{X}) = h_i(\boldsymbol{Y})]$, the probability of $C_{Xi} = C_{Yi}$ needs to be identified for each row. Let $\boldsymbol{X}^P$ and $\boldsymbol{Y}^P$ be the Hadamard multiplication product codes of P times permuted IrisCode $\boldsymbol{X}$ and $\boldsymbol{Y}$ respectively. Besides, let $\boldsymbol{X}_n^P$, $\boldsymbol{Y}_n^P$ represent the nth bit in the K-window of $\boldsymbol{X}^P$ and $\boldsymbol{Y}^P$ respectively, where $n \in [1, K]$. For instance, under a *single hashing* shows in Figure 3.4, starting from $n = 1$ (referring to the first K-window bit in both $\boldsymbol{X}^P$ and $\boldsymbol{Y}^P$), there only exist four possible permutation outputs. First of all, we know that $h(\boldsymbol{X}) = h(\boldsymbol{Y})$ if $\boldsymbol{X}_1^P = \boldsymbol{Y}_1^P = 1$. On the other hand, $h(\boldsymbol{X}) \neq h(\boldsymbol{Y})$ if $\boldsymbol{X}_1^P = 1, \boldsymbol{Y}_1^P = 0$ or $\boldsymbol{X}_1^P = 0, \boldsymbol{Y}_1^P = 1$. Lastly, there is an inconclusive case when $\boldsymbol{X}_1^P = 0, \boldsymbol{Y}_1^P = 0$.

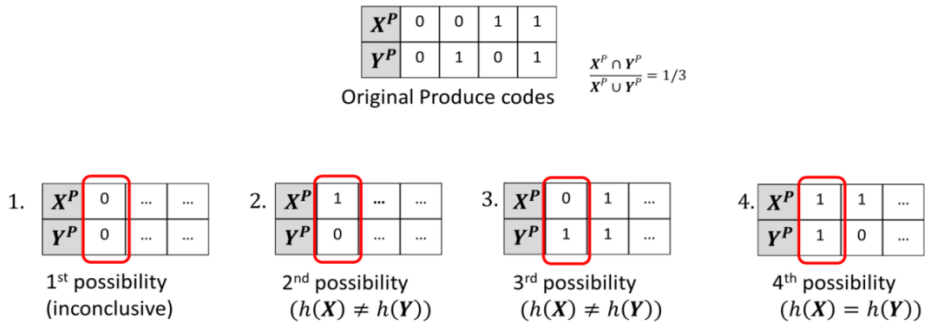

Figure 3.4: Possible permutation outputs under the Hadamard multiplication product codes with same permutation

Figure 3.4 shows the example of possible permutation outputs under Hadamard multiplication product codes $\boldsymbol{X}^P \in \{0,1\}^{1\times 4}$ and $\boldsymbol{Y}^P \in \{0,1\}^{1\times 4}$ (same permutation used to generate $\boldsymbol{X}^P$ and $\boldsymbol{Y}^P$ with single hashing).

As shown in Figure 3.4, only Case.1 (1st possibility) is inconclusive and require to further examine on the next consecutive bit position inside the K-

window (if we choose $K > 1$). From here, we know that for single IFO hashed elements to be the same, we have the following equation described the collision probability as:

$$\Pr\left(h(\boldsymbol{X}) = h(\boldsymbol{Y})\right) = \frac{Case(4)}{Case(2)+Case(3)+Case(4)}. \tag{3.5}$$

Thereafter, we have following lemma to describe the collision probability in IFO hashing.

Lemma 3.1: *Given two IFO hashed elements* $C_X \in [1, K]$ *and* $C_Y \in [1, K]$ *generated from IrisCodes* $\boldsymbol{X}$*and* $\boldsymbol{Y}$ *respectively, suppose the Jaccard similarity described as* $JS(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{3}$*, thereby we have* $\Pr(C_X = C_Y) = \frac{1}{\frac{2}{\alpha^P}-1}$ *for any constant* $\alpha \in (0,1)$ *and* $P \in [1, \infty)$.

Proof: Let α_X and α_Y denotes the probability of the particular bit in the IrisCode $\boldsymbol{X}$ and $\boldsymbol{Y}$ to be '1' respectively. Therefore we have ${\alpha_X}^P$ and ${\alpha_Y}^P$ refer to the probability of the particular bit in the Hadamard multiplication product code described as $\boldsymbol{X}^P$ and $\boldsymbol{Y}^P$ (inside K window) to be '1' respectively. All the while, $\boldsymbol{X}$ and $\boldsymbol{Y}$ should generate under same method and exhibit same characterization. We describe this same characterization by using same α value for α_X and α_Y, so we let $\alpha^P = {\alpha_X}^P = {\alpha_Y}^P$ where $\alpha \in (0,1)$, hence:

$$\frac{Case(4)}{Case(2)+Case(3)+Case(4)} = \frac{\mathbb{E}(Case(4))}{\mathbb{E}(Case(2))+\mathbb{E}(Case(3))+\mathbb{E}(Case(4))}$$

$$= \frac{{\alpha_X}^P {\alpha_Y}^P (\boldsymbol{X} \cap \boldsymbol{Y})}{{\alpha_X}^P (1-{\alpha_Y}^P) N_1 + (1-{\alpha_X}^P){\alpha_Y}^P N_2 + {\alpha_X}^P {\alpha_Y}^P (\boldsymbol{X} \cap \boldsymbol{Y})}$$

$$= \frac{\alpha^{2P}(\boldsymbol{X} \cap \boldsymbol{Y})}{\alpha^P (N_1+N_2) - \alpha^{2P}(N_1+N_2) + \alpha^{2P}(\boldsymbol{X} \cap \boldsymbol{Y})} \tag{3.6}$$

In principle, $N_1 + N_2 = \boldsymbol{X} \cup \boldsymbol{Y} - \boldsymbol{X} \cap \boldsymbol{Y}$. Suppose we let $\frac{\boldsymbol{X} \cap \boldsymbol{Y}}{\boldsymbol{X} \cup \boldsymbol{Y}} = \frac{1}{3} = JS(\boldsymbol{X}, \boldsymbol{Y})$, with Hadamard multiplication imposed of degree $P \in [1, \infty)$, and $\alpha \in (0,1)$ denoted the probability for a bit inside the IrisCode to be '1' or '0'. Eq. (3.6) can further simplify as following Eq. (3.7) and prove the lemma:

$$\frac{\alpha^{2P}(\boldsymbol{X} \cap \boldsymbol{Y})}{\alpha^P (\boldsymbol{X} \cup \boldsymbol{Y} - \boldsymbol{X} \cap \boldsymbol{Y}) - \alpha^{2P}(\boldsymbol{X} \cup \boldsymbol{Y} - 2\boldsymbol{X} \cap \boldsymbol{Y})} = \frac{1}{\frac{2}{\alpha^P} - 1}, \text{for } \frac{\boldsymbol{X} \cap \boldsymbol{Y}}{\boldsymbol{X} \cup \boldsymbol{Y}} = \frac{1}{3} \tag{3.7}$$

∎

Given the case where $JS(\boldsymbol{X}, \boldsymbol{Y}) \neq \frac{1}{3}$, $\Pr(C_X = C_Y)$ can still be calculated as described in Eq. (3.6). Based on Lemma 3.1, we able to further describe the IFO matching mechanism (without modulo thresholding) with the following theorem

Theorem 3.1: *Suppose we have* $\alpha \in (0,1)$, and $P \in [1, \infty)$. *Given a pair of constant value* (α, P), *for two IFO hashed elements* C_X *and* C_Y *generated from feature* $\boldsymbol{X} \in \{0,1\}^{n_1 \times n_2}$ *and* $\boldsymbol{Y} \in \{0,1\}^{n_1 \times n_2}$ *respectively,* $Pr(C_X = C_Y) = p\frac{|\boldsymbol{X} \cap \boldsymbol{Y}|}{|\boldsymbol{X} \cup \boldsymbol{Y}|}$, *with a positive constant scale factor* $p \in (0,1]$.

Proof: From Lemma 3.1, we have $\Pr(h(\boldsymbol{X}) = h(\boldsymbol{Y})) = \Pr(C_X = C_Y) = \frac{1}{\frac{2}{\alpha^P}-1}$. For the simplest case, i.e. $P = 1$ (without Hadamard multiplication & modulo thresholding), it is easily verified that given $\alpha = \frac{1}{2}$, we get $\Pr_{P=1}\big(h(\boldsymbol{X}) = h(\boldsymbol{Y})\big) = \frac{1}{\frac{2}{\alpha}-1} = \frac{\boldsymbol{X}\cap\boldsymbol{Y}}{\boldsymbol{X}\cup\boldsymbol{Y}} = \frac{1}{3}$ reduced to the Min-hashing case with $\frac{\boldsymbol{X}\cap\boldsymbol{Y}}{\boldsymbol{X}\cup\boldsymbol{Y}} = \frac{1}{3}$ described in Lemma 3.1. In this context, a scaling factor p is introduced through finding the ratio of IFO case over Min-hashing case described as:

$$p = \frac{\Pr(h(\boldsymbol{X}) = h(\boldsymbol{Y}))}{\Pr_{P=1}\big(h(\boldsymbol{X}) = h(\boldsymbol{Y})\big)} = \frac{\frac{2}{\alpha} - 1}{\frac{2}{\alpha^P} - 1} \tag{3.8}$$

One easily verify that for $P \to \infty$, $p = 0$; when $P = 1$ then$p = 1$. Hence $p \in (0, 1]$ is a positive constant characterized by a constant $\alpha \in (0,1)$ and different degree of Hadamard multiplication imposed $P \in [1, \infty)$. In this context, the probability of Cases (2)-(4) indicates a downscale on Jaccard similarity with $p \in (0,1]$ refers to the downscaled factor for the original Jaccard measure in Min-hashing hence prove the theorem. ▌

In practice, Theorem 3.1 is also applicable for the case when modulo thresholding is applied individually for each hashed elements. This can be easily explain as follow. Let the modulo thresholding denotes as a function $f(.)$, therefore, the new collision probability (with modulo thresholding) can be described as $\Pr(f(h(\boldsymbol{X})) = f(h(\boldsymbol{Y})))$. Particularly, we have:

$$\Pr\left(f\left(h(\boldsymbol{X})\right) = f\left(h(\boldsymbol{Y})\right)\right) \tag{3.9}$$

$$= \Pr\left[f\left(h(\boldsymbol{X})\right) = f\left(h(\boldsymbol{Y})\right) \middle| h(\boldsymbol{X}) = h(\boldsymbol{Y})\right]$$

$$+ \Pr\left[f\left(h(\boldsymbol{X})\right) = f\left(h(\boldsymbol{Y})\right) \middle| h(\boldsymbol{X}) \neq h(\boldsymbol{Y})\right]$$

Suppose that, $\Pr\left[f\left(h(\boldsymbol{X})\right) = f\left(h(\boldsymbol{Y})\right) \middle| h(\boldsymbol{X}) \neq h(\boldsymbol{Y})\right] = \varepsilon(\Pr\left(h(\boldsymbol{X}) = h(\boldsymbol{Y})\right) | h(\boldsymbol{X}) \neq h(\boldsymbol{Y}))$ with negligible probability ε, then the following equation shows that the collision probability able to preserve after modulo thresholding is applied:

$$\Pr\left(f\left(h(\boldsymbol{X})\right) = f\left(h(\boldsymbol{Y})\right)\right) \tag{3.10}$$

$$= 1 - \varepsilon(\Pr\left(h(\boldsymbol{X}) = h(\boldsymbol{Y})\right) | h(\boldsymbol{X}) = h(\boldsymbol{Y}))$$

$$+ \varepsilon(\Pr\left(h(\boldsymbol{X}) = h(\boldsymbol{Y})\right) | h(\boldsymbol{X}) \neq h(\boldsymbol{Y}))$$

$$\approx \Pr(h(\boldsymbol{X}) = h(\boldsymbol{Y}))$$

Eq. (3.10) shows that Theorem 3.1 is also applicable for the case when modulo thresholding is applied given that ε is negligible small.

