AN ENHANCED FUZZY COMMITMENT SCHEME IN BIOMETRIC TEMPLATE PROTECTION

TAQIYAH KHADIJAH GHAZALI

DOCTOR OF PHILOSOPHY

UNIVERSITI UTARA MALAYSIA

2023

Awang Had Salleh
Graduate School
of Arts And Sciences

Universiti Utara Malaysia

PERAKUAN KERJA TESIS / DISERTASI

(Certification of thesis / dissertation)

Kami, yang bertandatangan, memperakukan bahawa
(We, the undersigned, certify that)

TAQIYAH KHADIJAH GHAZALI

calon untuk Ijazah
(candidate for the degree of) **PhD**

telah mengemukakan tesis / disertasi yang bertajuk:
(has presented his/her thesis / dissertation of the following title):

"AN ENHANCED FUZZY COMMITMENT SCHEME IN BIOMETRIC TEMPLATE PROTECTION"

seperti yang tercatat di muka surat tajuk dan kulit tesis / disertasi.
(as it appears on the title page and front cover of the thesis / dissertation).

Bahawa tesis/disertasi tersebut boleh diterima dari segi bentuk serta kandungan dan meliputi bidang ilmu dengan memuaskan, sebagaimana yang ditunjukkan oleh calon dalam ujian lisan yang diadakan pada : ***20 Januari 2022.***
That the said thesis/dissertation is acceptable in form and content and displays a satisfactory knowledge of the field of study as demonstrated by the candidate through an oral examination held on:
20 January 2022.

Pengerusi Viva: *(Chairman for VIVA)*	**Assoc. Prof. Dr. Yuhanis Yusof**	Tandatangan *(Signature)*
Pemeriksa Luar: *(External Examiner)*	**Prof. Dr. Azman Samsudin**	Tandatangan *(Signature)*
Pemeriksa Dalam: *(Internal Examiner)*	**Assoc. Prof. Ts. Dr. Norliza Katuk**	Tandatangan *(Signature)*
Nama Penyelia/Penyelia-penyelia: *(Name of Supervisor/Supervisors)*	**Assoc. Prof. Dr. Nur Haryani Zakaria**	Tandatangan *(Signature)*
Nama Penyelia/Penyelia-penyelia: *(Name of Supervisor/Supervisors)*	**Prof. Dr. Mohd Aizaini Maarof**	Tandatangan *(Signature)*

Tarikh:
(Date) ***20 January 2022***

Permission to Use

In presenting this thesis for the fulfilment of the requirements for a postgraduate degree from Universiti Utara Malaysia, I agree that the Universiti Library may make it freely available for inspection. I further agree, in giving permission for the copying of this thesis, in any manner; in whole or in part, for scholarly purpose, the permission may be granted by my supervisor(s) or, in their absence, by the Dean of Awang Had Salleh Graduate School of Arts and Sciences. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my thesis.

Requests for permission to copy or to make other use of materials in this thesis, in whole or in part, should be addressed to:

Dean of Awang Had Salleh Graduate School of Arts and Sciences,
UUM College of Arts and Sciences,
Universiti Utara Malaysia,
06010 UUM Sintok

Abstrak

Perlindungan templat biometrik terdiri daripada dua pendekatan; Transformasi Ciri (FT) dan Kriptografi Biometrik (BC). Penyelidikan ini memfokuskan kepada Teknik Pengikatan Kunci berdasarkan Skim Komitmen Kabur (FCS) di bawah pendekatan BC. Dalam FCS, data pembantu tidak seharusnya mendedahkan sebarang maklumat tentang data biometrik. Walau bagaimanapun, literatur menunjukkan bahawa ia mempunyai isu pergantungan dalam data pembantunya yang menjejaskan keselamatan dan privasi. Selain itu, ini juga meningkatkan kebarangkalian kebocoran privasi yang membawa kepada serangan seperti serangan kekerasan dan padanan silang. Oleh itu, tujuan penyelidikan ini adalah untuk mengurangkan kebergantungan data pembantu yang boleh menyebabkan kebocoran privasi. Tiga objektif telah ditetapkan seperti (1) untuk mengenal pasti faktor yang menyebabkan pergantungan pada ciri biometrik (2) untuk meningkatkan FCS dengan mencadangkan pendekatan yang mengurangkan pergantungan ini, dan (3) untuk menilai pendekatan yang dicadangkan berdasarkan parameter seperti keselamatan privasi dan prestasi biometrik. Penyelidikan ini melibatkan empat fasa. Fasa satu, melibatkan kajian dan analisis penyelidikan, diikuti dengan mereka bentuk model konseptual dan pembangunan algoritma dalam fasa dua dan tiga masing-masing. Fasa empat, terlibat dengan penilaian pendekatan yang dicadangkan. Analisis keselamatan dan privasi menunjukkan bahawa dengan fungsi cincang tambahan, adalah sukar untuk musuh melakukan serangan kekerasan terhadap maklumat yang disimpan dalam pangkalan data. Tambahan pula, pendekatan yang dicadangkan telah meningkatkan aspek ketidakterpautan dan menghalang serangan padanan silang. Pendekatan yang dicadangkan telah mencapai ketepatan yang tinggi iaitu 95.31% dengan Kadar Ralat Sama (EER) sebanyak 1.54% yang menunjukkan prestasi yang lebih baik sedikit sebanyak 1.42% berbanding pendekatan sedia ada. Penyelidikan ini telah menyumbang ke arah teknik mengikat kunci perlindungan templat cap jari biometrik, berdasarkan FCS. Khususnya, penyelidikan ini direka bentuk untuk mencipta ciri binari rahsia yang boleh digunakan dalam sistem kriptografi tercanggih yang lain dengan menggunakan pendekatan pembetulan ralat yang sesuai yang memenuhi piawaian keselamatan.

Kata Kunci: Kriptografi biometrik, Perlindungan templat biometrik, Skim komitmen kabur, Kekunci-pengikat

Abstract

Biometric template protection consists of two approaches; Feature Transformation (FT) and Biometric Cryptography (BC). This research focuses on Key-Binding Technique based on Fuzzy Commitment Scheme (FCS) under BC approach. In FCS, the helper data should not disclose any information about the biometric data. However, literatures showed that it had dependency issue in its helper data which jeopardize security and privacy. Moreover, this also increases the probability of privacy leakage which lead to attacks such as brute-force and cross-matching attack. Thus, the aim of this research is to reduce the dependency of helper data that can caused privacy leakage. Three objectives have been set such as (1) to identify the factors that cause dependency on biometric features (2) to enhance FCS by proposing an approach that reduces this dependency, and (3) to evaluate the proposed approach based on parameters such as security, privacy, and biometric performance. This research involved four phases. Phase one, involved research review and analysis, followed by designing conceptual model and algorithm development in phase two and three respectively. Phase four, involved with the evaluation of the proposed approach. The security and privacy analysis shows that with the additional hash function, it is difficult for adversary to perform brute-force attack on information stored in database. Furthermore, the proposed approach has enhanced the aspect of unlinkability and prevents cross-matching attack. The proposed approach has achieved high accuracy of 95.31% with Equal Error Rate (EER) of 1.54% which performs slightly better by 1.42% compared to the existing approach. This research has contributed towards the key-binding technique of biometric fingerprint template protection, based on FCS. In particular, this research was designed to create a secret binary feature that can be used in other state-of-the-art cryptographic systems by using an appropriate error-correcting approach that meets security standards.

Keywords: Biometric cryptosystem, Biometric template protection, Fuzzy commitment scheme, Key-binding

Acknowledgement

First of all, I want to express my thanks and gratitude to Allah who gives me the ability to achieve this imperfect work and without His blessing and support nothing can be done.

I would like to thank my supervisors, Associate Professor Dr Nur Haryani Zakaria for her continuous support during my PhD program. I'm so grateful for her support who was also my lecturer during my master's study at UUM, there are no words to express my gratitude for her guidance in helping me to achieve my goal. Without his valuable support, my thesis would not have been possible. I would also like to thank my co-supervisor Professor Dr. Aizaini Maarof for his advanced ideas and his noble mind. His continuous advice and important comments helped improve my work successfully.

I would also like to offer my deepest gratitude to my parents and family for helping me, and for encouraging me to do my PhD work which enabled me to successfully accomplish my tasks.

I am deeply indebted to my dear friends in UUM especially from the School of Computing for taking so much time and interest in my work, always being there for me through my difficult situations and spending their time in guiding me despite their busy schedule.

Last but not least, thanks to all those who have been directly and indirectly involved in helping me complete this research. Finally, I apology to those whose names might be missing. But I am grateful to all of them for their help and on various matters of day-to-day life.

TABLE OF CONTENTS

Permission to Use i
Abstrak ii
Abstract iii
Acknowledgement iv
Table of Contents v
List of Tables viii
List of Figures ix
List of Abbreviations xi

CHAPTER ONE INTRODUCTION 1

1.1 Introduction 1
1.2 Background of Research 1
1.3 Background of the Problem 6
1.4 Problem Statement 8
1.5 Research Questions 10
1.6 Research Aim and Objectives 11
1.7 Research Scope 11
1.8 Significance of the Study 13
1.9 Thesis Organization 14

CHAPTER TWO LITERATURE REVIEW 16

2.1 Introduction 16
2.2 Biometric Template Protection 16
2.3 Biometric Cryptosystem (BC) 20
2.3.1 Key Generation Technique 22
2.3.2 Key Binding Technique 25
2.4 Fuzzy Commitment Scheme (FCS) 28
2.4.1 Feature Vector Module 30
2.4.2 Bit String Generation 41
2.4.3 Error-Correcting Codes (ECC) 41
2.4.4 Helper Data *(HD)* 44
2.4.5 Hash 46

2.5 Privacy Leakage in Fuzzy Commitment Scheme 51
2.5.1 Dependency on Biometric Features 53
2.5.2 Effect on Privacy Leakage: Reversible Attack 54
2.6 Privacy Requirements 56
2.6.1 Maximizing the Key Size 57
2.6.2 Tradeoff Between Performance, Privacy and Security 58
2.7 Related Works 59
2.8 Summary 61

CHAPTER THREE RESEARCH METHODOLOGY 63

3.1 Introduction 63
3.2 The Research Framework 63
3.2.1 Phase 1 – Research Review and Analysis 64
3.2.2 Phase 2 – Designing the Conceptual Model 66
3.2.3 Phase 3 – Algorithm Development 69
3.2.4 Phase 4 – Evaluation 72
3.3 Summary 79

CHAPTER FOUR THE ENHANCED FUZZY COMMITMENT SCHEME ... 80

4.1 Introduction 80
4.2 The Proposed Approach 80
4.2.1 Enrollment Module 84
4.2.2 Concealment Module 89
4.2.3 Verification Module 96
4.3 Summary 101

CHAPTER FIVE

5.1 Introduction 103
5.2 Security Evaluation 103
5.3 Privacy Evaluation 105
5.4 Performance Evaluation 107
5.5 Comparative Analysis 116
5.6 Summary 119

CHAPTER SIX

6.1 Introduction……...121
6.2 Discussion……..……..121
6.2.1 Research Question (1) 121
6.2.2 Research Question (2) 123
6.2.3 Research Question (3) 124
6.3 Research Contribution..... 125
6.4 Limitations of the Research 127
6.5 Future Works 127
6.6 Summary 128

REFERENCES 128

APPENDIX A

APPENDIX B SOURCE CODE OF THE PROTOTYPE 155

List of Tables

Table 2.1 Summary of techniques in FT and BC…….19
Table 2.2 Hash algorithms comparisons……..…50
Table 3.1 The main modules in the enhanced FCS……..…68
Table 4.1 Notations……..82
Table 4.2 Concealment process steps…….….90
Table 4.3 Verification process steps……..……..97
Table 5.1 The match score sample for the calculation of the true positive rate of the proposed approach……109
Table 5.2 The match score for the calculation of the true negative rate of the proposed approach……111
Table 5.3 Calculation of FRR, FAR, TPR and accuracy……...112
Table 5.4 FAR, FRR and accuracy values with different thresholds…….113
Table 5.5 Comparison between the proposed approach and the comparison study……..…116
Table 5.6 EER comparison between Sandhya et al. (2020) and the proposed approach……119

List of Figures

Figure 1.1. Points of attack of an existing fingerprint recognition system……4
Figure 1.2. Biometric template protection techniques……6
Figure 1.3. Scope of the research……...12
Figure 2.1. Categorization of template protection technique……18
Figure 2.2. Techniques in BC……21
Figure 2.3. The secure sketch scheme……23
Figure 2.4. The fuzzy extractor……..24
Figure 2.5. The Fuzzy Vault Scheme……27
Figure 2.6. Existing Fuzzy Commitment Scheme……..28
Figure 2.7. Ridges and valleys in a fingerprint image……32
Figure 2.8. Seven most common types of minutia……33
Figure 2.9. A section of a fingerprint captured at 1000 dpi……...34
Figure 2.10. OM and SPs are two global fingerprint traits……35
Figure 2.11. Examples of loop, whorl, and arch patterns……...36
Figure 2.12. The SKI attack……...51
Figure 2.13. Attack via record multiplicity (ARM)……...53
Figure 2.14. Inverse biometric method for synthetic sample generation……...55
Figure 2.15. An inversion method……..55
Figure 3.1. The research framework……...64
Figure 3.2. The conceptual model of the enhanced FCS……67
Figure 3.3. Percentage of FAR, FRR and EER versus sensitivity……76
Figure 3.4. Flow diagram of the approach proposed by Sandhya et al. (2020)……..78
Figure 4.1. Flowchart of the proposed approach……83
Figure 4.2. Flowchart of the enrollment module……88
Figure 4.3. Flowchart of the concealment module……..…...94
Figure 4.4. Flowchart of the verification module……..…98
Figure 4.5. Flowchart of the verification module (continued)……..…99
Figure 5.1. ROC curves of the proposed approach……..……...114
Figure 5.2. DET curves of the proposed approach……...……115
*Figure 5.3.*The EER of FAR and FRR for the proposed approach……..…….115
Figure 5.4. ROC curves are used to compare the performance of Sandhya et al. (2020) and the proposed approach from FVC 2002 DB1……118

List of Abbreviations

AAD	Average Absolute Deviation
ARM	Attacks via record multiplicity
BC	Biometric Cryptosystem
BCH	Bose–Chaudhuri–Hocquenghem
BTP	Biometric Template Protection
CHF	Cryptography Hash Function
DNA	Deoxyribonucleic Acid
dpi	Dots Per Inch
ECC	Error-Correcting Code
ECG	Electrocardiography
ED	Euclidean Distance
EEG	Electroencephalography
EER	Equal Error Rate
FAR	False Accept Rate
FCS	Fuzzy Commitment Scheme
FE	Fingerprint Extraction
FIPS PUB	Federal Information Processing Standards Publications
FN	False Negative
FP	False Positive
FRR	False Reject Rate
FT	Feature Transformation
FVC	Fingerprint Verification Competition
FVS	Fuzzy Vault Scheme
GF	Galois Field
HD	Helper Data
IEC	International Electrotechnical Commission
IoT	Internet of Things
ISO	International Organization for Standardization
JTC	Joint Technical Committee

m	meter
MD	message-digest
NIST	National Institute of Standards and Technology
nm	nanometer
NSA	National Security Agency
OM	Orientation Map
PI	Pseudonymous Identifiers
PIN	Personal Identification Number
RFC	Request for Comments
RNG	Random Number Generator
ROC	Receiver Operating Characteristic
ROT	Rotation
RP	Random Projection
SHA	Secure Hash Algorithms
SHR	Right Shift
SKI	Surreptitious key-inversion
SPs	Singular Points
TIR	Total Internal Reflection
TN	True Negative
TP	True Positive
TPR	True Positive Rate
UID	Uniformly and independently distributed
XOR	Exclusive OR

CHAPTER ONE

INTRODUCTION

1.1 Introduction

This chapter provides an introduction to the research, which begins with a description of the background of the study, followed by a discussion of the research problem. Then, the research questions are presented and used to formulate the research objectives. The chapter describes the scope of this research and highlights the significance of the study. Finally, the chapter concludes with a summary of the thesis organization.

1.2 Background of Research

Biometrics is a term made up of two words: 'bio', referring to the life of living beings, and 'metrics', referring to a system or standard for measurement (Ilchenko et al., 2020; Jegede et al., 2017). Biometrics were first used in law enforcement and legal applications, such as convict and inmate identification, biological identification, and forensics (Ashish & Sinha, 2017; Borgianni & Maccioni, 2020; Ross et al., 2020). Nowadays, biometric technologies are used in a variety of areas around the world, including financial and trade surveillance, physical access control, cybersecurity, customs and immigration, national identity cards, and driving licenses, to name a few (Manikpuri, 2017; Rane et al., 2020).

The emergence of the Internet of Things (IoT) has spawned a variety of applications that rely on authentication or registration to prove a person's identity, which requires the use of biometrics (Obaidat et al., 2019). This is due to the fact that biometrics has a stronger authentication mechanism than other available authentication mechanisms.

For example, traditional passwords or PIN numbers can be retrieved by hackers and easily forgotten, while smartcards can be easily lost or stolen (Riaz et al., 2017; Ross et al., 2020; Sabhanayagam et al., 2018).

Moreover, biometrics can curb the rise of identity theft and meet the increasing security requirements for secure networks and databases (Datta et al., 2020; Pagnin & Mitrokotsa, 2017). It is clear that security concerns have evolved beyond traditional methods like keys and padlocks, extending beyond physical security. The significance of a robust authentication mechanism is now widely recognized (Arora & Bhatia, 2021; Jain et al., 2016).

The majority of biometric traits currently utilized or in the process of development primarily focus on features derived from two types of biometric characteristic systems: physiological and behavioral characteristics. Physiological characteristics encompass various aspects such as face, fingerprint, hand geometry, hand vein, iris, retinal pattern, palm print, ear shape, fingernail bed, teeth, facial thermogram, deoxyribonucleic acid (DNA), as well as bioelectrical signals like heart signal (ECG) and brain signal (EEG) (Maiorana et al., 2016; Nezhad et al., 2020; Peter et al., 2016; Rinaldi, 2016). Conversely, behavioral characteristics include signature, gait, voice, body odor, and keystroke dynamics (Dargan & Kumar, 2020; Sabhanayagam et al., 2018).

Compared to the other biometric traits mentioned above, fingerprint recognition systems are the most researched and widely used (Geng et al., 2019). Human fingerprints are comprehensive, unique, unalterable, and long-lasting, making them ideal lifelong identifiers of personal existence (Bose & Kabir, 2017; Dwivedi et al.,

2020). The distinctive pattern of valleys and ridges in a fingerprint is established shortly after birth, and even identical twins exhibit different fingerprint patterns (Jain et al., 2016; Alsmirat et al., 2019).

The performance of fingerprint recognition systems is considered very high, and the general acceptance of fingerprint acquisition by the public is reasonable. As a result, it has been studied as one of the best mechanisms for personal authentication compared to traditional authentication (i.e., password) and is also the most widely used technology in the biometric field (Trivedi et al., 2020).

The existing fingerprint recognition system has made an important contribution in terms of access control mechanisms, law enforcement, and health issues. It can also provide an alternative to traditional access control mechanisms such as passwords, PIN, and smart cards (Kapoor & Sharma, 2016). Hence, fingerprint recognition systems have a large market share and are used in various applications (Ali et al., 2018).

While the fingerprint recognition system has a bright future, it also has some limitations (Yang et al., 2019). Fingerprint recognition systems are used in many fields, and their security and safety are an important issue (Galbally et al., 2019). With advances in technology, biometrics template information can be vulnerable to privacy and security threats (Mehmood & Selwa, 2019). For example, the theft of fingerprint templates can result in information leakage (Ashish & Sinha, 2017). This is because fingerprint remains unchanged throughout life, which means that if the fingerprint data has been disclosed, it is considered insecure (Trivedi et al., 2020).

Security threats can be defined by possible attacks on various components of a fingerprint recognition system (Jayapal, 2017; Joshi et al., 2020). An example of the attacks on the storage component where the template is stored is the so-called replay attack. A replay attack is when an intruder captures and uses or alters a victim's biometric data that is sent over a network as a template. The intruder pretends to be the victim and gets access without permission. Also, the template taken from the database is sent back to the matcher (Ali et al., 2020; Shelton et al., 2012). Figure 1.1 shows the points of attack of the existing fingerprint recognition system. The safety of the template stored in the database is very important and is the most vulnerable to attackers among these eight security sources (Dargan & Kumar, 2020; Ratha et al., 2003).

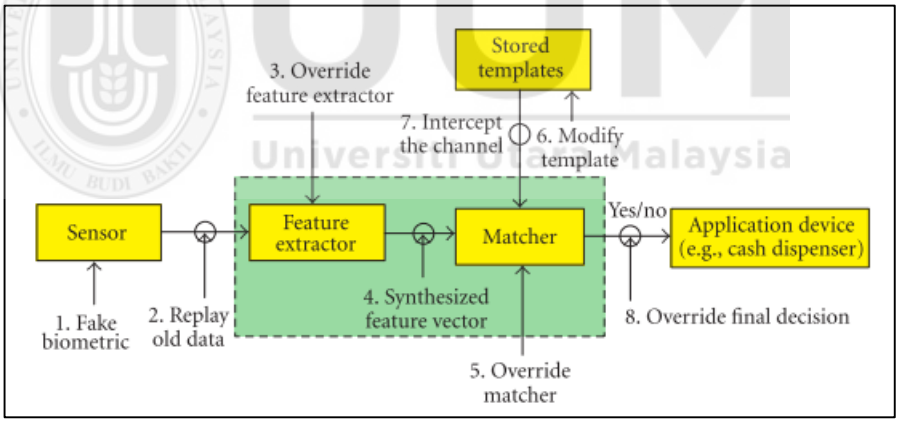


Figure 1.1. Points of attack of an existing fingerprint recognition system (Ratha et al., 2003)

Based on these points of attack, the biometric template may be vulnerable to attackers through one of the following:

1. Change, swap, and take the biometric data to unlawfully enter the device with the application.

2. Attempt to generate a biometric template with the intention of committing a physical forgery to gain unauthorized access to the system and other systems utilizing the same biometric traits.
3. Illicitly acquire the biometric templates to bypass authentication mechanisms and gain unauthorized access.
4. Utilize the biometric data for cross-matching with other databases, surreptitiously tracking an individual without their consent (Dwivedi et al., 2019; Yang et al., 2019).

In fact, spoofing biometric templates remains the most persistent form of attack on biometric systems, where the stolen templates are utilized to bypass the security checks of the biometric system at a later time (Galbally et al., 2019). As a result, data security is needed to address these issues and strengthen security and privacy features to be well protected (Lafkih et al., 2016; Obaidat et al., 2019).

In terms of drawbacks, the fingerprint recognition system is one of the biometric traits that has received the most coverage, not only from analysts and suppliers, but also from the media and consumers (Patel & Ramalingam, 2019). The biometrics community's growing interest in evaluating the protection of fingerprint recognition systems against attacks has sparked interest among researchers in revealing, reviewing, and testing (Galbally et al., 2019; Yang et al., 2019), which are also the focus of this research.

1.3 Background of the Problem

A biometric database stores a biometric template, which is a digital representation of the distinctive traits collected from a biometric sample. In order to protect these templates, two main techniques have been proposed, namely (1) feature transformation (FT) and (2) biometric cryptosystems (BC) (Riaz et al., 2017; Chauhan & Sharma, 2018; Sarkar & Singh, 2020; Dwivedi et al., 2020). Therefore, protecting fingerprint templates has been cited as one of the most important contributions to preventing attacks on the existing fingerprint-based recognition system (Müftüoğlu & Yildirim, 2019). Figure 1.2 summarizes the types of biometric template protection techniques.

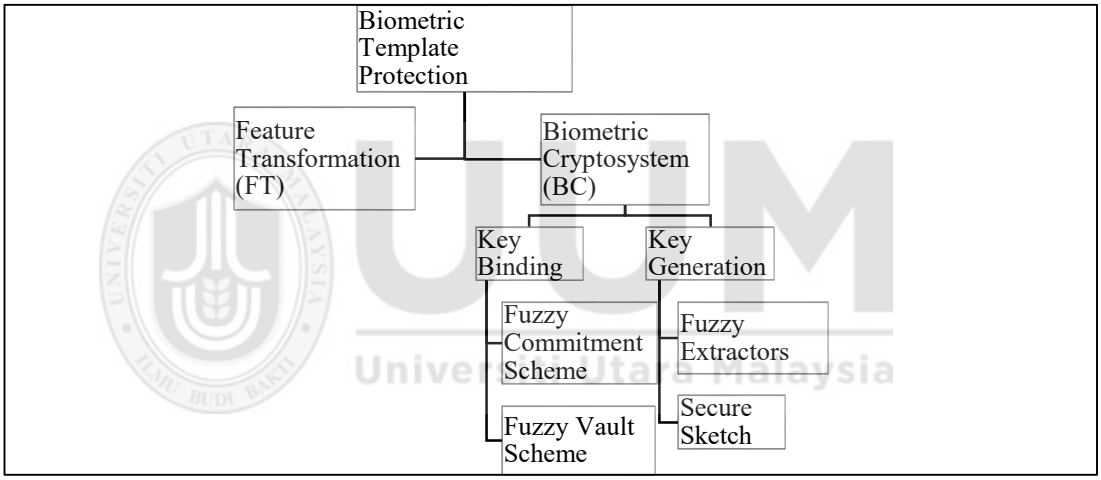


Figure 1.2. Biometric template protection techniques

The idea behind FT is the original biometric template is converted into a secure domain using a function. Usually, the function is dependent on a key (user-based or system-based). Instead of using the original template, the transformed template is used (Trivedi et al., 2020). The newly entered biometric data undergoes the same transformation during the matching stage, and matching occurs in the secure transformed domain. A transformation is chosen and performed to raw biometric

templates, and the changed template can be cancelled whenever required (Gunjan et al., 2020).

Another way of protecting biometric template, known as BC, is a standard technique in which a cryptographic key is linked with the biometric template data, resulting in 'helper data'. Helper data are the information that depend on and are referenced by the specific biometric trait (Riccio, Galdi, et al., 2016; Tantubay & Bharti, 2020). Depending on the method used to obtain the origin of helper data, BC relies on one of two techniques: (1) key binding or (2) key generation (Lutsenko et al., 2021). In the key binding technique, a secret key is associated with the biometric template during the enrollment process, resulting in the generation of helper data. The helper data combines *h(fingercode)* and *Key* and serves the purpose of safeguarding both the original biometric template and the key stored in the database (Adamovic et al., 2017). During authentication, the key is extracted by utilizing the query biometric data and the stored helper data. Key binding can be classified into two categories: (1) fuzzy vault and (2) fuzzy commitment schemes.