3.3.3 Definition and Theorem for IFO Matching

From previous section, we known that $\Pr(C_X = C_Y) = S(\boldsymbol{X}, \boldsymbol{Y})$ whereby $S(\boldsymbol{X}, \boldsymbol{Y}) = p\frac{|\boldsymbol{X} \cap \boldsymbol{Y}|}{|\boldsymbol{X} \cup \boldsymbol{Y}|}$. Let $P_S = S(\boldsymbol{X}, \boldsymbol{Y})$, and z denotes the number of agreed position between two IFO hashed code. With more than one IFO hashing function, under random oracle model, $z \sim \text{Bin}(m, P_S)$ follows a binomial

distribution for $i = 1,2, \dots, m$, with $\mathbb{E}(z) = mP_S$ and standard deviation $\sigma = \sqrt{mP_S(1 - P_S)}$. Therefore, we have:

$$\Pr[h_i(\boldsymbol{X}) = h_i(\boldsymbol{Y})] = \Pr[C_{Xi} = C_{Yi}] = \frac{\mathbb{E}(z)}{m} = P_S \tag{3.11}$$

We further characterize the matching between different IFO hashed code in the following theorem:

Theorem 3.2: *Given a fixed* m*, and* $z' = \lfloor \delta m \rfloor$ *refer to fraction of* m *with* $\delta \in [0,1]$*, a query IFO hashed code* $\boldsymbol{C'_Y}$ *collide with a enrolled IFO hashed code* $\boldsymbol{C'_X}$ *in at least* z' *elements comes with overwhelming probability* P_1 *for* $R_1 \leq \frac{z'}{m}(1 - \varepsilon)$*, and close to zero probability* P_2 *for* $R_2 \geq \frac{z'}{m}(1 + \varepsilon)$ *with a small deviation* ε*.*

Proof: Let $R = D(\boldsymbol{X}, \boldsymbol{Y})$ denoted the difference between the IrisCode used to generate $\boldsymbol{C'_Y}$ and $\boldsymbol{C'_X}$. Particularly, with LSH convention and Theorem 3.1, we can define $D(\boldsymbol{X}, \boldsymbol{Y}) = 1 - S(\boldsymbol{X}, \boldsymbol{Y}) = 1 - p\frac{|\boldsymbol{X} \cap \boldsymbol{Y}|}{|\boldsymbol{X} \cup \boldsymbol{Y}|}$. We hereby can proof Theorem 3.1 straightforwardly with an experiment by using different value of $\delta \in [0,1]$. In this case, we created four different plots for $\mathbb{P}(z \geq z')$ vs $1 - D(\boldsymbol{X}, \boldsymbol{Y})$, under arrangement of $\delta = [\frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}]$.

The output is an S curve shows in Figure 3.5 with the deepest point of the curve can be determined by calculating $\frac{z'}{m}$. Besides, with a small deviation

of ε away from the deepest point on the X-axis, it gives us that P_1 comes with overwhelming probability (close to 1) under a difference $D(\boldsymbol{X}, \boldsymbol{Y})$ denoted as $R_1 \leq \frac{z'}{m}(1-\varepsilon)$. On the opposite direction, P_2 comes with probability close to zero when $R_2 \geq \frac{z'}{m}(1+\varepsilon)$ given that $R_1 < R_2$ and $P_1 > P_2$, hence prove the theorem.

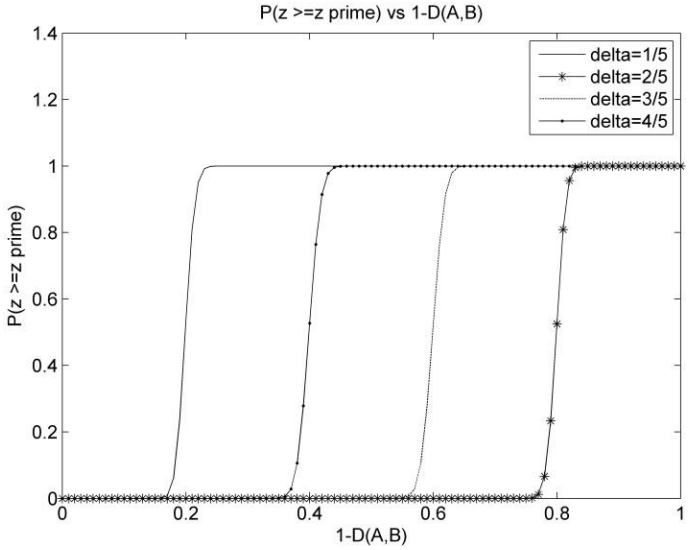

Figure 3.5: $\mathbb{P}(z \geq z')$ vs $1 - D(\boldsymbol{X}, \boldsymbol{Y})$

Based on Theorem 3.2, we give the following definition to generally describe the matching between different IFO hashed code under a given collision threshold $z' = \lfloor \delta m \rfloor$.

Definition 3.1: *Given a query IFO hashed code $\boldsymbol{C}'_Y$ generated from IrisCode $\boldsymbol{Y}$, and a enrolled IFO hashed code $\boldsymbol{C}'_X$ generated from IrisCode $\boldsymbol{X}$, the matching between $\boldsymbol{C}'_Y$ and $\boldsymbol{C}'_X$ is (P_1, P_2, R_1, R_2)- sensitive in the sense that $\boldsymbol{C}'_Y$ can always match with $\boldsymbol{C}'_X$ with overwhelming probability P_1 if $D(\boldsymbol{X}, \boldsymbol{Y}) \leq R_1$ and fail with probability P_2 close to zero if $D(\boldsymbol{X}, \boldsymbol{Y}) \geq R_2$ for $R_1 < R_2$ and $P_1 > P_2$.*

3.3.4 Matching in Practice

In this section, we discuss how we compute $\Pr[h_i(\boldsymbol{X}) = h_i(\boldsymbol{Y})]$, which is equivalent to the downscaled Jaccard similarity of $\boldsymbol{X}$ and $\boldsymbol{Y}$ in practice.

Let $\boldsymbol{C}'_X$ ($\boldsymbol{C}'_Y$) be the 2D IFO hashed code of the enrolled (query) IrisCode $\boldsymbol{X}$ ($\boldsymbol{Y}$). Each element inside the 2D IFO hashed code denoted as $C'_{X(j,i)}$ ($C'_{Y(j,i)}$) where $j \in [1, n_1]$, and $i \in [1, m]$. To compute the probability of two hashed codes become identical (P_1), it can be done by searching through the enrolled and query hashed codes and estimate the probability of $C'_{X(j,i)} = C'_{Y(j,i)}$. Note that there is an ill-case where bit '1' is absent in K-window especially K is small enough. In this case, it will result to $C'_{X(j,i)} = 0$. In practice, we can represent the matching algorithm in binary domain for simplicity. In order to exclude the ill case during matching, we first initialize a zeros binary matrix, i.e. $\boldsymbol{B}_X$ ($\boldsymbol{B}_Y$) $\in \{0,1\}^{n_1 \times m}$ and fill with '1' only if $C'_{X(j,i)}(C'_{Y(j,i)}) \neq 0$. On the other hand, we introduce another binary matrix, $\boldsymbol{Q}_{XY}$ which is also initialized with zero. Then, given $\boldsymbol{C}'_X$ and $\boldsymbol{C}'_Y$, $Q_{X(j,i)Y(j,i)}$ is set to '1' if $C'_{X(j,i)} = C'_{Y(j,i)}$ for $j \in [1, n_1]$, and $i \in [1, m]$. Finally, the similarity of $\boldsymbol{C}'_X$ and $\boldsymbol{C}'_Y$ can be calculated as:

$$S(\boldsymbol{C}'_X, \boldsymbol{C}'_Y) = \frac{|\boldsymbol{Q}_{XY}|}{|\boldsymbol{B}_X \cap \boldsymbol{B}_Y|} = P_S, |\boldsymbol{B}_X \cap \boldsymbol{B}_Y| \neq 0 \tag{3.12}$$

Since $0 \leq |\boldsymbol{Q}_{XY}| \leq |\boldsymbol{B}_X \cap \boldsymbol{B}_Y|$, thus, $S(\boldsymbol{C}'_X, \boldsymbol{C}'_Y) \in [0,1]$. When $S(\boldsymbol{C}'_X, \boldsymbol{C}'_Y) = 1$ indicates a perfect similar match. Eq. (3.12) is equivalent to calculate $\Pr[h_i(\boldsymbol{X}) = h_i(\boldsymbol{Y})] = p\frac{|\boldsymbol{X} \cap \boldsymbol{Y}|}{|\boldsymbol{X} \cup \boldsymbol{Y}|}$. The $\boldsymbol{Q}_{XY}$ is divided by $\boldsymbol{B}_X \cap \boldsymbol{B}_Y$ instead of the size of IFO hashed code is to exclude the ill-case. Note that $\boldsymbol{B}_X \cap \boldsymbol{B}_Y$ is used instead of $\boldsymbol{B}_X \cup \boldsymbol{B}_Y$ as the former ensures that the collision between $C'_{X(j,i)}$ and $C'_{Y(j,i)}$ is non-zero. In contrast, the latter considers either $C'_{X(j,i)} = 0$ or $C'_{Y(j,i)} = 0$, hence the reasoning of non-collision is invalid with ill-case.

In practice, the probability of the occurrence of bit 0 or 1 in different IrisCode is equal to Bernoulli trial (Daugman, 2006). Based on Lemma 3.1, this suggests that $\alpha = 0.5$ and the downscaled factor p will be a constant for a selected P (refer to Theorem 3.1). Therefore, the relative similarity of two IrisCodes is able to be preserved with respect to the Jaccard similarity $\frac{|\boldsymbol{X}^{\boldsymbol{P}} \cap \boldsymbol{Y}^{\boldsymbol{P}}|}{|\boldsymbol{X}^{\boldsymbol{P}} \cup \boldsymbol{Y}^{\boldsymbol{P}}|}$ in $\boldsymbol{X}^{\boldsymbol{P}}(\boldsymbol{Y}^{\boldsymbol{P}})$ domain when IFO applied in the IrisCode.

3.4 Alignment-Free IFO

In previous section, we have explored the IFO hashing method and its matching protocol. In fact, it is straightforward to show that the pre-alignment step in IFO hashing is required a significant amount of computational time in order to achieve desirable system performance in term of *EER* The computational time and achievable *EER* results have tabulated in Chapter 4, Table 3 and 4. In order to reduce the computation time consumed, we hereby proposed an solution with alignment-free IFO hashing to address the rotation inconsistency issues in the original IFO hashing. The main concept is to

incorporate the well-known Bloom filter technique that satisfied alignment-free properties (refer Chapter 2, Section 2.2).

We show that with the incorporated Bloom filter each column code word of the original IrisCode is being used to represent a single decimal value. This allows the formation of new unordered set generated from every single column code word from the original IrisCode and hence resolves the rotation inconsistent issues caused by head tile/rotation. The new unordered set generated from each block is just simply the index position of its corresponding bloom filter under binary representation. Afterward, this new binary representation will go through the IFO hashing procedure to generate IFO hashed code for iris template protection.

Noted here, there is not pre-alignment involved but instead the additional transformation step for the bloom filter generation. The matching process follows the conventional IFO matching under zero shift condition.

The proposed alignment-free IFO hashing can be described with the following steps:

1. *Bloom Filter transformation*: Generate a Bloom filter array $\boldsymbol{b} = \{0,1\}^{j\times 2^{l}}$ which form by concatenating j number of Bloom filter, where b_j denoted the jth Bloom filter.

2. *Random Permutation*: Generate a permutation token contains P number of randomly generated permutation vectors. Permute $\boldsymbol{b}$ in column-wise manner P times and yields $\boldsymbol{b}' = \{\boldsymbol{b}'_p | p = 1, \dots, P\}$.
3. *Hadamard Multiplication*: Element-wise multiplies all the permuted bloom filter array to generate a product code denoted as $\boldsymbol{b}^P = \prod_{l=1}^{p} (\boldsymbol{b}'_l)$.

4. *Select First K Elements*: For each row in the product code $\boldsymbol{b}^P$, only select the first K elements and discard others.

5. *Record Index of First Binary Bit '1'*: Among the selected K elements, record the index value, denoted as C_X corresponding to the first occurrence bit '1'.

6. *Modulo Thresholding*: Impose a modulo threshold function with a threshold value τ, $1 \leq \tau < K$ and compute $C'_X = C_X \bmod (K - \tau)$.