Key generation, on the other hand, uses a biometric template to create keys and helper data during enrollment, which are not always kept in the database. Using the query biometric features, the helper data during authentication assists in retrieving the key. Fuzzy extractors and secure sketches are two examples of key generation techniques (Riaz et al., 2017). The key generation technique is usually used for behavioral biometrics such as voice, gait, and typing patterns (Ballard et al., 2007). Thus, the key binding technique is considered more suitable for protecting fingerprint templates (Sadhya et al., 2016; Sapkal & Deshmukh, 2016).

The fuzzy commitment scheme (FCS) in key binding technique is one of the most implemented and investigated schemes in the research area as it is lighter in weight compared to the fuzzy vault scheme (FVS) and suitable for use in constrained devices (Sadhya et al., 2016). Thus, it is considered more appropriate for fingerprint template protection (Jin et al., 2016). However, FCS also has disadvantages. There are some vulnerable attacks on FCS, such as cross-matching attacks, blended-substitution attacks, hill-climbing attacks, and nearest-impostor attacks (Cavoukian & Stoianov, 2015; Jegede et al., 2017).

In FCS, the user relies on helper data, a component in a FCS, where it was created by binding a cryptographic key to the biometric template data and kept in the database (Riaz et al., 2017). It is important that the helper data does not reveal any information about the biometric data (Sadhya et al., 2016). However, previous research has shown that helper data reveals some important information about the user's biometric data that is vulnerable to privacy leakage and privacy attacks (Mwema et al., 2015; Sandhya & Prasad, 2017). In the next section, the problem statement of this research is explained in more detail.

1.4 Problem Statement

In the existing FCS, the helper data are created as a codeword from a specified error-correcting code that is used to encode a selected secret that is hidden by the biometric sequence seen during enrolment. The concept is particularly intended for biometrics features that are characterized by a binary uniform. In practice, however, biometric features are rarely uniform (Ignatenko & Willems, 2010).

This is because the existing FCS has some problems in its helper data due to the dependency on biometric features. The current FCS uses binary features as input, and the real-valued biometric features must first be binarized during enrollment. A straightforward binarization process results in the binary features taking dependency from the real-valued features, which significantly compromises security and privacy, increasing the probability of privacy leakage (Lafkih et al., 2016; Zhou et al., 2011).

Apart from that, a straightforward binarization occurs when the sample image from the dataset contains an inherent dependency, which makes the information accessible to attackers (Zhou et al., 2011). Besides, the biometric features are not uniformly random or have low entropy, which can be exploited by an attacker through statistical analysis to crack the helper data (Riaz et al., 2017). Previous literature has already observed that the helper data reveals information about the secret key in FCS when the biometric features are not uniformly random (Jin et al., 2016; Riaz et al., 2017; Zhou, 2012). In addition, privacy leakage may compromise an intrinsic characteristic of the individual (Ignatenko & Willems, 2010).

Moreover, privacy leakage has also led to reversibility attacks. These attacks happen when the information stored in reference templates is used to construct synthetic samples that are then analyzed. These samples can then be used to (1) launch masquerade attacks (i.e., impersonate a subject), thereby reducing the security of the system, or (2) obtain information from the subject's owner, resulting in the compromise of the user's personal information (Barrero, 2019).

In addition, the existing FCS does not guarantee unlinkability properties, which leads to cross-matching attacks (Simoens et al., 2009). A cross-matching attack is the process of linking reference templates of a person stored in different databases from different applications. This leads to an attacker gaining some knowledge about the user (Chauhan & Sharma, 2018).

Security and privacy are critical in this context to prevent unlawful access to the transmitted biometric template data. This indicates that the existing FCS does not meet the hiding and binding characteristics of biometric personalities and is considered a major drawback (Grigorescu et al., 2017; Sadhya et al., 2016). Therefore, this research intends to propose a suitable approach to enhance the existing FCS. The main aim is to improve the abovementioned problems, i.e., to reduce the dependency on biometric features in helper data while maintaining its current performance.

1.5 Research Questions

Based on the above problem statement, the following research questions were identified:

1. What are the factors that cause dependency on biometric features in helper data?
2. How can the dependency on biometric features in helper data be reduced?
3. Can the proposed approach enhance the existing FCS to reduce the dependency on biometric features in helper data to decrease privacy leakage?

1.6 Research Aim and Objectives

This research aims to reduce the dependency on helper data that may lead to privacy leakage. Thus, the specific objectives are as follows:

1. To identify the factors that cause dependency on biometric features in helper data.
2. To enhance FCS by proposing an approach that reduces the dependency on helper data. This objective is then divided into two sub-objectives:
   (a) to design a conceptual model for the proposed approach.
   (b) to develop an algorithm for the proposed approach.
3. To evaluate the proposed approach based on parameters such as security, privacy (irreversibility and privacy leakage), and biometric performance.

1.7 Research Scope

This research addresses the area of authentication mechanisms, specifically the protection of biometric fingerprint templates. The research is limited to the scope of BC and focuses on key binding techniques along with FCS. Figure 1.3 illustrates the scope by highlighting the specific scope.

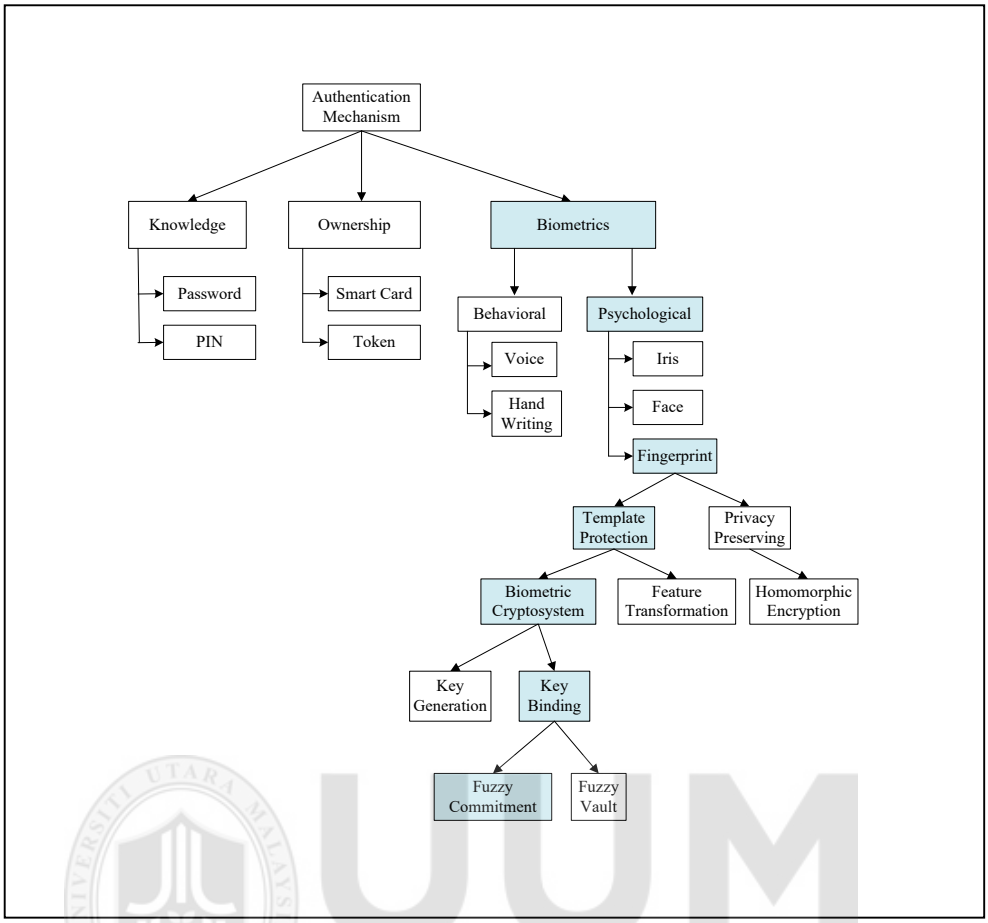


Figure 1.3. Scope of the research

The biometric fingerprint dataset from the Fingerprint Verification Competition (FVC) year 2002 (FVC2002, 2002) was used for this study. The FVC is an international competition that focuses on fingerprint verification software assessment. The objective of FVC 2002 is to track current advances in fingerprint verification in both scholarly and commercial areas. In addition, the dataset is also a well-known efficient standard for fingerprint technology.

The FVC2002 consists of four datasets, three of which contain three real fingerprint templates and one of which contains synthetic templates. Each dataset contains 100 fingers and 8 samples per finger (800 fingerprints in total). Each dataset was collected

using different sensors. For example, Dataset 1 and Dataset 2 were collected using optical sensors, but from different brands. Meanwhile, Dataset 3 was collected using a capacitive sensor and Dataset 4 was collected using synthetically generated fingerprints. For this study, fingerprint templates acquired with optical sensors are used, i.e., Dataset 1. This is because this type of fingerprint template has been used in many previous studies due to its simplicity and reliability.

This study uses "MATLAB R2017a environment on a machine with Intel (R) Core (TM) i7 ~2.50GHz CPU and 8-GB RAM". The proposed approach was developed using MATLAB Programming Language. In this research, the proposed approach was executed in a simulation environment.

1.8 Significance of the Study

Privacy protection for security applications of biometric cryptography technology has shown an increasing demand in terms of privacy protection and information security. Hence, the significance of this research work lies in the development of a biometric system that encompasses both privacy protection and security, which are necessary for the success of practical applications. Besides, biometric data can be placed under the sole control of the users while ensuring a high level of privacy for the data subject.

Apart from that, this research focuses on the FCS, which helps to protect the biometric template. This has a positive impact on the security and privacy of the system. For example, in the proposed approach, the helper data should not contain information about the biometric feature or the secret key. Furthermore, the analyzed information helps to further strengthen the security of the key.

Moreover, this research ensures the required level of security that can be used for law enforcement, military, administration, diplomatic, and other applications and can be used for all biometric traits.

1.9 Thesis Organization

This thesis is divided into six chapters. The synopsis of each chapter is as follows;

Chapter 1 introduces the research by explaining concepts related to the protection of biometric data, fingerprints and templates. It also introduces the problem statement, research questions, objectives, scope and significance of the study. This provides a general overview and direction of the research.

Chapter 2 focuses on the literature review, which provides a basic understanding of the context of existing research in the field. Towards the end of this chapter, a comprehensive overview of the research gap is highlighted to justify why this research was proposed. In addition, this chapter also analyzes the elements that cause the dependency on biometric features in helper data, which directly addresses the first objective of this research.

Chapter 3 describes in detail the research processes that were conducted as part of the research methodology. It explains in more detail how each objective was achieved through different phases that were organized throughout the research process. This chapter also includes some parameters used in the evaluation to validate the proposed approach.

Chapter 4 presents the proposed approach. This includes the conceptual design along with the algorithm of the proposed approach. Further explanation of the operation of the proposed approach is given in this chapter, which directly addresses research objectives 2 (a) and (b).

Chapter 5 presents the experimental results to allow an evaluation of the proposed approach. It also includes a comparative analysis with existing studies and a discussion of the experimental findings in order to highlight the contribution of this research.

Chapter 6 provides an overall discussion of how the research was conducted and how it achieved its aim and objectives. It also shows that the research questions posed in chapter one was answered. Besides, this chapter highlights the contribution of the research and identifies some limitations that could not be avoided but may be improved if a different platform or location is chosen for implementation. Finally, the conclusion is discussed along with some recommendations for future work.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This chapter presents the relevant literature on this area of research. It begins with a discussion of biometric template protection in Section 2.2, followed by a discussion of BC in Section 2.3. In the subsections under 2.3 (i.e., 2.3.1 – 2.3.2), two main techniques of BC are explained, namely key generation and key binding. The discussion then continues with Section 2.4, which focuses on FCS. In the following subsections under 2.4, all related components of FCS are described in detail. Section 2.5 discusses the issue of privacy leakage in FCS, which also highlights the gap in research. This is followed by Section 2.6, which explains the privacy requirements of FCS. Section 2.7 presents the related work and discusses all related research in this area. Finally, the chapter ends with a summary in Section 2.8.

2.2 Biometric Template Protection

A biometric system is a technological solution that collects biometric data samples from individuals, analyzes their distinctive features, and matches them against previously stored samples in a database (Vorobyeva et al., 2014). This established system can serve as either a verification mechanism or an identification tool.

An ideal biometric template protection method must satisfy various criteria such as diversity, revocability, security, and superior recognition performance. The subsequent traits outline the sought-after attributes of template protection schemes (Mehmood & Selwa, 2019; Sapkal & Deshmukh, 2016).

1. *Diversity*: Given two secure templates obtained from exactly similar biometric information, it should be almost impossible to detect that the two identical templates were derived from the same original biometric information.
2. *Revocability*: A compromised template should be revocable. This will ensure a new template can be re-released from the identical biometric information.
3. *Security*: Assuming a secure template, it must be computationally difficult to find biometric information that matches the query template.
4. *Performance*: The protection mechanism procedure shall not affect the accuracy performance (i.e., false acceptance rate and false rejection rate) of the biometric system.

There are several studies that apply protection techniques to the security of biometric information. The template protection approaches studied in the literature generally fall into two categories: (1) FT and (2) BC (Mehmood & Selwa, 2019; Siswanto et al., 2018). Figure 2.1 shows the categories of template protection approaches, namely feature transformation and biometric cryptosystem.

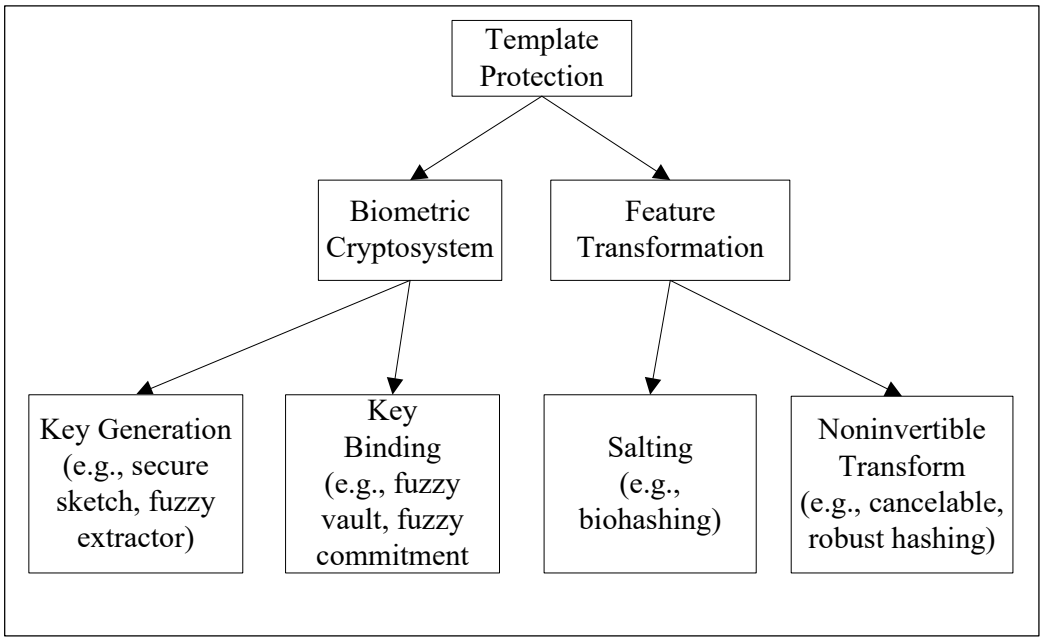


Figure 2.1. Categorization of template protection technique (Jain et al., 2016)

The biometric template undergoes a transformation using FT, utilizing parameters derived from external information like user passwords or keys. During the authentication process, the query is also subjected to the same transformation function and compared to the stored template in the transformed domain. Two primary techniques employed in FT are (1) salting and (2) non-invertible transformation.

On the other hand, the primary objective of BC is to gather error-correcting data from biometric characteristics, either with or without the presence of an external key, referred to as helper data. It is essential for the helper data to not disclose significant information about the biometric features or the key. Error-correcting codes are commonly utilized in such systems to retrieve the registered biometric features or secret key by utilizing the provided biometric data (Nagar et al., 2010). In the context of BC, there are two main techniques, namely; (1) key binding and (2) key generation.

Table 2.1 summarizes the different techniques in FT and BC, respectively (Sarkar, 2018). The next section discusses BC in more detail, as this is the focus of this study.

Table 2.1

Summary of techniques in FT and BC

BC Technique	Name of the Technique	Methodology	Advantages	Disadvantages
FT	Salting	Each individual's unique secret key determines the modification of the original biometric features through a designated transformation function.	The rate of erroneous acceptance is low. A single user can generate many templates by using a different key each time. If the compromised key has been hacked, it can be revoked easily and be restored with a fresh one.	Large variations between users cannot be accounted for.
	Non-Invertible Transformation	A non-invertible transformation function is employed to alter the biometric template, rendering it irretrievable to its original form.	A transformation without using the key. The original template is difficult to determine. Provide information specific to the program and the user.	The transformation function must possess the dual characteristics of discriminability and non-invertibility.
BC	Key generation	The helper data encompasses both the biometric template and the key.	Very useful for numerous cryptographic applications.	Simultaneously achieving both high key stability and high entropy is a challenging task.
	Key binding	The helper data holds both the biometric template and the key.	The ability to correct errors helps deal with the variability that occurs within a single individual.	It is necessary to adjust for the reduction in matching precision caused due to matching, by using a matcher with unique error-correction capabilities.

2.3 Biometric Cryptosystem (BC)

Long-term keys are needed to prevent conventional cryptographic methods from exploiting the stored biometric information. It is important to keep these keys secret so that no one else can access the user's biometric data. The goal of BC technology is to eliminate the disadvantages of both techniques (key binding and key generation) by eliminating the need for long-term keys. This prevents the misuse of keys by the owner of the biometric system while still enabling biometric identification. Thus, the goal of BC technology may be to protect sensitive biometric information from being taken from biometric storage facilities without the requirement for long-term keys (Turakulovich et al., 2018).

On the other hand, some researchers have discovered that the goal of BC is to create or disseminate a cryptographic key that may be used, among other things, to encrypt personal data (Ankit & Rekha, 2016; Dwivedi et al., 2020). However, key generation from a biometric is just a side effect. The primary purpose of BC is to protect biometric information. One reason for this is that the generated key can only be as secure as the biometric data, which attackers could obtain through other means such as traces or physical touches (Dwivedi et al., 2020).

The ISO standardization activities [ISO/IEC JTC1/SC37] indicate that protecting the privacy of data subjects includes the following aspects (ISO/IECJTC1, 2021);

1. Ensuring that third parties and external observers do not have access to biometric data or derived attributes that are not required by and consented to by the data subject (e.g., any information that is not required for biometric identification or verification in a specific service context)

2. Ensuring that third parties and external observers do not have access to biometric references (and associated identity records)

As mentioned in the previous section, there are generally two main techniques in BC: (1) key generation and (2) key binding. Figure 2.2 shows the differences between key binding and key generation techniques.

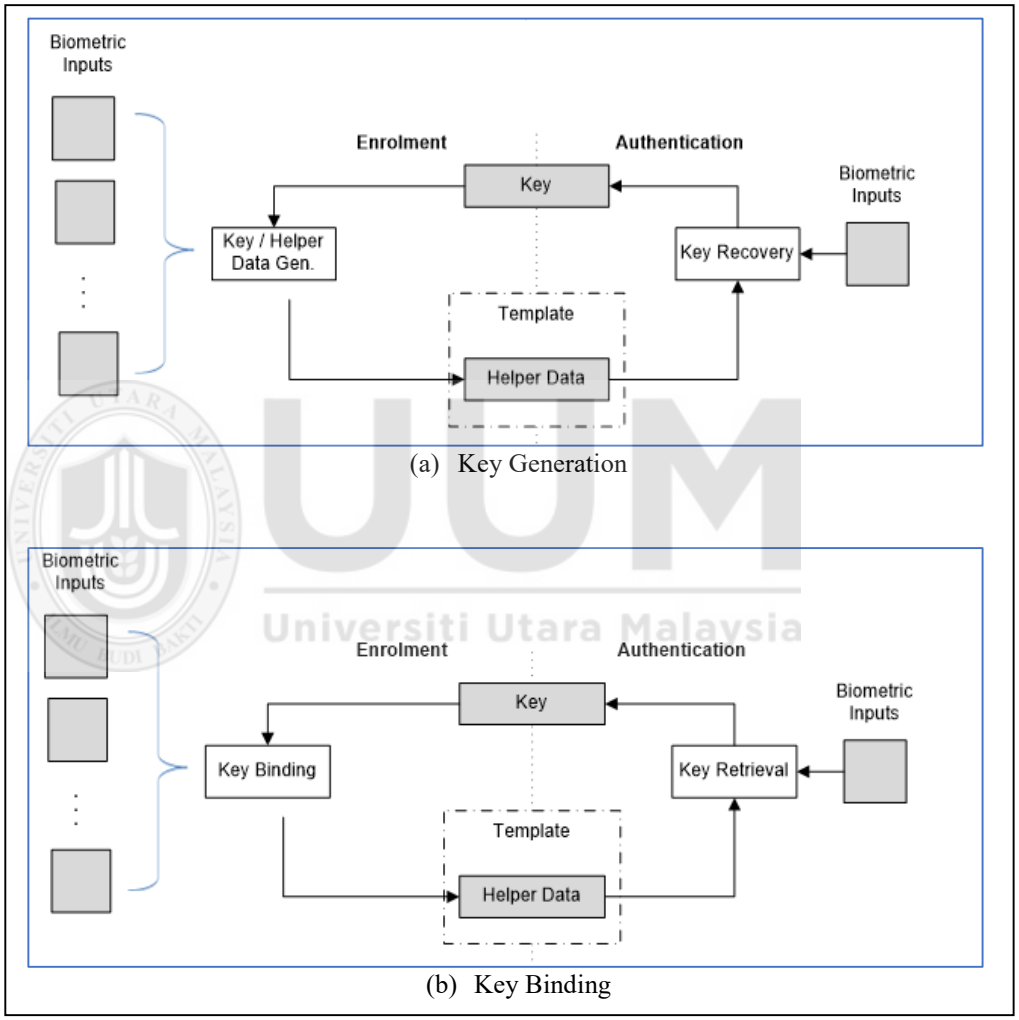


Figure 2.2. Techniques in BC; (a) key generation and (b) key binding (Rathgeb & Uhl, 2011)

The main difference is in the technique itself. In key generation, the cryptographic key is generated right away from various biometric traits, whilst in key binding the helper

data contains both the biometric template and the key. The following subsection describes each of these techniques in detail.

2.3.1 Key Generation Technique

The helper data in the key generation technique is taken solely from the biometric template. The keys are directly produced using helper data and a particular biometric sample. Although not compulsory, most key generation techniques choose to keep the helper data (in the event of a breach, key generation techniques that extract keys without using helper data are not updatable). Key generation technique-based helper data are sometimes known as fuzzy extractors or secure sketches (Müftüoğlu & Yildirim, 2019).

Due to the variability within the user that occurs with different biometric traits, it is challenging to generate the key from various biometric traits. Low discriminability is another issue of the key generation technique. This discriminability may be determined using key permanence and key entropy. Cryptographic key generation using biometric features is straight forward, however it is challenging to attain high key stability and entropy (Sarkar & Singh, 2020).

Two main schemes are used in biometric key generation: (1) secure sketches and (2) fuzzy extractor.

1. **Secure Sketch Scheme**

   The secure sketch comprises two procedures: (1) the sketch procedure takes the enrollment biometric data f^e as input and generates the protected template P as public data, ensuring that it does not divulge excessive information about f^e, and

(2) the recovery procedure reconstructs f^e if the verification biometric sample f^v is indistinguishable from f^e. This reconstructive capability allows it to function as a key (Riaz et al., 2017). The flow of the secure sketch scheme is depicted in Figure 2.3.

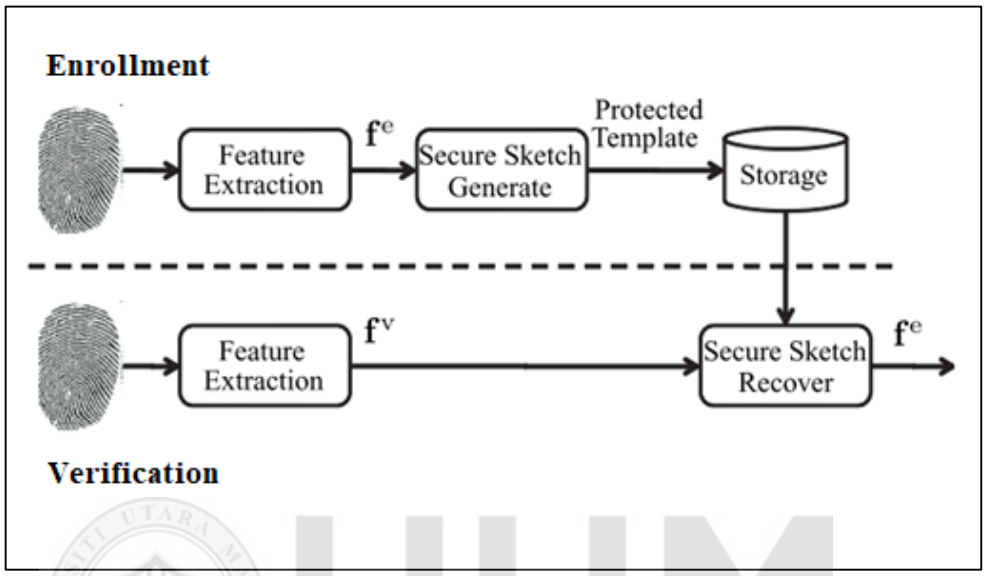


Figure 2.3. The secure sketch scheme (Kelkboom, 2010)

2. Fuzzy Extractor

A fuzzy extractor is comprised of a generate-and-replicate technique. The replicate process generates the open data *P* as a protected template and a key *K* from the enrollment biometric data. If the enrollment and verification samples are comparable, the replicate process returns the same key *K* based on the verification biometric data and the secured template *P*. A fuzzy extractor is essentially formed by collaborating a secure sketch with a key extractor that released a key whose bits are nearly equally random and independent (Lutsenko et al., 2021). Figure 2.4 shows the fuzzy extractor scheme.