When comes to matching, for a query alignment-free IFO hashed codes $\boldsymbol{C'_{Yi}}$, similar to conventional IFO matching, we can simply measure the similarity score by calculating the probability described as Pr ($C'_{Yi} = C'_{Xi}$) for $i = 1,2, \dots, m$. This matching process is essentially the conventional IFO matching without pre-alignment (bits shifting) step. Figure 3.6 shows the example of the procedure in alignment-free IFO hashing with $K = 3, \tau = 0, P = 1$.

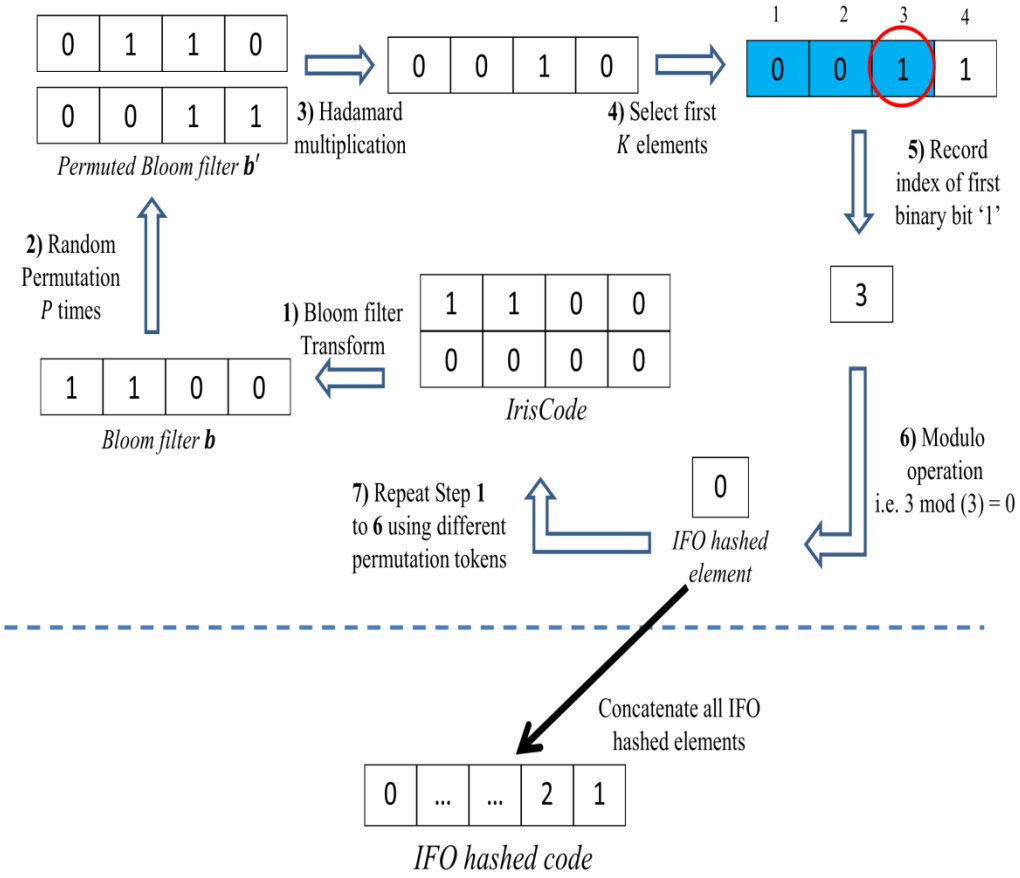

Figure 3.6: Alignment-free IFO hashing

CHAPTER 4

EXPERIMENTS AND ANALYSIS

To evaluate the accuracy performance of the IFO Hashing, CASIA database v3-interval is adopted. This dataset contains 2639 iris images from 396 different classes (eyes). In our experiments, to consistent with the stat-of-the-art's works, only left eye images are considered. Due to the fact that the dataset consists of uneven number of sample for different classes, to standardize the number of matching per classes, we only consider the classes that include at least 7 iris samples and only the first 7 iris samples are selected. Therefore, a smaller subset of the dataset with total of 124 classes, and hence $124 * 7 = 868$ iris images is created and used for IFO hashing. This standardization by using constant number of available iris samples per class is to avoid statistical bias in security analysis, particularly, the revocability analysis that is justified by using a total number of 99x7 newly generated IFO hashed code per users (Section 4.7). Moreover, the new created subset that having fixed number of sample per class is more suitable for future experiments testing that required certain number of training samples e.g. machine learning approaches.

4.1 Test for IrisCode's Performance

We first show the verification rate and its variants, i.e. *EER*, *FAR*, *FRR* and *GAR* (100% - *FRR*) for the comparison/matching between different IrisCode.

For intra-class comparisons, each iris template is matched against the templates generated from other iris samples of the same classes, leading to a total of 2604 genuine comparisons. For inter-class comparisons, every template is matched with all other templates generated from different iris samples of different classes, yielding a total 373674 impostor comparisons. *EER* has been used to evaluate the recognition performance where the *FAR* and *FRR* are equal (Mentioned in Section 1.6). The metrics are computed based on the Hamming distances during matching with ± 16 bits being shifted as shown in Figure 4.1.

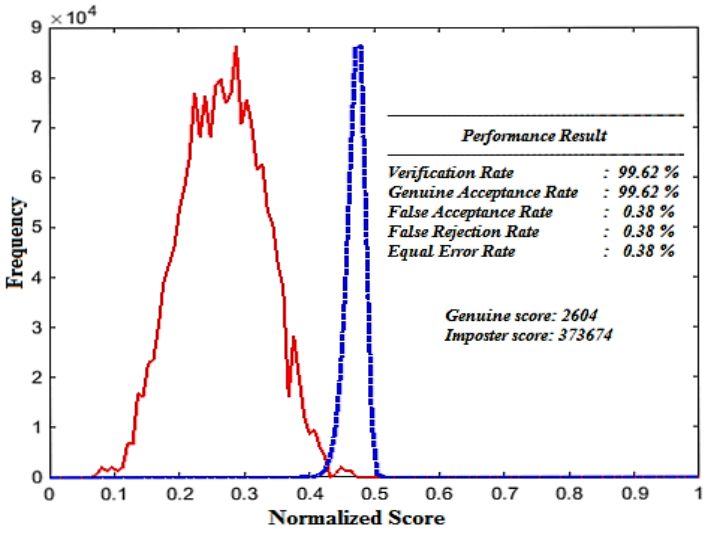

Figure 4.1: Original accuracy performance of IrisCode

4.2 Test for IFO & Alignment-Free IFO Performance

In this section, the IFO template has gone through several testing to evaluate its performance under pre-alignment applied IFO and alignment-free IFO. For performance evaluation between the pre-alignment applied IFO and

alignment-free IFO hashing, the matching result for IFO hashed codes and alignment-free IFO hashed codes are recorded.

For intra-class comparisons, each IFO hashed code is matched against other IFO hashed code generated from other IrisCode of the same eye, thus, a total number of 2604 genuine comparisons is produced.

For inter-class comparisons, every IFO hashed code is matched with all other hashed codes generated from different IrisCode of different eyes, yielding a total 373674 impostor comparisons. All the while, *EER* has been used to evaluate the recognition performance. Table 3 and 4 tabulates the experimental result by using pre-alignment applied IFO hashed codes and alignment-free IFO hashed code in term of *EER*. From Table 3, it is obvious that alignment-free IFO achieved lower *EER* compared to conventional IFO even under high bits shift of ± 8 bits in matching.

Apart from this, the average time (sec) consumed for a single matching process in applied pre-alignment IFO hashing has been recorded and compared to the average time (sec) consumed for one-time Bloom filter generation and alignment-free IFO hashed codes matching process. The result tabulated in Table 4 shows significant drops for alignment-free IFO hashing compared to pre-alignment applied IFO hashing which required a bit shift of ± 8 bits in order to achieve a desirable low *EER* (1%).

Overall, despite the alignment-free IFO hashing process shows higher time consumed when compared with pre-alignment applied IFO hashed codes matching with bit shifting less than ± 6. However, it offers higher recognition performance with *EER* = 0.69 % which is lower than the pre-alignment applied IFO hashing under no shift, shifted ± 2, and ± 4 bits. Nevertheless, the pre-alignment applied IFO hashing only outperform the alignment-free IFO hashing in term of lower *EER* with arrangement of ± 16 bits, but it significantly require more computational time (sec).

	Equal Error Rate (*EER* %)					
	No shift	Shift ± 2 bits	Shift ± 4 bits	Shift ± 6 bits	Shift ± 8 bits	Shift ± 16 bits
IFO	6.72	5.19	3.12	1.34	1.00	0.54
Alignment-free IFO	0.69					

Table 3: Matching result for applied pre-alignment IFO hashed codes ($m = 400$, $K = 400$, $\tau = 0$, $P = 1$) and alignment-free IFO hashed codes ($n = 32,\ l = 10,$ $m = 400$, $K = 400$, $\tau = 0$, $P = 1$)

	Time (sec)					
	No shift	Shift ± 2 bits	Shift ± 4 bits	Shift ± 6 bits	Shift ± 8 bits	Shift ± 8 bits
IFO	0.903	2.968	4.470	8.178	14.265	>25.000
Alignment-free IFO	5.820					

Table 4: Average computation cost (sec) for applied pre-alignment Matching in IFO hashed codes ($m = 400$, $K = 400$, $\tau = 0$, $P = 1$), and alignment-free IFO hashed codes (Bloom filter generation + Matching) ($n = 32,\ l = 10,\ m = 400,$ $K = 400$, $\tau = 0$, $P = 1$)

4.3 Test for IFO Template Performance

In this section, the IFO template has gone through several testing to evaluate its performance. Comprehensive experiments have been carried out to analyze and evaluate the effect of different parameters of $K, m, p,$ and τ in terms of *EER*. Besides, the non-invertibility, revocability, and unlinkability properties are also well-evaluated and justified. Noted here, for optimum performance evaluation, pre-alignment step is applied.

4.3.1 Effect of Parameter m

In this subsection, we examine the relation of the number of the hashing function, m and the verification performance. Experiments have been carried out by increasing m from 10, 20, 30, 40, 50, 100, 200, 300, and 400 while fixing $P = 3$ and $\tau = 0$ as shown in Table 5. As expected, the increment of m gives rise to better Jaccard similarity estimation subjected to different K-window size. With $m = 200$, the *EER* approaches the IrisCode performance. For $m > 200$, the performance level off at *EER* = 0.54%. By using lower K-window, the verification performance deteriorates (*EER* increase). This is subjected to the effect where the algorithm fails to locate the 1st binary bit '1' during IFO hashing. The effect of K will be discussed more in Section 4.3.2.

Equal error rate (*EER*) (%)									
	m=10	***m=20***	***m=30***	***m=40***	***m=50***	***m=100***	***m=200***	***m=300***	***m=400***
K=50	3.37	1.46	1.17	0.99	0.55	0.58	0.54	0.55	0.55
K=100	3.41	1.41	0.98	0.88	0.56	0.76	0.54	0.58	0.54
K=200	3.05	1.29	0.97	0.86	0.62	0.60	0.54	0.54	0.54

Table 5: Performance result using $P = 3, \tau = 0$

4.3.2 Effect of Parameters *K* and *P*

In this section, the effect of K and P with respect to *EER* is investigated. We vary K from 5 to 50 by fixing m = 50 and $\tau = 0$. Same experiments are repeated for $P = 4, 5, 6$. From Table 6, we observe that *EER* drops rapidly with respect to an increment of K. The changes of *EER* is enormous for small K and level off at large K. This is caused by the failure in detecting the first bit '1' in the chosen K-window. This affects the matching efficiency of the proposed scheme and hence degrades the performance. The increase in K eventually includes more binary bits inside the K-window, this offers more bitwise comparison inside the K-window thus the ill-case can be reduced. On the other hand, we notice that large P resulted in high *EER*. This is caused by the less occurrence of bit '1' in the K-window due to the direct consequence of Hadamard multiplication, which is equivalent to bitwise AND operation. Therefore, the increment of K permits more bits to be taken into consideration hence will compensate the effect of Hadamard multiplication. However, the decreasing of *EER* level-off at certain big K and large P requires bigger K to gain lowest *EER*.