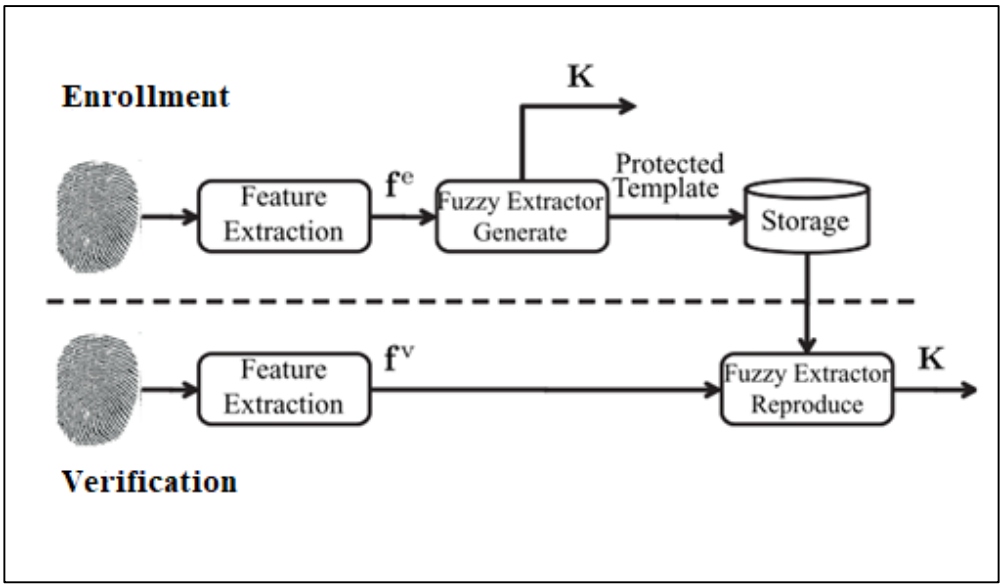


Figure 2.4. The fuzzy extractor (Kelkboom, 2010)

The primary distinction linking the fuzzy extractor and the secure sketch is that the fuzzy extractor seeks to draw out a key whose bits are essentially consistent and self reliant, but the key characteristics in the secure sketch are determined by the enrollment feature vector f^e (Kelkboom, 2010).

In key generation, biometric data are not stored directly, as a result, recovering biometric data from the key string is challenging. Since it cannot keep helper data, the key generation technique cannot offer retrievable keys. Besides, key generation techniques based on helper data that use a secure sketch are vulnerable to attacks over a wide range of registry data. Alternatively, key binding is one of the most successful methods in bio-cryptography and is the equivalent of biometric encryption (Velciu et al., 2014). This justifies the focus of this research, which is to address the technique of key binding.

2.3.2 Key Binding Technique

The key binding technique involves choosing a biometric template and cryptographically embedding a secret key within it. This operation generates a unique element, which is saved in the database as helper data. The helper data provides no relevant information until the individual's biometric data is given.

During the authentication procedure, error-correcting codes are utilized. The request biometrics differ from the stored biometrics by a given degree of error within a set acceptable restriction; the acquired key has the unvarying level of error as the stored biometrics. Error-correcting methods can be used to determine the key (Turakulovich et al., 2018). If the correct key is received, it means that the matching was successful. On the other hand, in key-generating biometric systems, the helper data are generated using a biometric feature template, and subsequently kept in a database. The keys are produced in real time by combining the user's request template with previously saved helper data. While maintaining helper data is optional, it does provide the flexibility to update and revoke keys as needed. If the requested template specified by the user during authentication is the same as the query template used during the enrollment phase, a random string known as a key can be precisely replicated in the same manner (Sarkar, 2018).

The key binding technique generates helper data by binding a specific key to a biometric feature. Correspondingly, the binding, which is a combination of the secret key and the biometric template, is saved as helper data. In the course of authentication, the keys are acquired from the helper data using an appropriate key retrieval procedure (Gilkalaye et al., 2019). Due to its independency on biometric traits, cryptographic

keys are reversible. Nonetheless, reenrollment is required for the purposes of key upgrading to generate fresh helper data (Macek, Franc, Bogdanoski, & Aca, 2018; Rathgeb & Uhl, 2011).

The key-binding technique has two main schemes: (1) fuzzy commitment and (2) fuzzy vault:

1. Fuzzy Commitment Scheme (FCS)

This is theoretically the most straightforward and most studied BC scheme. The biometric template must adhere to a specific length, represented as an organized sequence of bits. A key is associated with an error-correcting code (ECC) codeword of the same length, denoted as n, as the biometric template. The size of the biometric template is determined by the length of the bit sequence. The codeword and template undergo an XOR operation, resulting in a new *n*-bit string that is stored in the helper data alongside the hash value of the key (Chauhan & Sharma, 2019). Since the focus of this research is more on FCS, a detailed discussion is given in the following section (Section 2.4).

2. Fuzzy Vault Scheme (FVS)

Another scheme based on key binding is known as Fuzzy Vault Scheme (FVS). The basic plan is to lock a key *k* with a haphazard set *A*, which results in a vault V_A. In the course of enrollment, a polynomial *p* encodes the key *k*. *A* is extrapolated onto *p*. Chaff points are included to strengthen the genuine points of *p*. If it happens that another set *B* overlaps with *A* during authentication, then the key *k* is reproduced. Figure 2.5 illustrates the FVS.

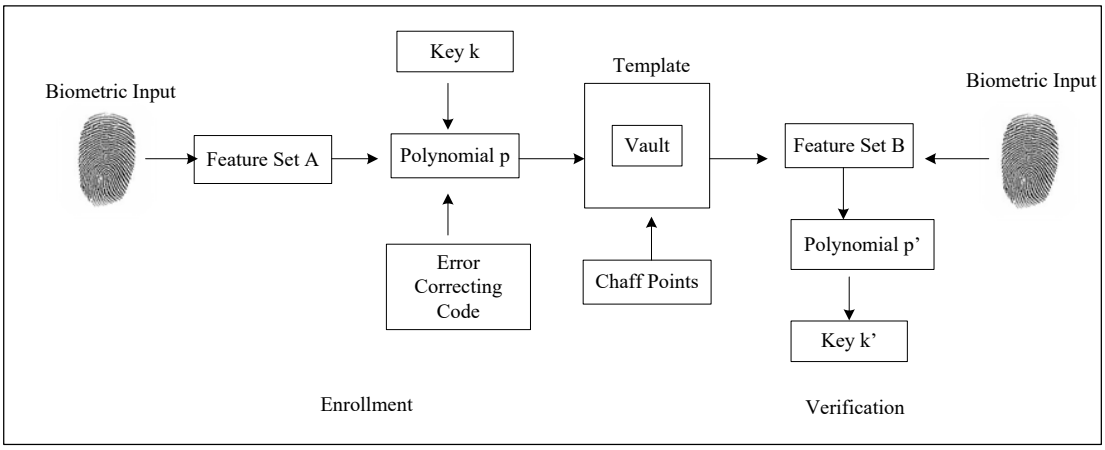


Figure 2.5. The Fuzzy Vault Scheme

The FVS is best suited for arbitrary dimensions unordered data, such as fingerprint minutia. A secret message (i.e., a key) is expressed as polynomial coefficients in a Galois field (GF), for example, GF (216). Unlike other BC methods, the FVS vault preserves true minutia, even if it is concealed among the chaff points. This might lead to various vulnerabilities. A secret minutia permutation controlled by the user's password may be used to increase system security. This 'transform-in-the-middle' approach is similar to the majority of BC methods (Cavoukian & Stoianov, 2015; Sapkal & Deshmukh, 2016).

Recently, a study comparing FVS and FCS was conducted for lightweight sensors using symmetric keys (Zheng, Fang, Orgun, and Shankaran, 2015). The FCS method may seem complicated in terms of feature extraction processes, but the key concealing and revealing method is much less complicated than FVS. Meanwhile, the false acceptance rate (FAR) outperforms FVS, whereas the false rejection rate (FRR) of the two techniques is comparable. Since FVS uses polynomial calculation and reconstruction, FCS is preferable from the viewpoint of computational complexity for lightweight sensors.

2.4 Fuzzy Commitment Scheme (FCS)

Among the schemes that have emerged from current developments in biometric secrecy systems is FCS. The earliest to put forward FCS were Juels and Wattenberg (1999). The basic idea is to allot an arbitrary key to a subject rather than employing biometric data as it is. Figure 2.6 illustrates the existing FCS. FCS consists of several components, such as:

1. Feature Vector Module
2. Bit String Generation
3. Error Correction Codes (ECC)
4. Helper Data (HD)
5. Hash

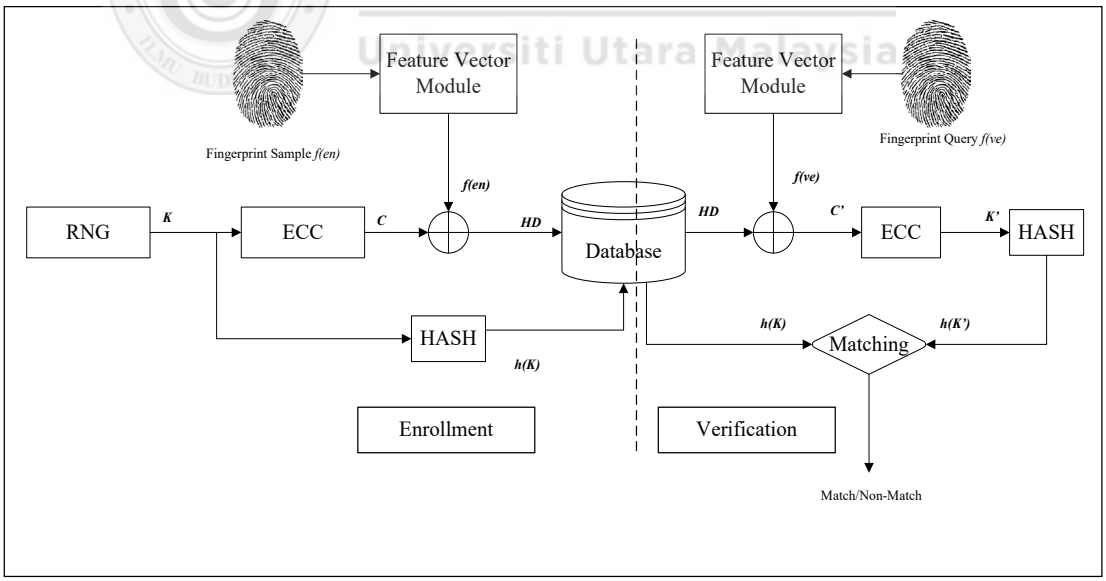


Figure 2.6. Existing Fuzzy Commitment Scheme

According to this scheme, authentication relies on correctly reproducing a secret by utilizing both biometric data and helper data, which compensates for any discrepancies

between the enrolled and queried biometric data. During the enrollment phase, the biometric binary feature vector *f(en)* is combined with an encrypted confidential key *C*. An ECC encoder encodes a randomly generated binary secret key *K* into the codeword *C*. The codeword *C* is then XORed with the binary feature vector *f(en)* to generate the helper data (HD). The database stores the helper data (HD) along with the hash value of the key *h(K).* The flexibility of ECC allows for accommodating minor variations in biometric data. In this scheme, a stored template consists of both the helper data and the hash value of the key, ensuring that no information about the secret key or the feature vector is revealed, thereby theoretically ensuring the security of the FCS (Tantubay & Bharti, 2020).

During the verification phase, the binary feature vector *f(en)* obtained from the requested biometric sample is XORed with the stored helper data HD, resulting in the codeword *C*. This outcome is achieved by decoding the XOR operation using the ECC decoder module and evaluating the key *K′*. The hash value of *K′* is then compared to the hash value of *h(K)* stored in the database. A match is established if *K* and *K′* are found to be identical. In the case of a collision-free hash function, *h(K) = h(K′)* is possible only if *K = K′* (Adamovic et al., 2017). The Hamming distance is utilized to compare the binary feature vectors of the stored database and the query biometric templates, where t represents the error-correcting capacity of the code. This number serves as the threshold for the classifier or comparator (Chauhan & Sharma, 2018; Sarkar & Singh, 2020; Turakulovich et al., 2018).

In developing a safe FCS, the issue of addressing various security threats must be considered. Privacy, secrecy, and unlinkability must be guaranteed by the secure

system. Consider the following scenario: An attacker has access to two distinct templates. Due to unlinkability, an opponent cannot identify whether the templates belong to the same individual or to someone else. Unlinkability is guaranteed by preventing an adversary from performing cross-matching (Chauhan & Sharma, 2019).

Especially in the case of fingerprints, which are usually characterized by an unordered set of minutia, the creation of a practical FCS is not straightforward and requires in-depth knowledge of classification theory, signal processing, information theory, and many other fields. Specialized processing is required to, for example, assess image quality, align input images, remove discrepancies between images, and improve the signal-to-noise ratio (Turakulovich et al., 2018). The subsequent sections present an outline of FCS, and its description follows subsequently.

2.4.1 Feature Vector Module

The first step in the chain is to generate a real-valued feature vector representing the input fingerprint image. It is important that the signal-to-noise (S/N) ratio is sufficiently high so that the following step (bit string generation) results in a robust binary representation. The result of this first stage is the feature vector *f(en)*.

To achieve the best recognition performance, most systems begin a series of processing steps after a fingerprint image is read. The first step is to align the input images to adjust translation and rotation. In the case of fingerprints, these methods are usually derived from the location of the core together with other singularities. The fingerprint image is then processed using (digital) signal processing techniques such as linear and nonlinear refinement to eliminate noise from the image, additional

filtration is required if to assist removal of differentiating features, image alignment, and so on. The actual processing usually depends on the subsequent processing steps in which the distinguishing features are obtained from the image (Lutsenko et al., 2021). There are two important characteristics that a well-functioning feature vector module must have: (1) fingerprint features and sensing, (2) feature extraction techniques.

2.4.1.1 Fingerprint Features and Sensing

Fingerprint features are parameters that can be used to extract information from images of the epidermis of the fingertips (the fingerprint) that can only be associated with a specific individual. Fingerprint sensing is a computational method that uses digital images, such as those created by digitizing ink-rolled or latent fingerprint images, or optical or solid-state scanners, in real time to determine these characteristics.

1. **Fingerprint Features**

   Fingerprint features are the pictorial representation of the external appearance of the epidermis of the fingertip that forms a fingerprint. An interleaved pattern of ridges and valleys is the most prominent structural feature of the fingerprint. In a fingerprint image, the ridges (also called ridge lines) are dark and the valleys (also called valley lines) are light, as shown in Figure 2.7 (Maltoni et al., 2009). The width of the ridges can range from 100 nm to 300 nm, depending on the thickness of the ridge. A ridge/valley cycle has an average period of 500 m (Ali et al., 2016). Finger wounds caused by heat or scratches do not alter the ridge structure in most cases, as it is replicated in any new skin that forms (Van De Haar et al., 2013).

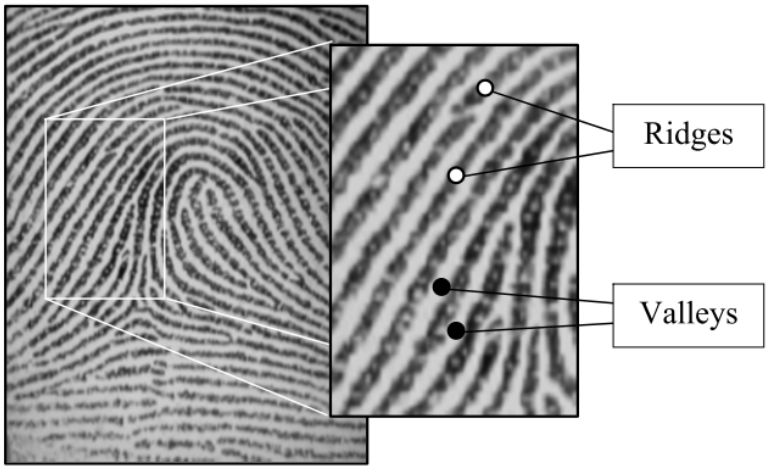


Figure 2.7. Ridges and valleys in a fingerprint image (Dargan & Kumar, 2020)

Generally, fingerprint features are grouped into three levels (Dargan & Kumar, 2020);

a. Level 1 (Global) – refers to the characteristics produced by the singular points as well as the global progression of ridge lines (orientations).
b. Level 2 (Local) – Minute information extracted from the ridge skeleton.
c. Level 3 (Fine-detail) – This level includes ridge features such as width, shape, ridge contours, sweat pores, and wrinkles, in addition to ridge contours.

Ridges are frequently seen to be uniformly parallel at the global level (Level 1), but they also contain one or more locations where they take on unique features (characterized by a high degree of curvature and, frequent ridge terminations) (Marasco & Ross, 2014).

The second level of analysis, known as the local level, or Level 2, allows for the detection of additional significant details, or minutia, in fingerprint patterns. In the context of fingerprinting, they are defined as minute features. In this case, they

refer to the numerous ways ridges can be discontinuous, as depicted in Figure 2.8. For example, a ridge may end abruptly (ridge ending) or split into two ridges (ridge division/bifurcation). When it comes to automated fingerprint matching, minutia are the most commonly used characteristics (Jothi, 2018).

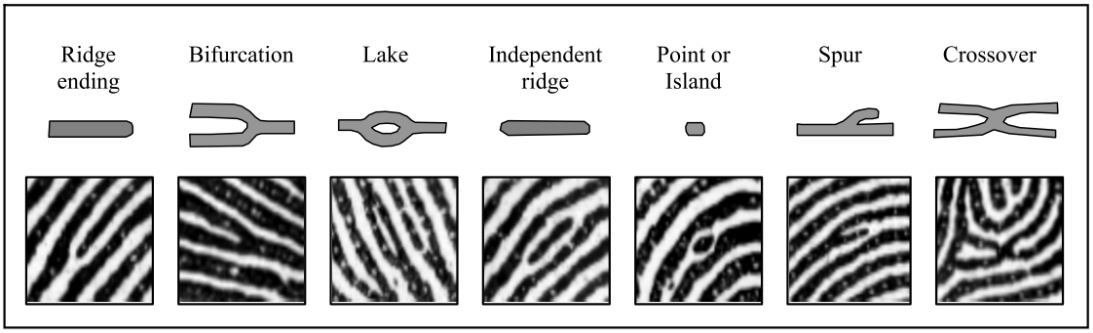


Figure 2.8. Seven most common types of minutia (Jothi, 2018)

A direct physical analog (e.g., singularities or minutia) is typically found in features obtained from fingerprint images. However, there are also situations where these features are not directly associated with physical characteristics (e.g., local orientation images or filter responses). Depending on the situation, the features can be utilized for matching or as an intermediate step in deriving other features. Several preprocessing and enhancement processes, for example, are frequently conducted to make minutia extraction task more manageable (Mirza, 2014).

Additional small information in the fingerprint pattern can be recovered at the fine-detail level (Level 3). These comprise all dimensional ridge properties such as width, shape, edge contour, and pores (Figure 2.9a), newly formed ridges (Figure 2.9b), ruptures, wrinkles, and blemishes (Figure 2.9c). Each ridge of the epidermis (outer skin) has pores (or sweat pores) that run the entirety of it and is connected to

the dermis (inner skin) by a two arrays of peg-like protrusion called papillae (Sarier, 2016). Although Level 3 traits are very unique and critical to latent fingerprint examiners, they are currently used by relatively few automated matching systems because spotting them calls for high-resolution fingerprint scanners (e.g., 1000 dpi) and high-quality fingerprint images (Hasan & Abdul-Kareem, 2013).

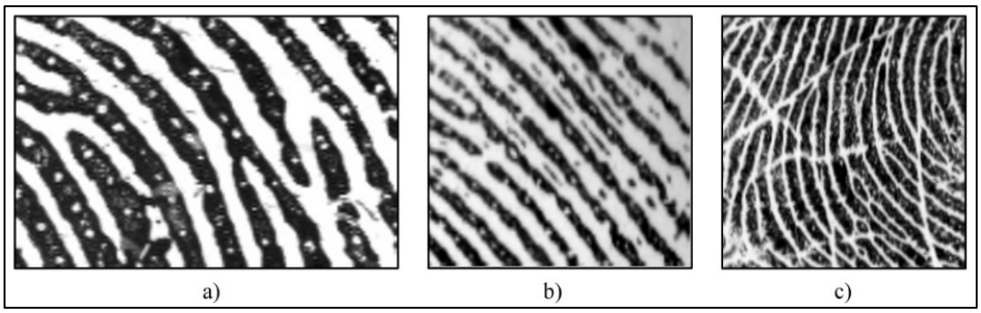


Figure 2.9. A section of a fingerprint captured at 1000 dpi, with visible pores. (a) The width and shape of the ridge have localized unevenness, and the ridge contours are not uniform; (b) incipient ridges are ridges that are not fully formed and can occur in the middle of normal ridges; they are frequently cracked and do not contain pores; and (c) some folds can be seen in some parts of a fingerprint (Galar et al., 2015).

Only the first level (with some exceptions) is used to categorize fingerprints, since classes of fingerprints are logically created based on global features. Nevertheless, Level 2 and 3 characteristics are typically employed for fingerprint matching since they allow the identification of a fingerprint's uniqueness. Consequently, the majority of fingerprint extraction (FE) techniques focus on utilizing Level 1 features, which are intricately connected to fingerprint orientations and singular points (SPs), for classification purposes (Dargan & Kumar, 2020). An orientation map (OM), that represents the fingerprint's local ridge flow, records fingerprint orientations. SPs are places in the fingerprint that show the largest variation in ridge orientation, i.e., the location of frequent abruptness of ridge variation (Aithal & Prasad, 2017). SPs are classified into two types: cores and deltas (Babatunde, 2015; Dargan & Kumar,

2020). Figure 2.10 depicts a fingerprint image (a), its OM (b), and the SPs in both images.

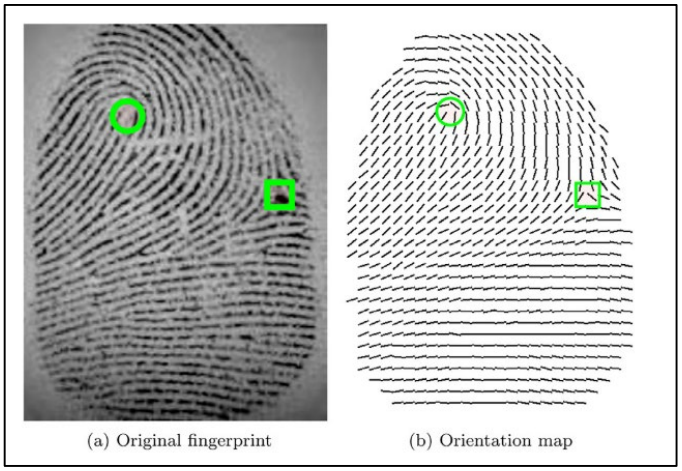
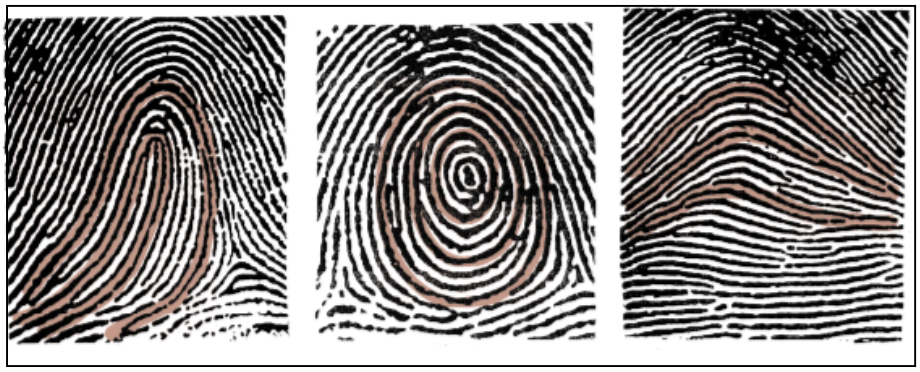


Figure 2.10. OM and SPs are two global fingerprint traits. The core point is denoted by a circle, whereas the delta point is denoted by a square (Yager & Amin, 2004)

After outlining the primary characteristics of fingerprints used for categorization, the following are brief descriptions of each fingerprint class. For example, the fingerprint pattern that results when an inked finger is placed on paper is composed of the friction ridges on that particular finger. Friction ridge patterns are classified as loops, whorls, and arches. Each of these patterns has its own expression due to the form and interconnection of the ridges (Aithal & Prasad, 2017). Figure 2.11 shows the types of friction ridge patterns.



(a) Loop (b) Whorl (c) Arch

Figure 2.11. Examples of loop, whorl, and arch patterns (Vats et al., 2016)

1. Loop: Fingerprints that have a single core, a single delta (below the core), plus a minimum of one ridge that begins at the right or left side, loop backwards, and leave on the same side.
2. Whorl: Fingerprints with two cores and two deltas, with at least one ridge circling entirely around the fingerprint's center.
3. Arch: Fingerprints flow from one side to the other when there are no SPs or ridges, generating a little bulge.

2. Fingerprint Sensing

Various technologies can be used to get fingerprints. The image is taken by a camera once the finger is put on a transparent prism in optical sensors. Ridges and valleys are contrasted in Total Internal Reflection (TIR) sensors. When light enters the prism from one side, it is thrown back at the valleys and absorbed at the ridges because the ridges are in touch with a glass platen (Maltoni, 2005). Sensors built on this automation are susceptible to being deceived caused by the use of materials that have a light reflectance similar to skin. Furthermore, optical devices from different manufacturers often have physical differences (e.g., lenses). As a result,

the detection rate of fake fingerprints varies from device to device. Devices that utilize microprisms implanted in thin plastic, in particular, are resistant to spoof attacks (Memon et al., 2008).

In resistive devices, the finger is used to represent the top electrode of a capacitor and a metal plate is used to represent the bottom electrode. When the finger is positioned on the sensor, the new capacitance reading between valleys and ridges can be recorded due to the deviation in capacitive values between skin–sensor and air–sensor contact. Soft fake fingerprints made of gelatin are sensitive to capacitive sensors (Suzuki et al., 2014). For thermal sensors, the thumb is positioned on pyroelectric stuff in thermal sensors that converts temperature changes into voltage (Kapoor & Sharma, 2016). The temperature changes when the ridges come into contact with the sensing material, while the temperature does not change in the valleys that are contactless with the substance. The signal disappears the moment heat equilibrium between the thumb and chip is achieved (Marasco & Ross, 2014).