Equal error rate (*EER*) (%)										
	K=5	***K=10***	***K=15***	***K=20***	***K=25***	***K=30***	***K=35***	***K=40***	***K=45***	***K=50***
P =4	2.34	0.96	0.97	0.84	0.83	0.76	0.79	0.82	0.70	0.69
P =5	6.04	2.22	1.16	1.00	0.98	1.08	0.91	0.78	0.86	0.78
P =6	23.04	6.02	2.72	1.84	1.71	1.57	1.42	1.39	1.27	1.15

Table 6: Results for *EER* subject to different values of K and P with $m = 50$

4.3.3 Effect of Parameter τ

The parameter τ found in $C'_{Xi} = C_X \, mod \, (K - \tau) \in [0, K - \tau - 1]$ is being introduced as a mean to measure non-invertibility. An experiment has carried out by fixing $P = 3$, and $m = 50$ with increment of τ from 0 to $K - 1$ subject to $K = 50, 100,$ and 200. From Figure 4.2, the increment of τ does not affect EER until $\tau > 0.9K$. This is because large τ shrinks the effective range of C'_{Xi} which also implies information loss. However, this trait is favorable as the loss promotes the non-invertibility capability of the IFO hashing. Despite large τ deteriorates performance. This issue is indeed controllable as long as $\tau < 0.9K$. Nevertheless, different range of τ has different effects on the non-invertibility. As τ increase, it will intensify the effect of modulo thresholding and resulted in more modulo mapping for $C_X \geq K - \tau$. Simply speaking, increasing τ actually enhancing the non-invertibility of the IFO hashed codes at the same time. More details about the effect of τ on non-invertibility will be discussed in Section 4.5.

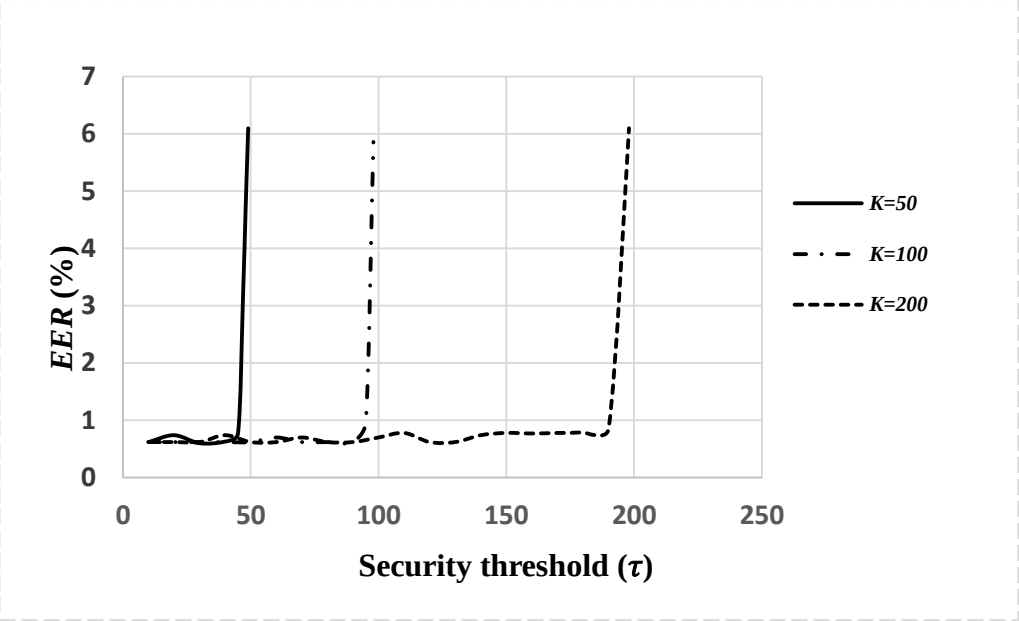

Figure 4.2: Equal Error Rate versus Security Threshold

4.4 Comparison with Other Schemes

In the iris's template protection literature, the number of iris samples used in the experiments is varying. Even in the CASIA v3 database, the number of iris images of each subject is not the same. In our experiments, as discussed in the beginning of this section, we use seven iris samples since most of the subjects in CASIA v3 database contain at least seven iris samples. Since there is no standard experimental protocol used in existing iris template protection schemes, it is difficult to compare them in a fair manner. However, it is worth to show the performance result of IFO hashing with state of the arts for benchmarking purpose.

Proposed method	No. Iris Images Used	Without Template Protection EER (%))	With Template Protection EER (%))	Performance Deteriorate EER (%))
IFO hashing	868 (left eye)	0.38	0.54	0.16
Block Remapping (Hämmerle –Uhl et al., 2009)	2653	1.10	1.30	0.20
Bio-Encoding (Ouda et al., 2010)	740	6.02	6.27	0.25
Adaptive Bloom filter (Rathgeb et al., 2013)	1332 (left eye)	1.19	1.14	-
Bin-Combo (Zuo et al., 2008)	1332 (left eye)	0.81	4.41	3.60

Table 7: Summarized results in EER of IFO hashing with the state of the arts (CASIA v3-Interval database).

For each technique used, we have highlighted the number of iris images used in CASIA v3 database, their lowest EER before and after their respective protection schemes are applied as well as the change of EER before and after their respective protection schemes are applied. The summarized results are shown in Table 7.

4.5 Non-Invertibility Analysis

In this section, the non-invertible analysis of IFO hashed code is presented. In our analysis, an adversary is assumed to have acquired all the information regarding IFO hashing algorithm, parameters ($K, P, \tau,$ and m), stolen hashed code(s) and permutation token. In this case, the relationship between the K-window bits and the IrisCode could be explored thoroughly by the adversary. Four

major attacks have been focused namely single hash attack, multi-hash attack, attack via record multiplicity (ARM) and pre-image attack. All the while we have refers to $\tau \geq \frac{K}{2}$ in order for IFO hashing to withstand all the mentioned privacy and security attacks.

Since the permutation of IrisCode is done in the column-wise manner as discussed in Section 3.2, the inversion attack complexity is the same for every row. Hence our analysis can be simplified into *row-wise* IrisCode regeneration.

4.5.1 Single Hash Attack (SHA)

This section discusses the possibility of IrisCode restoration from a single IFO hashed code entry, C'_{Xi} in which we called Single Hash Attack (SHA). Recall IFO hashing function, $H(\boldsymbol{X}) = \{h_i(\boldsymbol{X}) | i = 1, \ldots, m\}$ exploits m independent hash functions for hashing. In other words, each hashed code entry C'_{Xi} is generated by each hash function $h(.)$, which is subjected to brute force attack. Hence, SHA is indeed a special case of brute force attack of IFO. Unlike conventional brute force attack that applies to the entire template, SHA only targets to a single entry of hashed code C'_{Xi}.

In order to invert the IFO hashed code, by knowing the hashed code entry C'_{Xi}, the adversary has to traverse through a path to first learn C_{Xi} from C'_{Xi}, then *K*-window bits and finally IrisCode restoration. For modulo thresholding, τ can be set up to $0.9K$ without incurring performance lost as discussed in Section 4.3.3.

When $\tau \geq \frac{K}{2}$, the reverse mapping of C'_{Xi} to C_{Xi} is of one-to-many relation. This would hinder the K-window bits reconstruction from guessing the actual C_{Xi}. The one-to-many relation is indeed intensified when we set $\tau \geq \frac{K}{2}$.

For example, let $K = 6$, $\tau = \frac{K}{2}$, hence $C'_{Xi} = C_{Xi}\ mod\ (3)$ where $C'_{Xi} \in [0, 2]$. Different C_{Xi} may map to the same C'_{Xi} thanks to the one-to-many relation of C'_{Xi} and C_{Xi}. For instance, C'_{Xi} is identical for $C_{Xi} = 1$ and $C_{Xi} = 4$ since $4\ mod\ (3) = 1$, thus an adversary needs to guess for the exact C_{Xi} among all the possible modulo mappings. In order to reconstruct the K-window bits, the exact C_{Xi} value needs to be known by the adversary since it represents the first bit '1' of the IrisCode.

To measure the SHA complexity for K-window bits reconstruction, the number of possible modulo mappings, say *r* needed to be estimated first. Followed the previous example, we noticed that $r = 3$ for all C'_{Xi} values including $C'_{Xi} = 0$. However, since '0' carries no information, we can always ignore it. Finally, without taking into consideration of '0', $r = 2$ for $C'_{Xi} \in [1,\ K - \tau - 1]$. We can establish a relation of *r*, *K* and τ in which $K - \tau = \frac{K}{r}$. With algebra manipulation, we obtain:

$$r(1 - \frac{\tau}{K}) = 1 \tag{4.1}$$

This implies that for $\tau < 0.9K$, $C'_{Xi} \in [0,\ K - \tau - 1]$ contains $r \in [2, 9]$ possible mappings. This allows us to estimate the minimum number of guessing as $n_1 r^m$ is required to fully recover C_{Xi} from C'_{Xi}.

Once all the C_{Xi} values are correctly guessed, the K-window bits could be recovered with another guessing, in which the complexity be $n_1 2^{K - C_{Xi}}$. Recall that each K-window bits are the output of the multiplication of tokenised randomly permuted P bits in the IrisCode. When the permutation token is compromised, the locations of all P bits could be disclosed. These locations also can be seen as the indices of the IrisCode. For each single bit in the K-window, the corresponding indices values of the P bits in the IrisCode is given as $\boldsymbol{\varphi}_d \in \{1, n_2\}^P$, where $d \in [1, K]$ represents the bit index in the K-window.

When the K-window bit is '1', the adversary can infer that the corresponding P bits is also 1 based on the indices given in $\boldsymbol{\varphi}$. This is because the only way for K-window bit to be '1' is when all the P bits given $\boldsymbol{\varphi}$ are 1. On the other hand, when the K-window bit is '0', due to the P random permuted Hadamard multiplication of '0' and '1', it allows $2^P - 1$ combinations for each K-window bit to be '0'. For example when $P = 2$, there exist $2^2 - 1$=3 ways for the product of two binary bits to be '0', as (0, 0), (0, 1), and (1, 0). In this case, the adversary will have to search all the possible combinations which is harder compared to the case when the K-window bit is '1'. Figure 4.3 illustrates the IrisCode recovery in a stolen token case for $K = 3, P = 3$.

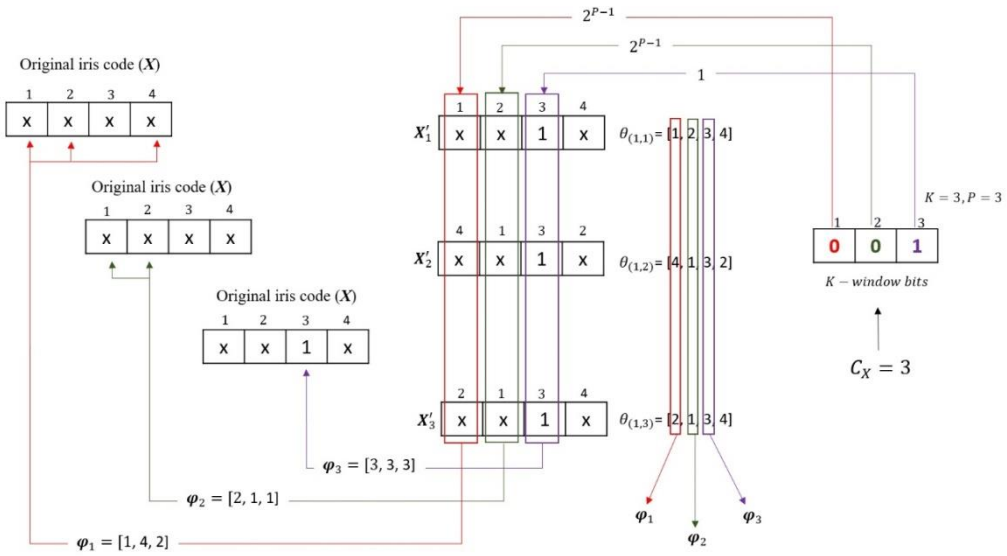

Figure 4.3: IrisCode recovery process using permutation token (best view in color)

Based on Figure 4.3, each single binary bit in K-window is constructed by multiplying three bits $(P = 3)$ from the randomly permuted IrisCodes $\boldsymbol{X}'_1, \boldsymbol{X}'_2$, and $\boldsymbol{X}'_3$. The corresponding P bits are framed with different colors for different K-window bits (red, green, and purple for 1st, 2nd, and 3rd K-window bit, respectively). Besides that, the corresponding indices of the P bits are also framed with distinct colors (red, green, and purple) for different K-window bits based on the permutation token θ. In this example, all the unknown bits are marked with 'x'. The only way that allows an adversary to re-generate maximal P number of 1s in the IrisCode is when the K-window bit equal to one. The P number of 1s refers to the number of distinct indices given by $\boldsymbol{\varphi}_d$. As shown in Figure 4.3, for $d = 1$, which refers to the 1st K-window bit, $\boldsymbol{\varphi}_1 = \{1, 4, 2\}$ consists of three distinct

indices values. This indicates that the 3 bits in the IrisCode with indices location of '1', '4' and '2' are multiplied together to form the 1st window bit. For 2nd K-window bit ($d = 2$), $\boldsymbol{\varphi}_2 = \{2, 1, 1\}$, which is a degenerate case, because of the index of '1' has repeated two times. To be exact, we call this as *2nd degree of degeneracy* due to two repeated elements. With a higher degree of degeneracy, the number of restored bit '1' would be decreased. This can be explained by further looking into the 3rd K-window bit ($d = 3$). In this case, all the indices values are repeated as '3', hence, it is in Pth degree of degeneracy, while $P = 3$. The number of bit '1' that can be restored is only one, i.e. the one that is located in the index of '3' of the IrisCode.

The degeneracy is due to the collisions between each permutation token. For upper bound, we always assume $\boldsymbol{\varphi}_d$ consist of P distinct indices values in which we denote as $|\boldsymbol{\varphi}_d * | = P$, while $|.|$ is the cardinality, and $\boldsymbol{\varphi}_d *$ represents the distinct elements in $\boldsymbol{\varphi}_d$. Based on the example shown in Figure 4.3, the adversary can re-generate maximally P number of '1' in the IrisCode from $\{\boldsymbol{\varphi}_d | d = 1,2, \dots, K\}$. For the adversary to regenerate the remaining unknown bits in the IrisCode $\boldsymbol{X}$, The upper bound SHA complexity can be described as a function of $, n_1, n_2\ |{\varphi_d}^*|$, K, C_{Xi}, r and m as SHA$(n, n_1, n_2\ |{\varphi_d}^*|, K, C_{Xi}, r, m)$ in the following equation:

$$\text{SHA}(n, n_1, n_2\ |{\varphi_d}^*|, K, C_{Xi}, r, m) = n_1(2^{n_2 - n|{\varphi_d}^*|} . 2^{K - C_{Xi}} . r^m) \qquad (4.2)$$

Where n represents the number of '1' in the K-window, while $n|\varphi_d{}^*|$ indicates the total number of '1' of the original IrisCode can be regenerated. The second term $n_1 2^{K-C_{Xi}}$ refers to the number of guessing for K-window bits reconstruction, while third term $n_1 r^m$ be the number of guessing from C'_{Xi} to C_{Xi}.

Based on the statistical theory, if two different IrisCodes are totally independent, then the matching of two IrisCodes can be regarded as an independent test with 50 % chance to be matched or unmatched (Daugman, 2006). Due to the fact that n depends on the total numbers of '1' in the IrisCode, and different IrisCodes comprise a different number of '1', it is difficult to infer an absolute value of n. An assumption can be made that allows *n* to be estimated is as follows: *For each IrisCode be independent, all of the bit values are equally likely*. From this assumption, each randomly permuted IrisCode can be considered as a new independent instance, thus, the expected number of '1' in the K-window after P Hadamard multiplication of the permuted IrisCode is:

$$\mathbb{E}(n) = K\left(\frac{1}{2}\right)^P \tag{4.3}$$

In order to justify that our assumption is valid, the average number of remaining '1' in the P order Hadamard multiplication have been computed experimentally for all the iris images in CASIA v3-interval database. The number of '1' in the product code has also been calculated by substituting $K = n_2$ into Eq. (4.3) for $P = 1, \ldots, 5$. Since when $K = n_2$ (size of IrisCode), K-window is just

equal to the product code. Finally, a graph of the remaining bits '1' in the K-window (n) versus P is as shown in Figure 4.4.

Our assumption is supported by the empirical result in which the experimental n are almost identical to the theoretical values calculated with Eq. (4.3). Hence, we give the following proposition to generally describe the SHA complexity under IFO hashing.

Proposition 4.1: *Under non-degenerate case* $|\boldsymbol{\varphi}_d^*| = P$, *given* $\mathbb{E}(n) = K\left(\frac{1}{2}\right)^P$, *for any* $C_{\boldsymbol{X}i} = K$, *the expected SHA complexity to regenerate* $\boldsymbol{X}$ *from* $C_{\boldsymbol{X}i}$ *is lower bounded as* $\mathbb{E}\big(SHA(n_1, n_2\ P, K, r, 1)\big) \geq n_1 r.\, 2^{n_2 - [KP\left(\frac{1}{2}\right)^P)]}$.

Proof: By substituting $\mathbb{E}(n)$ from Eq. (4.3) into Eq. (4.2), the expected SHA complexity for IrisCode restoration now can be estimated as follow:

$$\text{SHA}(n_1, n_2\ |\varphi_d{}^*|, K, C_{Xi}, r, m, P) = 2^{-[\mathbb{E}(n)(|\boldsymbol{\varphi}_d^*|)]}\, 2^{n_2}.\, 2^{K - C_{Xi}}.\, r^m.\, n_1. \quad (4.4)$$

To estimate the lower bound for SHA, the term $2^{-[\mathbb{E}(n)(|\boldsymbol{\varphi}_d^*|)]}$ can be written as a convex function $g(\mathbb{E}(n))$. Therefore, by using *Jensen's inequality* we have

$$\mathbb{E}\big(2^{-n(|\boldsymbol{\varphi}_d^*|)}\big) \geq 2^{-[\mathbb{E}(n)(|\boldsymbol{\varphi}_d^*|)]}. \quad (4.5)$$

By taking max $(C_{Xi}) = K$, the maximum hashed value in hashed code, $m = 1$ (a single C'_{Xi} due to SHA) and with $|\boldsymbol{\varphi}_d^*| = P$ for the non-degenerate case, together with Eq. (4.5), we able to describe the expectation SHA complexity as $\mathbb{E}\big(\mathrm{SHA}(n_1, n_2\ P, K, r, 1)\big) = n_1 r.\, 2^{n_2}\mathbb{E}(2^{-nP})$. Hence, Eq. (4.4) can be simplified as following inequality and prove the proposition:

$$\mathbb{E}\big(\mathrm{SHA}(n_1, n_2\ P, K, r, 1)\big) \geq n_1 r.\, 2^{n_2 - [KP\left(\frac{1}{2}\right)^P)]}. \tag{4.6}$$

▮

In an ideal case, as shown in Eq. (4.6), for large P, the term $n_2 - KP\left(\frac{1}{2}\right)^P \approx n_2$ and this suggests that $\mathbb{E}\big(\mathrm{SHA}(n_1, n_2\ P, K, r, 1)\big) \geq n_1 r.\, 2^{n_2}$. This means the adversary is expected require to try at least 2^{512} times for IrisCode restoration under SHA.

The result shows that despite in the scenario where all information is revealed, it is computationally hard for an adversary to restore the IrisCode from the stolen hashed code. Figure 4.5 shows the entire inversion process and its calculated minimum number of guessing for each step.

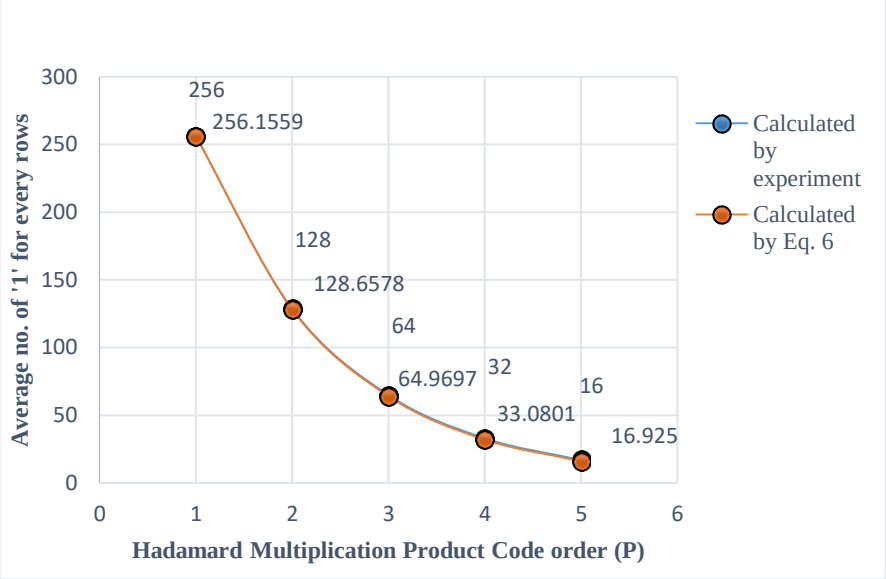

Figure 4.4: Graph of the mean remaining bits with value '1' in the K-window (n) versus P

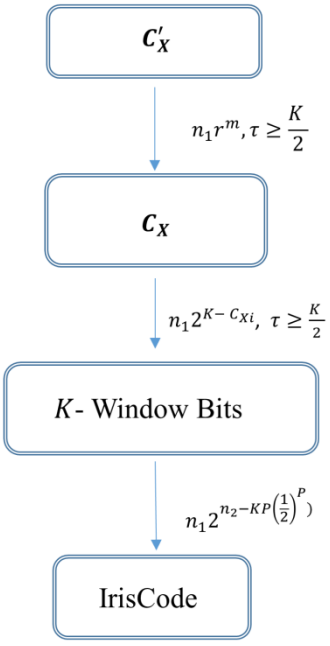

Figure 4.5: Estimated SHA complexity for IrisCode restoration.

4.5.2 Multi-Hash Attack (MHA)

In this section, a non-uniform attack under our non-invertible analysis, namely *Multi-Hash Attack* is analysed. Unlike SHA that solely based on a single C'_{Xi}, it is possible for an adversary to launch the attack with multiple hashed code elements in the hashed code [C'_{X1}, C'_{X2}, … , C'_{Xm}].

To calculate the complexity of MHA, we follow the previous verified assumption such that the number of '0' and '1' in the IrisCode is equally likely, hence the IrisCode can be fully restored if approximately half of the bit '1' information (i.e. $\frac{n_2}{2}$) in each row of IrisCode is being known by the adversary. This is possible by first generating a sparse code which is the same size as IrisCode and simply guessing the remaining unknown bits to be all '0' or '1'. By doing so, it allows the adversary to gain knowledge about 50% of the bits in the IrisCode. As we known each hash entry C'_{Xi} contained the binary information of the first occurrence bit '1', and this allows maximally P number of binary information of the original IrisCode to be explored by the adversary when the permutation token is being revealed (refer Section 4.5.1). Thus, the minimum number of C_{Xi} values required to fully regenerate the entire original IrisCode can be calculated by simply dividing $\frac{n_2}{2}$ by P, ie. $\frac{n_2}{2P}$.

However, there are r possible mappings for each C'_{Xi} to be C_{Xi}, hence, the adversary will need to try different mappings of C_{Xi} in brute force searching, which lead to a MHA complexity with minimum number of C_{Xi} described as:

$$\text{MHA}(n_1, n_2, r, P) \geq n_1 r^{\frac{n_2}{2P}} \tag{4.7}$$

The complexity of MHA has been greatly reduced compared with SHA, ie. $n_1 r^m$. This implies MHA only required a single trial to regenerate the IrisCode once all the hashed value C_{Xi} are fully recovered from C'_{Xi}.

In order to prevent MHA, one can reduce m such that $m < \frac{n_2}{2P}$. In our experiment, for $n_2 = 512, P = 3$, Then, $\frac{n_2}{2P} = 85$. If we set $m < 85$ for IFO hashed code, MHA can be avoided due to insufficient information of C_{Xi}, for $i \in [1, m]$. In this context, MHA will never succeed if $m < \frac{n_2}{2P}$ is set.