Ultrasonic sensors are used to magnify the acoustic impedance error between the skin of the ridges and the air in the valleys. The reflected signal is picked up by a receiver when acoustic waves are sent to the fingertip surface. Substances with similar sound characteristics as the thumbs are passed through the scanner, a sensor which is highly sensitive to false thumbprints (Marasco & Ross, 2014).

2.4.1.2 Feature Extraction Techniques

The aim of feature extraction is to minimize feature counts in a data file by generating a new one from existing ones. This new, compressed collection of the features is expected to qualify in describing the majority of details in the existing set of features.

By merging with the existing set, a summarized description of the existing features can then be constructed. Feature extraction techniques were classified into three categories: (1) orientation image, (2) singular points, and (3) filter responses.

1. Orientation Maps (OM)

The OM is extracted by constructing a depiction of the course of the localized ridge for each of the blocks in the fingerprint. Referring to previous Figure of 2.10 (b) (page 35) present a model an OM, whereby for a known image with "*N* x *M* pixels and given that orientation blocks of *n* x *m*, an OM is a matrix of *N* / *n* x *M* / *m*, where angles are usually stored in radians in the range [0, π] or [-π / 2, π / 2]". The block size need to be sufficient to produce a decent approximation of the localized ridge course, yet not too large, as variances in localized directions must be recorded so that it can adequately determine the universal properties of the fingerprints (Dargan & Kumar, 2020).

The significance of OM extraction is a reality as it employed virtually all fingerprint procedures. For example, SP recognition techniques evaluate the behavior of orientations; similarly, OM can be used to determine ridge structure (Nilsson & Bigun, 2003).

2. Singular Points (SPs)

SPs are spots in the fingerprint where the ridges fluctuate at a higher frequency, indicating that the ridge curvature is greater than typical (Figure 2.10). These spots are significant since they decide the fingerprint's topological structure and class. As was raised earlier, SPs are classified into two categories (Yager & Amin, 2004a):

a. Core: The uppermost point of the innermost ridge of the loop, i.e., the point where the ridges meet.
b. Delta: The triangular-shaped pattern formed when ridge flow diverges, i.e., when three distinct orientations of ridges meet.

3. Filter Responses

Some feature extraction techniques rely on the fingerprint image's reaction to various purifying processes. Responses after filtration in each pixel normally depend on the local orientation of the image's ridges and valleys, and so give important information for classification. Gabor filters are a method developed by Jain et al. that stands out from the others in this category (Jain et al., 2000). This approach is widely used and has gained widespread acceptance.

Gabor filters have the capacity to eliminate noise, maintain actual ridge and valley patterns, and deliver information held in a specific orientation. The Gabor filter process begins by reading an image (i.e., fingerprint). The area of interest was then calculated as the position around the core point. The predetermined area is filtered in eight distinct directions using a bank of 2D Gabor filters (eight orientations are necessary to secure the localized ridge features in a fingerprint, but only four orientations are needed to secure the universal configuration). The Average Absolute Deviation (AAD) derived using the average of gray values in different parts of the filtrated image is used to specify the feature vector or fingerprint. Fingerprint image verification is constructed from the Euclidean Distance (ED) between two fingerprints (Azzoubi & Ibrahim, 2015; Jain et al., 2000).

The advantage of Gabor filters is the greater flexibility in defining the functional form due to a wider range of parameters (degrees of freedom). Gabor filters can represent numerous image areas more effectively than other extraction methods due to their greater flexibility in parameter selection (Moreno et al., 2005).

The majority of fingerprint recognition techniques rely on minutia features, often known as ridge endings or ridge bifurcations. Most fingers contain between 30 and 40 minutia, and each person is thought to have a unique set of minutia locations. A fingerprint can be well represented using minutia as a (unordered) list of minutia locations (Segun et al., 2020).

Unfortunately, these unordered sets and inconsistent minutia points (ridge termination and ridge bifurcation) are difficult to combine with the FCS because they produce a fixed-size binary representation (Shukla & Patel, 2021). For this reason, methods have been developed to convert minutia sets into feature vectors. For example, Al-Assam et al. (2009), Gilkalaye et al. (2019) present a method that generates a feature vector that is independent of the order of minutia locations and the measurements of the feature vector is fixed and does not depend on the number of minutia.

In the past decades, there have also been a lot of advancements in the fields of recognition model and image processing. This has led researchers to treat fingerprints as a pattern of ridges rather than a set of minutia locations for several reasons. In many cases, patterns can be described as vectors (Brindha, 2012; Hasan & Abdul-Kareem, 2013). Combining the feature vector generated from the minutia information with the

feature vector from the pattern information, a real-valued feature vector *f(en)* representing the fingerprint image is obtained (Zhou et al., 2012).

2.4.2 Bit String Generation

As mentioned above, the FCS requires biometric measurement in the character of a binary string. The bit string generation block converts the feature vector *f(en)* into a binary string *X* that can be classified using a Hamming distance classifier.

In the previous section, it was shown that fingerprints can be symbolized as feature vectors *f(en)*. In order to achieve good classification results when combined with FCS, these real-valued feature vectors must be converted into binary strings such that binary strings derived from similar input images have a low Hamming distance. The bits in the binary string should ideally be statistically independent (or at least uncorrelated). Special quantizers are used to convert the data into bit strings (BSI, 2011). After all separate features have been converted into a (short) bit string, all bit strings can be pooled to procure a binary string representation *X* of the feature vector *f*. In more advanced schemes, the quantity of bits extracted for each feature may be different. By assigning additional bits to good features and less bits to bad features, the overall accuracy of the systems can be improved (Dwivedi et al., 2020).

2.4.3 Error-Correcting Codes (ECC)

Error-correcting codes (ECCs) are often used to correct errors in messages transmitted across rowdy transmission networks. ECCs can be described as a collection of codewords, denoted as C. Each codeword, represented by $c \in C$, corresponds to an *n*-

bit sequence in which the *k*-bit messages, denoted as $m \in M\ (n > k)$, are mapped prior to transmission. The parity bits, consisting of (*n*-*k*) bits, are utilized to recover the transmitted codeword from a corrupted received codeword. If the error-correcting capability is denoted as *t*, it implies that *c* can rectify up to t errors provided that the minimum distance between any two codewords in C is at least 2t + 1 (Teoh & Kim, 2015). One of the most important aspects of a BC scheme is the selection of an ECC. The ECC ought to act noise blocker from the biometric data while being safe, i.e. without revealing particulars information to an adversary (Laban & Drutarovsky, 2021). Some of the most commonly used ECCs in BC systems are Reed-Solomon codes, Hadamard codes, Convolutional code, and Bose-Chaudhuri-Hocquenghem (BCH) codes.

2.4.3.1 Reed-Solomon Code

In 1960, Irving S. Reed and Gustave Solomon invented the Reed-Solomon code, a set of ECCs (Reed & Solomon, 1960). The Reed-Solomon code operates on a set of data represented as a series of symbols with finite fields. Reed-Solomon codes is able to spot as well amend numerous errors. A Reed-Solomon code is capable of detecting (although incorrectly) any combination of a maximum of *t* incorrect symbols. It can also locate and rectify a maximum of *t/2* incorrect symbols in hidden locations, by adding $t = n - k$ checked symbols to the data (Reed & Solomon, 1960). It can rectify a maximum of *t* deletion in recognized and named spots as deleted codes, or it able to identify and rectify combinations errors and deletions as combination codes. Reed-Solomon codes are well-suited for correcting multiple-burst bit errors, as a sequence of $b + 1$ consecutive bit errors can impact only two symbols of size *b*. The choice of *t*, which represents the error-correcting capability, is determined by the code's designer

and can be selected from a wide range of possibilities (Reed & Solomon, 1960; Sklar, 2020).

2.4.3.2 Hadamard Codes

The Hadamard code, named for Jacques Hadamard, is an ECC used to detect and fix errors when delivering messages across noisy or unstable channels. The Hadamard code is a 2^m long linear code that operates on a binary alphabet. Unfortunately, there is considerable ambiguity in this statement because some sources assume a message length of $k=m$ and others assume $k=m+1$. The first example is referred to as the Hadamard code, whereas the second example is referred to as the enhanced Hadamard code (Malek, 2006).

2.4.3.3 Convolutional Code

Convolutional code is an ECC that produce parity symbols by sliding a Boolean polynomial function over a data stream. The sliding application reflects the encoder's 'convolution' across the data, thus the term 'convolutional coding.' Because of the sliding nature of convolutional codes, trellis decoding with a time invariant trellis is possible. Using time invariant trellis decoding, convolutional codes may be optimum soft choice decoded with manageable complexity (Verlinde, 2003).

Convolutional codes have a number of benefits, including the capacity to carry out cost-effective maximum probability soft choice decoding. This contrasts with conventional block codes, which are often denoted by a time-dependent trellis and are hence tough-choice decoded. Frequently, the base coding rate and encoder depth of convolutional codes are used to identify them (or memory) [n,k,K]. The base coding rate, often denoted as n/k, represents the ratio between the raw input data rate (n) and

the data rate of the output channel encoded stream (*k*). Due to the addition of redundancy by channel coding, the value of n is smaller than *k*. The memory, commonly known as the 'constraint length' *K*, determines the output based on both the current input and the previous *K-1* inputs. The depth can also be expressed as the number of memory elements *v* in the polynomial or as the maximum number of encoder states (typically: 2^v) (MacKay, 2005; Yan et al., 2017).

2.4.3.4 Bose Chaudhuri and Hocquenghem (BCH)

Bose–Chaudhuri–Hocquenghem codes, commonly known as BCH codes, are a type of cyclic error-correcting code (ECC) constructed using polynomials over a finite field, also referred to as a Galois field. These codes were independently developed by the French mathematician Alexis Hocquenghem in 1959 and by Raj Bose and D. K. Ray-Chaudhuri in 1960 (Bose & Ray-Chaudhuri, 1960; Hocquenghem, 1959).

The most essential attribute of BCH codes is the capability to precisely manage the amount of symbol errors that the code is able to rectify during code generation. In particular, it is practical to develop binary BCH codes that can correct multiple bit faults (Teoh & Kim, 2015). Besides that, BCH also has the feature of easy decoding, achieved by an algebraic method called syndrome decoding. This facilitates the usage of decoder design for these codes and permits the use of small, low-power electrical equipment (Pellikaan & Wu, 2012).

2.4.4 Helper Data *(HD)*

The biometric template contains a cryptographically encoded secret key. The resultant single object is saved as helper data in the database. Unless the user's biometric data is

given, the cryptic helper data provide no information. As a result, the helper data (HD) should ideally contain no details of incoming real-valued feature vector or biometric data.

In Section 2.4, a general overview of FCS was presented in a diagram (refer to Figure 2.6). For simplicity, a binary ECC is used (operating on the Galois field GF (2)) and thus XOR operations between bit strings are computed. In the case of a symbol-based code, the XOR would be replaced by (symbol-wise) addition and subtraction in the finite field (de Groot et al., 2016).

During enrollment, an arbitrary number *K* is generated. Employing any cryptographic hash function (such as SHA-1, SHA-256, and MD5) a hash value *h(K)* is obtained and stored. Furthermore, *k* is passed through the encoder *(ENC)* of an ECC to obtain the codeword *C*. Finally, the biometric feature vector *f(en)=X*, depicted as a binary string obtained by the feature extraction module, is combined with *C* by an XOR operation to obtain helper data $HD=C \oplus X$, which is also stored. Thus, the load of FCS is *X* and the output consists of *(HD, h(K))*.

During verification, a new biometric feature vector *f(ve)=Y* is combined with *HD* to obtain a candidate codeword $C'=HD \oplus Y=C \oplus (X \oplus Y)$. This candidate codeword *C'* is sent through an ECC decoder *DEC* to obtain a candidate secret *K'*. Finally, the hash value of *K'* is matched to *h(K)* and if they are identical, an Accept message is generated indicating that X and Y were generated from the same biometric.

Helper data is used in BC to create keys from biometric features. Helper data is a data recorded in a database that contains both the pair, biometric template and the key. However, there appears to be no clue regarding the key or the biometric template in this helper data. Decrypting the key without knowing the user's biometric data is mathematically difficult. Both the helper data and the ECC are linked with the biometric template. If the query biometric differs from the stored template within a specific restricted inaccuracy, the associated codeword can be retrieved with an appropriate amount of error, which can then be decoded to recover the correct codeword, and a successful match is returned upon successful key restoration (Sarkar & Singh, 2020).

2.4.5 Hash

In the FCS, the hashed fingerprint information is the actual reference information. A comparison in FCS only leads to an Accept if the hash value generated during enrollment and the hash value generated during verification are identical.

The purpose of the hash value in FCS is to prevent access to the secret *K*. Knowledge of *K* (and *HD*) reveals *X* as $X=ENC(K) \oplus W$. Even partial knowledge of *K* could reveal information about *X*. Therefore, the hash function must hide information. Although theoretically no deterministic function can provide this property, it is generally believed that strong cryptographic hash functions hide information sufficiently for practical purposes. First, instead of choosing *k* as the only argument of the hash function, the argument of the hash consists of the concatenation of *K* and *HD.* Thus, instead of storing *h(K)*, the value *h (K |HD)* is stored.