4.5.3 Attack via Record Multiplicity (ARM)

ARM refers to a privacy attack, which utilized multiple compromised protected templates with and without the associated information, i.e. helper data, parameters etc. to reconstruct the original biometric template (Scheirer et al., 2007). In this context, ARM can be regarded as a generalization of MHA by increasing the number of available C_{Xi} as discussed in previous Section 4.5.2. As discussed in MHA, the number of available C_{Xi} is limited to be less than $\frac{n_2}{2P}$, i.e. only in one IFO hashed code. For ARM attack, we need to take into account where there is possible for an adversary to gain extra information through multiple compromised IFO hashed codes.

For example, let the input be a single row of IrisCode, $\boldsymbol{X} \in \{0,1\}^{1\times N}$ and $\boldsymbol{X}^P$ be the Hadamard multiplication product code. When $K = N$, each of the elements in the K-window is denoted as X_k^P, for $k \in [1, N]$. Since $K = N$, the K-window is equivalent to the product code. Hereby, X_k^P can be seen as a product of P elements chosen randomly (based on permutation token) from X_1, X_2, $X_3, \dots, \dots, X_N$ which refer to the 1st, 2nd, 3rd,…, N^{th} elements of $\boldsymbol{X}$, respectively. The X_k^P also can be read as the kth bits in the K-window. Let the permutation tokens be $[1, 2, 4, 3]$ and $[2, 4, 3, 1]$, while $P = 2,\ N = 4$. Meanwhile, the permutation token is known to the adversary, the corresponding indices values of the P bits in the IrisCode are $\boldsymbol{\varphi}_1 = \{1, 2\}$, $\boldsymbol{\varphi}_2 = \{2, 4\}$, $\boldsymbol{\varphi}_3 = \{4, 3\}$, $\boldsymbol{\varphi}_4 = \{3,1\}$. For each X_k^P, four equations can be established based on each $\boldsymbol{\varphi}$ as follows:

$$
\begin{aligned}
X_1^2 &= X_1 X_2 \\
X_2^2 &= X_2 X_4 \\
X_3^2 &= X_4 X_3 \\
X_4^2 &= X_3 X_1
\end{aligned}
\tag{4.8}
$$

From Eq. (4.8), it is obvious that for $X_1^2 = 1$, X_1 and X_2 must both also be '1'. This indeed signals us that the bits '1' in IrisCode appear at the 1st and 2nd position. The same goes for X_2^2, X_3^2, and X_4^2. By doing so, the recovery of bits '1' in the IrisCode is now depending on the information that can be obtained when $X_k^P = 1$, which is essentially equivalent to the number of bits '1' in the K-window as discussed in Section 4.5.1.

Normally, the occurrence of $X_k^P = 1$ can be suppressed by increasing P (decreased number of bit '1' inside K-window via Hadamard multiplication). By doing so, limited information can be acquired by the adversary for IrisCode restoration. However, since same IrisCode may be use for different IFO hashing for different application purpose, a new set of equations can form from different IFO hashed codes once compromised. For each IFO hashed code, their K-window bits are different due to permutation token. Yet, the permutation token only provides the relation of the K-window bits and the IrisCode. As such, ARM only succeeds when the K-window bits information of each IFO hashed codes are exhaustively explored. The complexity of ARM can be measured with the complexity of MHA as $n_1 r^{\frac{n_2}{2P}}$, since the adversary still needs to try different possible mappings for C'_{Xi} to C_{Xi}.

As an illustration, based on Eq. (4.8), say $X_1^2 = 0$, $X_2^2 = 0$, $X_3^2 = 1$, and $X_4^2 = 0$, we have $C_X = 3$ since the first '1' occurs at 3rd location. We can further assume the output of modulo thresholding to be $C'_X = C_X \bmod (2)$. For $C_X = 3$, then $C'_X = 1$. When the adversary constructs $X_1^2 = X_1 X_2$, this will result to wrong reconstruction due to the shifted output C_X from 3 to 1. The actual value of $X_1^2 = 0$ will be restored by the adversary as $X_1^2 = 1$, but $X_1 X_2 \neq 1$. This yields an invalid equation. Hence, the adversary has to search all the possible mappings for C'_{Xi} to C_{Xi} for IrisCode restoration, which is estimated to be similar as $\mathrm{MHA}(n_1, n_2, r, P)$.

4.6 Potential Security Attack

In this section, we will discuss a security attack against IFO hashing namely pre-image attack (PIA). Unlike previous attacks mentioned in non-invertibility analysis which meant for privacy by preventing an adversary to fully recover the original IrisCode, the section exploits the potential security attacks that mean to get access to the system without the needs of 100% recovery of the original IrisCode. We here provide two potential security attacks on IFO namely pre-image attack, and false accept attack.

4.6.1 Pre-image Attack (PIA)

In this section, we will discuss a security attack against IFO hashing namely pre-image attack (PIA). Unlike previous attacks mentioned in non-invertibility analysis which meant for privacy by preventing an adversary to fully recover the original IrisCode, PIA is meant to access biometric systems illegally by exploiting the close approximation of the original biometric data (also known as pre-image) from the protected biometric template with lower attack complexity (Nandakumar et al., 2015).

In order to access the biometric systems, the biometric input does not necessary require to be 100% similar with the enrolled template. For example, Jernish et al., (2011) managed to launch PIA where only 60% of the IrisCode information is exploited. Besides, Bringer et al., (2015) were also able to launch

PIA on Bloom filter based protected iris template (Rathgeb et al., 2013) with a block width of 16 and 32. Besides that, Nagar et al, (2010) demonstrated the technique to learn the pre-image from BioHash based protected biometric template.

For perfect restoration of IrisCode under assumption that the occurance of bit '0' and '1' are equal likely, the attack complexity discussed in ARM and MHA is estimated as $n_1 r^{\frac{n_2}{2P}}$. This complexity is computed based on the advance knowledge of 50% of the IrisCode information. To be more general, $n_1 r^{\frac{n_2}{2P}}$ can also be written as:

$$\text{PIA}(n_1, n_2, r, P, t) = n_1 r^{\frac{n_2 t}{P}}, \qquad \text{for } 0 \leq t \leq 0.5 \tag{4.9}$$

Here, let t be a *regeneration threshold* which determines the attack complexity to restore $2t \times 100\%$ of the IrisCode with length n_2. As we can see, the complexity of ARM and MHA is being reduced to $n_1 r^{\frac{n_2}{2P}}$ when $t = 0.5$, which indicates perfect restoration of IrisCode. Therefore, we known that $\text{PIA}(n_1, n_2, r, P, t) = (n_1 r^{\frac{n_2}{2P}})^{2t} = (\text{MHA})^{2t}$. Straight forwardly, from Eq. (4.7) we got:

$$\text{PIA}(\text{MHA}, t) \geq n_1 r^{\frac{n_2 t}{P}}, \qquad \text{for } 0 \leq t \leq 0.5 \tag{4.10}$$

Now, we show how IFO withstands PIA based on different t. For example, we assume the adversary would be able to access to the system only when he/she

successfully restore at least $2(0.25) \times 100\% = 50\%$ of the IrisCode where $t = 0.25$. As compared with ARM and MHA, the attack complexity of PIA is greatly reduced to $\geq n_1 r^{\frac{n_2}{4P}}$ which indicates lesser effort is needed for inversion. Nevertheless, one can still increase r to counter PIA. For this instance, with $n_1 = 20, n_2 = 512, P = 3$ (optimal experiment setting), the PIA complexity is $\geq 20r^{43}$. The highest attack complexity without accuracy performance degradation can be attained at $20(9)^{43} = 2^{136}$ for $r = 9$ since $r \in [2, 9]$ as shown in Section 4.5.1.

4.6.2 False Accept Attack (FAA)

This section discuss another potential security attack named as false-accept attack (FAA). In practice, the matching between different IFO hashed code merely calculating the collision probability for two hashed elements to be the same. Refers to Eq. (3.10), we have discussed that $\Pr[C_{Xi} = C_{Yi}] = \frac{\mathbb{E}(z)}{m} = P_S$ comes with expected number of collision denoted as $\mathbb{E}(z)$ that merely depends on $P_S = S(\boldsymbol{X}, \boldsymbol{Y}) = p\frac{|\boldsymbol{X} \cap \boldsymbol{Y}|}{|\boldsymbol{X} \cup \boldsymbol{Y}|}$.

Hereby, we let $\delta \in [0,1]$ and then δm denoted a fraction of hashed elements in a single row of IFO hashed code with length m. With certain arrangement of δ, i.e. one is required to have at least $z \geq \delta m$ number of collision in order to get access into the system, then, given a fixed value of m, we are able to measure the genuine accept probability (*GAP*) for a genuine user as:

$$GAP = \Pr(z \geq \delta m),\ \delta \in [0,1]. \tag{4.11}$$

Since $\mathbb{E}(z) \sim \text{Bin}(m, P_S)$, the false rejection probability (*FRP*) for genuine user indeed can be easily calculated, using the following formula:

$$FRP = 1 - GAP = 1 - \Pr(z \geq \delta m). \tag{4.12}$$

We hereby give an example to analyse the FRP of an genuine user. Suppose one with arrangement $\delta = \frac{3}{4}$ and $m = 1024$ and a formerly enrolled IFO hashed code generated from Iriscode $\boldsymbol{X}$. Given an genuine user possessing a similar IrisCode $\boldsymbol{X}'$ with $S(\boldsymbol{X}, \boldsymbol{X}') = 0.9$ i.e. 90% similar, based on Eq. (4.11), the *GAP* is calculated to be $GAP_{X'} = \Pr\left(z \geq \frac{3(1024)}{4}\right) = 1$, and hence $FRP_{X'} = 0$.

On the other hand, we can calculate the false accept probability *FAP* for a given adversary trying to get access through the system. For example, we hereby assume an adversary possessing another IrisCode $\boldsymbol{Y}$ with $S(\boldsymbol{X}, \boldsymbol{Y}) = 0.3$ i.e. only 30% similar. In this case, because we are now referring to an adversary, the *GAP* for genuine now becomes *FAP* for adversary. Therefore, *FAP* for the adversary indeed can be calculated based on Eq. (4.11). Thus, the *FAP* for an adversary is denoted as $FAP_Y = \Pr\left(z \geq \frac{3(1024)}{4}\right) = 2.3 \times 10^{-87}$. This implies the adversary

certainly fail to get access into the system with only 30% similarity of IrisCode under above mentioned parameter arrangement.

In practice, under the case when the adversary possessing $\boldsymbol{Y}$ with $S(\boldsymbol{X}, \boldsymbol{Y}) = 0.5$, this case is likely to occur due to the similarity between different IrisCode is expected to be half (Daugman 2004). We thereby can easily show that the false acceptance probability $FAP_Y = 9.8 \times 10^{-12}$ is equivalent to only 33 bits security.

However, one can still increase the false acceptance security by increasing the number of hashed elements (m). For another instance, suppose the arrangement now remains $\delta = \frac{3}{4}$ and increase $m = 2048$, under same analysis apply on the same adversary ($S(\boldsymbol{X}, \boldsymbol{Y}) = 0.5$), it shows that the achievable false acceptance probability increased up to $FAP_Y = 1.36 \times 10^{-118}$ which is > 256 bits.

4.7 Revocability Analysis

For revocability, the ARM analysis in Section 4.5.3 shows that it is computationally infeasible to derive the original counterpart from the IFO hashed code. This means multiple IFO hashed codes are able to generate from a single IrisCode hence revocability can be achieved.

To further evaluate the revocability of IFO hashed codes, 100 hashed codes derived from a single IrisCode have been generated with 100 random permutation tokens. Then, the first hashed code is matched with other 99 hashed codes generated from the same IrisCode. The entire process is repeated for different users and generated $99 \times 7 \times 124 = 85932$ pseudo-imposter scores. For fair revocability analysis, every user must contribute constant amount of pseudo-imposter scores (e.g. constant number of new hashed code matching) to avoid statistical bias. In this experiment, our major focus is to observe the score distributions of the imposter and pseudo-imposter. Note that, the imposter and pseudo-imposter matchings are identically conducted under no shift condition (without pre-alignment) to reduce the computational burden. The genuine, imposter, and pseudo-imposter distributions are generated with $P = 3, K = 50, m = 50$ and $\tau = 0$ as shown in Figure 4.6.

From Figure 4.6, a large degree of overlapping is observed between the imposter and pseudo-imposter distributions. This implies that the refreshed hashed codes are sufficiently distinctive albeit they are generated from the same IrisCode. Indeed, the new hashed code acts as an 'imposter' to the old hashed code since they are uncorrelated. This verifies that IFO Hashing satisfies the revocability requirement whereby new hashed code is able to replace the old one with different permutation tokens.

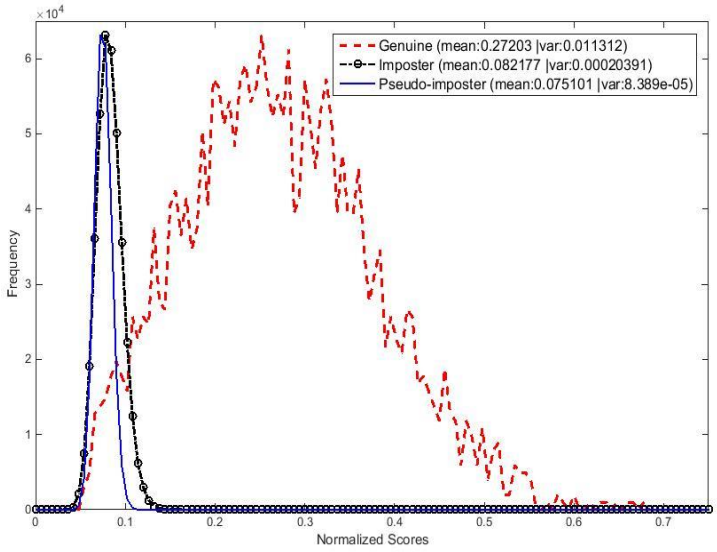

Figure 4.6: The genuine, imposter and pseudo-imposter distributions; large overlap between imposter and genuine due to no pre-alignment.

4.8 Unlinkability Analysis

This section covers the brief unlinkability analysis of IFO hashed codes. The generation of IFO hashed code majorly depends to the permutation of the IrisCode. Each hash entry C'_{Xi} is generated from the Hadamard multiplication of P permuted IrisCode. If all the permuted IrisCode is independent, C'_{Xi} would be independent as well. This implies there has no link in between each hash entry C'_{Xi} . Thus, the IFO hashed codes composed by m hash entry will be independent and then unlinkability able to satisfy.

Hereby, we have followed Daugman's independent test reported in Daugman (2004, 2006) to carry out our unlinkability test for the permuted IrisCode. Daugman reported that due to the phase encoding of IrisCode is equally

likely, two different IrisCodes are uncorrelated and their expected Hamming distance is 0.5 ideally. Daugman performed around 9.1 million empirical comparisons between different pairs of IrisCodes and produced a binomial alike distribution curve with mean $= 0.499$, and standard deviation $= 0.0317$. This further suggests that it is improbable for two different IrisCode to disagree less than $\frac{1}{3}$ of their phase code. The standard deviation of 0.0317 indicates very small internal correlations for any given IrisCode. The distribution curve will be very much sharper (smaller standard deviation) if all the bits in the IrisCode are independent.

With Daugman's independent test in mind, we use the same procedure to verify whether the permuted IrisCodes are independent. In our experiment, we first generate 100 random permutation vectors and each IrisCode has been randomly permuted. Each permuted IrisCode is then matched with the remaining 99 permuted IrisCode resulting a total 4950 hamming distance scores. To avoid statistical bias, this process is repeated for each different iris images in the database resulted in a total of 4296600 hamming distance scores.

Figure 4.7 shows the outcomes of our experiment. As expected we obtained a distribution with mean $= 0.49725$, and standard deviation $= 0.0073$. The standard deviation is smaller and the distribution is sharper as compared to Daugman's experiment. This is because each permuted IrisCode is totally random and independent, their internal correlations are expected to be lesser. From this

experiment, we can say that the permuted IrisCode are independent and satisfies the unlinkability property.

Besides, merely tested with the independency of the permuted IrisCode is not enough, we also introduced the pseudo-genuine score with another experiment to further evaluate the unlinkability of IFO hashed codes. This pseudo-genuine score refers to the matching scores between the IFO hash codes generated from different IrisCodes of the same individual by using different permutation tokens. Like in the genuine matching, the pseudo-genuine scores contain 2667 matching scores. Recall that, the pseudo-imposter scores (Section 4.7) is the matching score between the IFO hashed codes generated from each IrisCode using different permutation token. In this context, when the pseudo imposter and pseudo-genuine distribution are overlapped, it means that we cannot differentiate the IFO hash codes generated from the same user or from the others. On the other hand, if both distributions are separated far apart, this will allows us to differentiate the IFO hash code easily whether it is generated from the same individual or not. The difficulty in differentiating the IFO hash codes has contributed to the unlinkability property. Figure 4.8 shows the pseudo-imposter and pseudo-genuine distribution plot. From Figure 4.8, the pseudo-imposter and pseudo-genuine distribution are largely overlapped. This further supports our claim in which the IFO hashed codes satisfied unlinkability property.

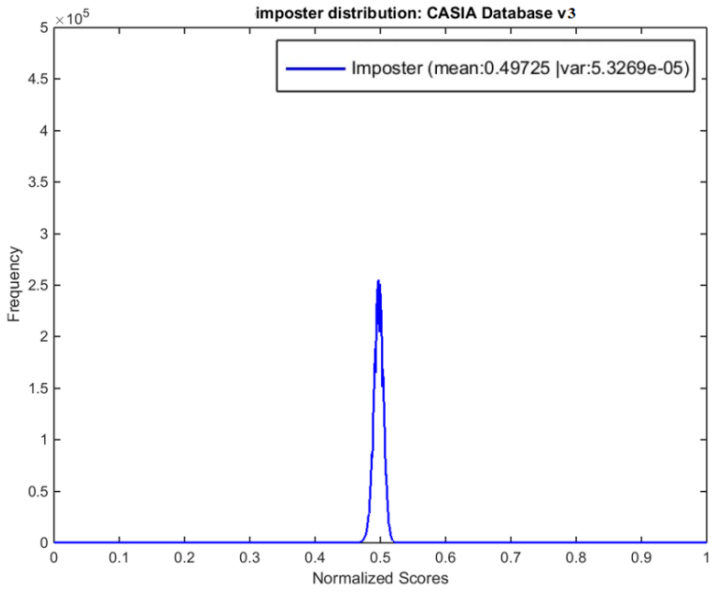

Figure 4.7: Hamming score distribution of randomly permuted IrisCodes

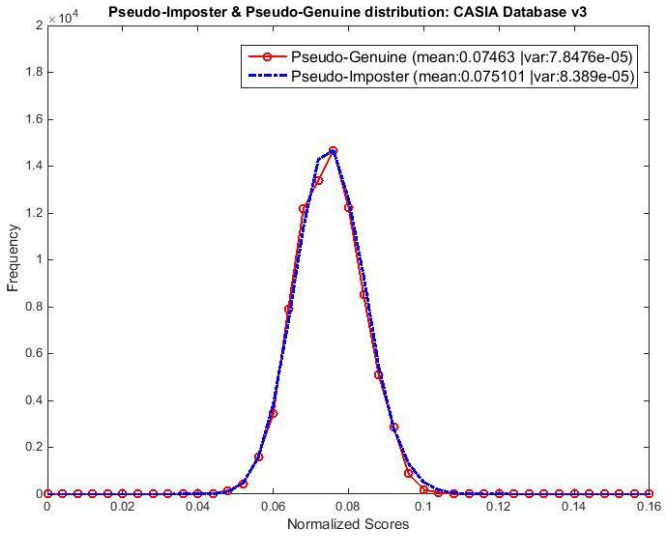

Figure 4.8: Pseudo-Imposter & Pseudo-Genuine distribution: CASIA Database v3

CHAPTER 5

CONCLUSION AND FUTURE WORKS

5.1 Conclusion

In this dissertation, we proposed a cancelable iris scheme, known as IFO hashing which is inspired from the Min-hashing. Two new mechanisms namely Hadamard multiplication and modulo thresholding function are introduced to further enhance the scheme. Several comprehensive experimental evaluations vindicate the accuracy performance of the proposed scheme is preserved with respect to its original counterpart. With rigorous analysis that backed by empirical data, we showed that IFO hashing scheme survives several major security and privacy attacks such as single hash attack, multi-hash attack, attack via record multiplicity and pre-image attack. We also demonstrated that the scheme satisfies the revocability and unlikability requirements and the users are not required to keep their permutation token in secret. The IFO hashed code size can be estimated as $\log_2 K$ bit for each hashed value. With small K, one can save more space for storage but with lower security payoff. Nevertheless, IFO enjoys fast similarity search property inherited from Min-hashing. Finally, the proposed technique can potentially be extended for identification task and other binary biometric features.

5.2 Future Works

For future works, since IFO is naturally fit for binary input as discussed in this research, we are looking for its potential to be imposed into other biometric models which represented in binary format, i.e. binary fingerprint, face, etc. As each hashed code elements is independently derived from a single permutation. Conversely, through concatenate the hashing code elements from different biometric modality i.e. $\{h_1(\mathbf{Iris})\|h_2(\mathbf{face})\|h_3(\mathbf{Iris})\|\ldots\|h_m(\mathbf{face})\}$, it shows a straight forward and simple way to perform feature level fusion.

Besides, IFO is also potential to be used for cryptographic symmetric encryption. In fact, Rivest (2016) has proposed a symmetric encryption technique by using Min-hashing approach with the incorporated error correction code to tolerate the noise effect. Since IFO can be regarded as a special case of Min-hashing, it inherited the properties of "local distant preserving" from Min-hashing that potentially to be used as a key for message encryption/decryption with error correction code.

REFERENCES

Bledsoe, W.W., 1966. *Man-machine face recognition*. Panoramic Research Inc., Palo Alto, CA Technical Report PRI, 22.

Bodo, A., 1994. *Method for producing a digital signature with aid of a biometric feature*. German patent DE 4243908 A1.

Boles, W. W. and Boashash, B., 1998. A human identification technique using images of the iris and wavelet transform. *IEEE transactions on signal processing*, 46(4), pp.1185-1188.

Bringer, J., Morel, C. and Rathgeb, C., 2015, May. Security analysis of Bloom filter-based iris biometric template protection. *In 2015 International Conference on Biometrics (ICB)*, pp. 527-534.

Broder, A. Z., 1997, June. On the resemblance and containment of documents. In Compression and Complexity of Sequences 1997. *Proceedings IEEE*, pp. 21-29.

Broder, A. Z., Charikar, M., Frieze, A. M. and Mitzenmacher, M., 2000. Min-wise independent permutations. *Journal of Computer and System Sciences*, 60(3), pp.630-659.

Cappelli, R., Maio, D., Lumini, A. and Maltoni, D., 2007. Fingerprint image reconstruction from standard templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9), pp.1489-1503.

CASIA iris image database, Available: http://www.cbsr.ia.ac.cn/Database.htm.

Chin, C.S., Jin, A.T.B. and Ling, D.N.C., 2006. High security iris verification system based on random secret integration. *Computer Vision and Image Understanding*, 102(2), pp. 169-177.

Daugman, J., 2004. How iris recognition works. *IEEE Transactions on circuits and systems for video technology*, 14(1), pp. 21-30.

Daugman, J., 2006. Probing the uniqueness and randomness of IrisCodes: Results from 200 billion iris pair comparisons. *Proceedings of the IEEE*, 94(11), pp.1927-1935.

Daugman, J., 1993. High confidence visual recognition of persons by a test of statistical independence. *IEEE transactions on pattern analysis and machine intelligence*, 15(11), pp.1148-1161.

Dwivedi, R. and Dey, S., 2015, February. Cancelable iris template generation using look-up table mapping. *In Signal Processing and Integrated Networks (SPIN), 2015 2nd International Conference on*, pp. 785-790.

Ernst, R. H., 1971. *Hand ID system*. U.S. Patent 3,576,537, issued April 27.

Faulds, H., 1880. On the skin-furrows of the hand. *Nature*, 22, pp. 605.

Galbally, J., Ross, A., Gomez-Barrero, M., Fierrez, J. and Ortega-Garcia, J., 2013. Iris image reconstruction from binary templates: An efficient probabilistic approach based on genetic algorithms. *Computer Vision and Image Understanding*, 117(10), pp. 1512-1525.

Galton, F., 1889. Personal identification and description. *Journal of Anthropological Institute of Great Britain and Ireland*, pp.177-191.

Gomez-Barrero, M., Rathgeb, C., Galbally, J., Busch, C. and Fierrez, J., 2016. Unlinkable and irreversible biometric template protection based on bloom filters. *Information Sciences*, 370, pp.18-32.

Hämmerle-Uhl, J., Pschernig, E. and Uhl, A., 2009, September. Cancelable iris biometrics using block re-mapping and image warping. Springer Berlin Heidelberg. *In International Conference on Information Security (pp. 135-142).*

Hermans, J., Mennink, B. and Peeters, R., 2014, September. When a Bloom filter is a Doom filter: Security assessment of a novel iris biometric template protection system. *In Biometrics Special Interest Group (BIOSIG), 2014 International Conference of the,* pp. 1-6.

Herschel, W.J., 1880. Skin furrows of the hand. *Nature*, 23, pp.76.

Hollingsworth, K.P., Bowyer, K.W. and Flynn, P.J., 2009. The best bits in an iris code. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6), pp. 964-973.

Indyk, P., 2001. A small approximately min-wise independent family of hash functions. *Journal of Algorithms*, 38(1), pp.84-90.

Indyk P. and Motwani, R., 1998. Approximate nearest neighbors: Towards removing the curse of dimensionality. *In Proceedings of the. 30th STOC*.

Indyk, P., Motwani, R., Raghavan, P. and Vempala. S., 1997. Locality-preserving hashing in multidimensional space. *In Proceedings of the 29th STOC*.

Jain, A.K., Nandakumar, K. and Nagar, A., 2008. Biometric template security. *EURASIP Journal on Advances in Signal Processing*, 2008, pp. 113.

Jenisch, S. and Uhl, A., 2011, September. Security analysis of a cancelable iris recognition system based on block remapping. *In 2011 18th IEEE International Conference on Image Processing*, pp. 3213-3216.

Juels A., and Wattenberg, M., 1999. A fuzzy commitment scheme, *In Proceedings of the 6th ACM conference on Computer and communications security*.

Kong, A., Cheung, K.H., Zhang, D., Kamel, M. and You, J., 2006. An analysis of BioHashing and its variants. *Pattern Recognition*, 39(7), pp. 1359-1368.

Lacharme, P., Cherrier, E. and Rosenberger, C., 2013, July. Preimage attack on biohashing. *In Security and Cryptography (SECRYPT), 2013 International Conference on* pp. 1-8.

Lacharme, P., 2012. Analysis of the iriscodes bioencoding scheme. *International Journal of Computer Science and Security (IJCSS)*, 6(5), pp. 315.

Lai, Y. L., Jin, Z., Goi, B. M., Chai, T. Y. and Yap, W. S., 2016, September. Iris Cancelable Template Generation Based on Indexing-First-One Hashing. *In International Conference on Network and System Security*, Springer International Publishing, pp. 450-463.

Lim, S., Lee, K., Byeon, O. and Kim, T., 2001. Efficient iris recognition through improvement of feature vector and classifier. *ETRI journal*, 23(2), pp. 61-70.

Maltoni, D., Maio, D., Jain, A. and Prabhakar, S., 2009. *Handbook of fingerprint recognition*. Springer Science & Business Media.

Mauceri, A. J., 1965. *Feasibility studies of personal identification by signature verification*. Report no. SID, 65, pp. 24.

Masek, L., 2003. *Recognition of human iris patterns for biometric identification*. Master Thesis, The University of Western Australia, 2.

Nagar, A., Nandakumar, K. and Jain, A.K., 2010, February. Biometric template transformation: a security analysis. *In IS&T/SPIE Electronic Imaging*, pp. 75410O-75410O.

Nandakumar, K. and Jain, A.K., 2015. Biometric Template Protection: Bridging the performance gap between theory and practice. *IEEE Signal Processing Magazine*, 32(5), pp. 88-100.

Natgunanathan, I., Mehmood, A., Xiang, Y., Beliakov, G. and Yearwood, J., 2016. Protection of Privacy in Biometric Data. *IEEE Access*, 4, pp. 880-892.

Osama, O.U.D.A., Tsumura, N. and Nakaguchi, T., 2011. On the Security of BioEncoding Based Cancelable Biometrics. *IEICE TRANSACTIONS on Information and Systems*, 94(9), pp. 1768-1777.

Ouda, O., Tsumura, N. and Nakaguchi, T., 2010, August. Tokenless cancelable biometrics scheme for protecting iris codes. *In Pattern Recognition (ICPR), 2010 20th International Conference on*, pp. 882-885.

Patel, V. M., Ratha, N. K. and Chellappa, R., 2015. Cancelable biometrics: A review. *IEEE Signal Processing Magazine*, 32(5), pp. 54-65.

Pillai, J.K., Patel, V.M., Chellappa, R. and Ratha, N.K., 2010, March. Sectored Random Projections for Cancelable Iris Biometrics. *In ICASSP* pp. 1838-1841.

Prabhakar, S., Pankanti, S. and Jain, A.K., 2003. Biometric Recognition: Security and Privacy Concerns. *IEEE Security and Privacy*, 1(2), pp.33-42.

Pruzansky, S., 1963. Pattern-Matching Procedure for Automatic Talker Recognition. *The Journal of the Acoustical Society of America*, 35(3), pp.354-358.

Quan, F., Fei, S., Anni, C. and Feifei, Z., 2008, December. Cracking cancelable fingerprint template of Ratha. *In Computer Science and Computational Technology*, 2008. ISCSCT'08. *International Symposium on* 2, pp. 572-575.

Quinn, G. and Grother, P., IREX III Supplement I: Failure Analysis.

Ratha, N.K., Chikkerur, S., Connell, J.H. and Bolle, R.M., 2007. Generating cancelable fingerprint templates. *IEEE Transactions on pattern analysis and machine intelligence*, 29(4), pp. 561-572.

Rathgeb, C. and Uhl, A., 2011. A survey on biometric cryptosystems and cancelable biometrics. *EURASIP Journal on Information Security*, 2011(1), pp.1.

Rathgeb, C., Uhl, A. and Wild, P., 2013. Iris Recognition: From Segmentation to Template Security, volume 59 of Advances in Information Security.

Rathgeb, C., Breitinger, F. and Busch, C., 2013, June. Alignment-free cancelable iris biometric templates based on adaptive bloom filters. *In 2013 International Conference on Biometrics (ICB)*, pp. 1-8.

Rivest, R. L., 2016 *Symmetric encryption via keyrings and ecc*, [Online]. Available: http://arcticcrypt.b.uib.no/files/2016/07/Slides-Rivest.pdf.

Samir, N., Michael, T. and Raj, N., 2002. Biometrics: Identitiy Verification in a Networked World.

Scheirer, W. J. and Boult, T. E., 2007, September. Cracking fuzzy vaults and biometric encryption. *In Biometrics Symposium*, pp. 1-6.

Trauring, M., 1963. Automatic comparison of finger-ridge patterns. *Nature*, 197, pp.938-940.

Uhl, A. and Wild, P., 2012, March. Weighted adaptive hough and ellipsopolar transforms for real-time iris segmentation. *In 2012 5th IAPR International Conference on Biometrics (ICB)* pp. 283-290.

Teoh, A. B., Goh, A. and Ngo, D. C., 2006. Random multispace quantization as an analytic mechanism for biohashing of biometric and random identity inputs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12), pp. 1892-1901.

Venugopalan, S. and Savvides, M., 2011. How to generate spoofed irises from an iris code template. *IEEE Transactions on Information Forensics and Security*, 6(2), pp. 385-395.

Wildes, R. P., 1997. Iris recognition: an emerging biometric technology. *In Proceedings of the IEEE*, 85(9), pp.1348-1363.

Wolff, E., 1940. Anatomy of the Eye and Orbit, Including the Central Connections, Development, and Comparative Anatomy of the Visual Apparatus. *Nature*, 132, pp. 767.

Zhou, X., 2012. *Privacy and security assessment of biometric template protection*. Phd Thesis, it-Information Technology Methoden und innovative Anwendungen der Informatik und Informationstechnik, 54(4), pp.197-200.

Zuo, J., Ratha, N.K. and Connell, J.H., 2008, December. Cancelable iris biometric. *In Pattern Recognition, 2008. ICPR 2008. 19th International Conference on* pp. 1-4.

APPENDICES

APPENDIX A: Example of IrisCode and IFO hashed Code

IrisCode (binary bit 0 or 1)

0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	0	0	0
0	1	1	0	0	0	0	0	0	0	0
1	1	1	0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	0	0	0
0	0	0	0	0	1	1	1	0	0	0
0	1	1	1	1	1	1	1	0	0	0
0	0	0	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	0	0	0
0	1	1	0	0	0	0	0	0	0	0
1	1	1	0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	0	0	0
0	0	0	0	0	1	1	1	0	0	0
0	1	1	1	1	1	1	1	0	0	0
0	0	0	1	1	1	1	1	1	1	0

IFO hashed code with $K = 8$, we can see that a lot of ill-case with small K

0	2	0	4	1	0	1	0	0	1	0
0	1	1	0	0	0	0	0	0	0	3
1	0	4	1	0	2	1	1	0	1	0
0	1	0	2	0	0	0	0	2	0	1
6	1	3	5	0	4	1	1	2	0	2
2	7	3	4	3	0	2	0	1	0	1
2	1	0	4	0	1	1	1	2	1	2
3	1	0	2	2	0	0	3	2	0	1
1	5	0	2	0	1	1	1	1	0	0
1	0	3	4	1	0	0	3	1	0	1
1	0	0	1	0	1	2	2	0	0	0
0	1	2	0	4	3	0	1	0	4	2
0	2	0	4	1	0	1	0	0	1	0
0	1	1	0	0	0	0	0	0	0	3
1	0	4	1	0	2	1	1	0	1	0
0	1	0	2	0	0	0	0	2	0	1
6	1	3	5	0	4	1	1	2	0	2
2	7	3	4	3	0	2	0	1	0	1
2	1	0	4	0	1	1	1	2	1	2
2	1	0	4	0	1	1	1	2	1	2

ACHIEVEMENTS

Publications:

Journal article:

- Yen-Lung Lai, Z. Jin, A. B. J. Teoh, B-M Goi, W-S Yap, T-Y Chai, C. Rathgeb: **Cancelable iris template generation based on Indexing-First-One hashing**, ***Pattern Recognition***, Volume 64, April 2017, Pages 105-117, ISSN 0031-3203

Conference Papers:

- Yen-Lung Lai, Zhe Jin, Bok-Min Goi, Tong-Yuen Chai, Wun-She Yap, "**Iris Cancelable Template Generation Based on Indexing-First-One Hashing**", the 10th International Conference on *Network and System Security* (***NSS 2016***), Taipei, Taiwan, 28th Sept – 30th Sept 2016, Pages 450-463.
- Yen-Lung Lai, Zhe Jin, Bok-Min Goi, Tong-Yuen Chai, "**Generating Non-Invertible Iris Template for Privacy Preserving**", the 5th International Cryptology and Information Security Conference 2016 (***Cryptology2016***), Sabah, Malaysia, 31st May – 2nd June, 2016.
- Yen-Lung Lai, Zhe Jin, Bok-Min Goi, Tong-Yuen Chai, "**Study of The Similarity Function in Indexing-First-One Hashing**", the 2nd International Workshop on Pattern Recognition (***IWPR 2017***), Singapore, 1st May – 3rd May, 2017.
- Yen-Lung Lai, Bok-Min Goi, Tong-Yuen Chai, "**Alignment-free Indexing-First-One Hashing with Bloom Filter Integration**", IEEE International Conference on Intelligence and Security Informatics (***ISI 2017***), Beijing, China 22th -24th , 2017.

Patent:

- Malaysia Patent Application No: PI 2017702830. Entitled: ***Method and System for Binding Key***, University Tunku Abdul Rahman, 2 August 2017

Exhibition/Exposition:

- Silver Medal, "Bio-Crypto Authentication Solution (BCA)", 14th International Conference and Exposition on Inventions by Institutions of Higher Learning (PECIPTA 2015), Kuala Lumpur, Malaysia
- UTAR Engineering Fiesta 2016