Two important properties of hash functions are single direction and impact resistance. In the current setting, single direction is required to prevent access to *K* and is related to the privacy of biometric information. However, hash collision could induce an incorrect acceptance and thus has an impact on the safety of the biometric system. By shortening the output of the hash value, the collision resistance decreases while the single direction increases. For example, if the shortening is such that the length of the hash output is $|h(K)|<|K|$, the uncertainty in *K* is $|K|-|h(K)|$ bits, since there are on average $2^{|K|-|h(K)|}$ strings *K* that map to the same hash value. Thus, hash truncation provides a tradeoff between privacy and security. Moreover, it is worth nothing that in a practical situation, hash truncation does not alter the overall accuracy of the system unless the collision probability approaches the biometric false acceptance probability (FAR) (Premasathian, 2013).

A hash function is a function that is employed to convert any size of data into a fixed-size value. A hash value is the output of a hash function. A frequent use for the values is indexing a fixed-size table known as a hash table. Hashing, also known as scatter storage addressing, is the method of determining the index of a hash table using a hash function (Kumar et al., 2010).

In data storage and recovery applications, hash functions and their related hash tables are used to process data in a least and nearly sustained amount of time per retrieval. They use a fragment of the storage space needed for the data or records. Hashing is a way to access data that is easy to compute and takes up little storage space. It circumvent the inconsistent ingress time of classed and unsorted index and structured trees, as well as the potentially exponential storage needs of direct access to state spaces with large or variable-length keys (Pittalia, 2019).

A cryptographic hash function (CHF) is a mathematical process for converting any data into a definite-size bit array. It is an irreversible function, thus inverting or reversing the computation is nearly impossible. In an ideal environment, the one solution to locate a message that creates a certain hash is to utilize a rainbow table of matching hashes or to execute a brute-force search on all conceivable load to check whether they yield a worth comparing. Cryptographic hash functions are a key component of modern cryptography (Al-Kuwari et al., 2010).

There are several cryptographic hash algorithms; this section includes a few algorithms which are well known, such as (1) MD5, (2) SHA-1, (3) SHA-2, (4) SHA-3, and (5) Whirlpool.

2.4.5.1 MD5

MD5 was developed by Ronald Rivest in 1991 to restore an older hash algorithm, MD4. RFC 1321 was assigned to it in 1992. Collisions with MD5 may be computed in seconds, rendering the technique inappropriate for the majority of use cases requiring a cryptographic hash. MD5 produces a 128-bit digest (16 bytes) (Rivest, 1992).

2.4.5.2 SHA-1

SHA-1 was created as part of the Capstone project of the United States Government. The initial specification of the algorithm, now referred to as SHA-0, was released in 1993 as part of the United States government standards agency NIST's Secure Hash Standard, FIPS PUB 180. However, shortly after publication, the US National Security Agency (NSA) withdrew SHA-0, replacing it with an updated version released in 1995

known as SHA-1 in FIPS PUB 180-1. However, the SHA-1 algorithm has been deemed vulnerable to the shattered attack, which can lead to collisions, rendering the hash function compromised. Consequently, SHA-1 should be considered as broken. SHA-1 generates a hash digest consisting of 160 bits (20 bytes) (Maetouq et al., 2018; Stevens et al., 2017).

2.4.5.3 SHA-2

The National Security Agency (NSA) developed the Secure Hash Algorithm 2 (SHA-2), which is a collection of cryptographic hash functions first introduced in 2001. These functions are designed using the Merkle-Damgard structure, which relies on a one-way compression function built using the Davies-Meyer structure. The Davies-Meyer structure, in turn, employs a specialized block cipher (whose details are classified) (Pittalia, 2019).

2.4.5.4 SHA-3

On August 5, 2015, the National Institute of Standards and Technology (NIST) released the Secure Hash Algorithm 3 (SHA-3) (National Institute of Standards and Technology, 2015). SHA-3 belongs to the Keccak family of cryptographic primitives, with the Keccak algorithm being developed by Guido Bertoni, Joan Daemen, Michael Peeters, and Gilles Van Assche (Bertoni et al., 2013). Keccak is built using a sponge framework which is also involved in creating other cryptographic primitives like a stream cipher. The output sizes of SHA-3 are the same as those of SHA-2: 224, 256, 384, and 512 bits. The SHAKE-128 and SHAKE-256 functions also provide configurable output sizes. The -128 and -256 in the name refer to the function's security level instead of the output size in bits (Chang et al., 2012; Yalçin & Kavun, 2013).

2.4.5.5 Whirlpool

In 2000, Vincent Rijmen and Paulo Barreto introduced Whirlpool, a cryptographic hash function. Whirlpool is based on a significantly enhanced version of the Advanced Encryption Standard (AES). It produces a 512-bit hash digest (64 bytes) (Barreto, 2008; Pittalia, 2019; Shirai & Shibutani, 2003). An overview of the hash algorithms can be found in Table 2.2.

Table 2.2

Hash algorithms comparisons (Pittalia, 2019)

Parameters	**MD5**	**SHA-1**	**SHA-2**	**SHA-3**	**Whirlpool**
"Block size (bits)	512	512	512, 1024	1600-2*bits	512
Digest size (bits)	128	160	160, 224, 256, 384, 512	160, 224, 256, 384, 512	512
Word size (bits)	32	32	32, 64	64	8
Rounds	4	80	80	24	10
Collision found	Yes	Theoretical Attack	None	None	Yes
Operations	AND, OR, XOR, ROT	AND, OR, XOR, ROT	AND, OR, XOR, ROT, SHR	AND, OR, XOR, ROT, SHR	AND, OR, XOR, ROT"

2.5 Privacy Leakage in Fuzzy Commitment Scheme

The assumption is that the helper data is communal and hence must not at all contain information about the secret. As a result, leakage of information should be minimal. A biometric system's crucial characteristics are (1) the dimensions of the secret key as well as (2) the particulars included in the helper data about the biometric observation. The last-mentioned parameter, when disclosed, is referred to as privacy leakage. Preferably, the privacy leakage is supposed to be trivial so that an individual's biometric data are not compromised (Dwivedi et al., 2020; Laban & Drutarovsky, 2021). Some of the attacks can be done in the traditional FCS, for example surreptitious key-inversion (SKI) attack and attacks via record multiplicity (ARM) (Scheirer & Boult, 2007). Figure 2.12 shows how SKI attacks work.

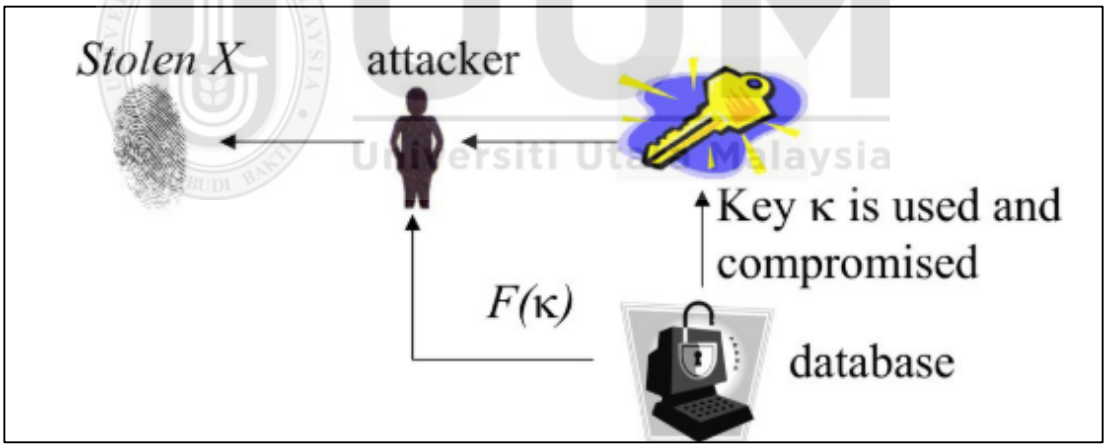


Figure 2.12. The SKI attack (Scheirer & Boult, 2007)

Based on Figure 2.12, the agreed goal of the FCS is to unloose a hidden key, the encoded data *F(K),* as well as the intercepted secret *K*. An attacker may decode the biometric template data X by determining the values associated with *K* if they know *K*. If an attacker knows the cryptographic key, the biometric string that combined with the codeword is simply reconstructed using the agreed cryptographic key and the

secure sketch. As a result, privacy leakage is unavoidable (Jin et al., 2016; Scheirer & Boult, 2007).

Simoens et al. (2009) introduced cross-matching against FCS based on a decodability attack which makes the most of the link between numerous helper data generated from the similar theme biometric data. Kelkboom et al. (2011) significantly investigated the attack and presented a bit-permutation strategy to counter the decodability attack. The work of Cavoukian et al. (2008) inspired such an attack with the intention of determining if decoding two pieces of helper data results in a valid codeword. If the case of 'yes' answer, the two pieces of helper data are convincingly from the aforesaid user. This technique is sometimes termed as attacks via record multiplicity (ARM) (Chauhan & Sharma, 2018; Jin et al., 2016). Figure 2.13 shows how the ARM attacks work.

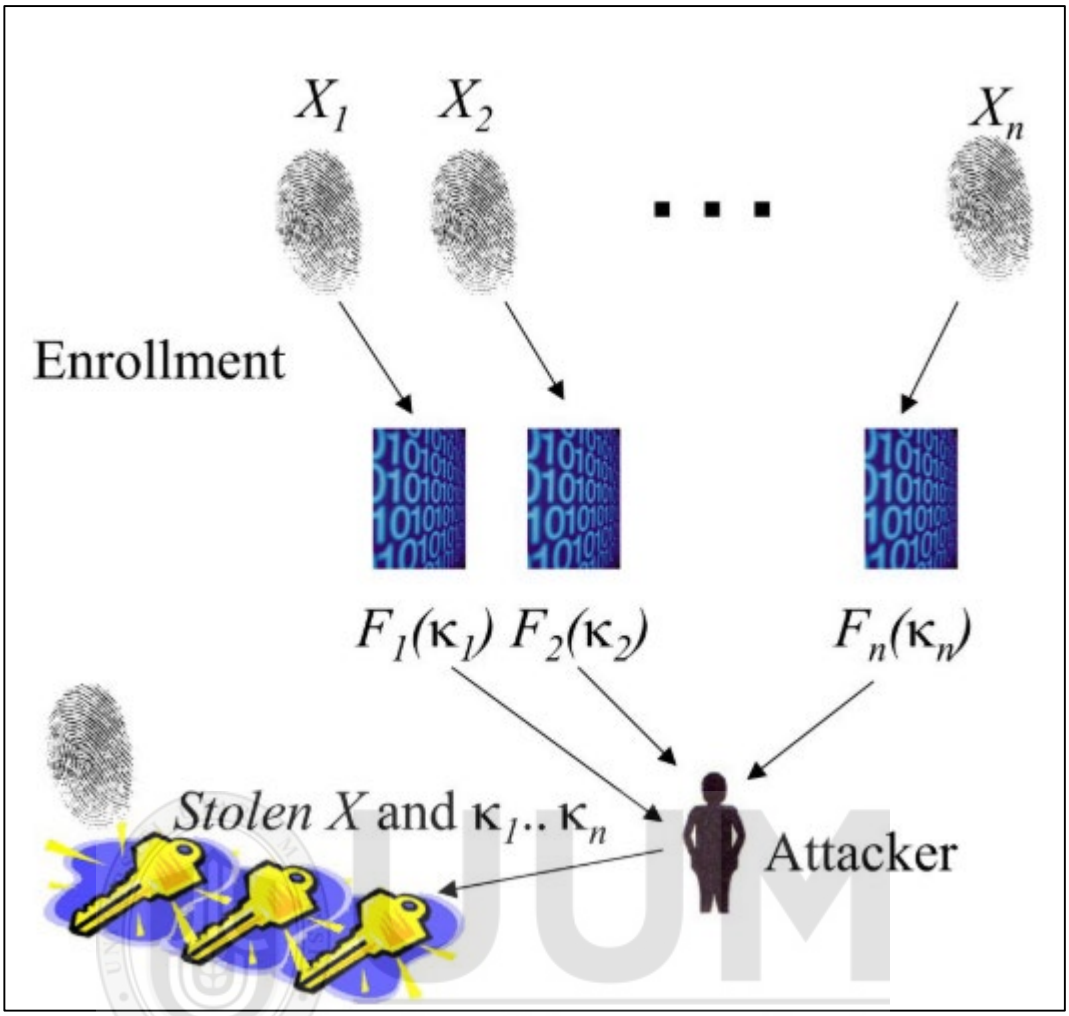


Figure 2.13. Attack via record multiplicity (ARM). An attacker gathers various enrollment templates, is able to merge the data and at least link the records, and in the worst case scenario can obtain template X and secret K (Scheirer & Boult, 2007)

2.5.1 Dependency on Biometric Features

Dependency on biometric features significantly reduces security and privacy and increases privacy leakage. This happens because the bits drawn out from biometric features are not uniformly independently distributed (Zhou et al., 2012). First, the real-valued biometric features need to be binarized. Then, the feature extraction module takes binary features as input. This causes the binary features to take dependency on the real-valued features, which lessens the security of the system. An ideal binarization process should maintain the potential acceptance of biometric systems from one angle,

and on the one angle, it should be able to generate uniformly and independently distributed (UID) binary features and adjust the distributions of biometric features for security reasons (Riaz et al., 2017).

Helper data in FCS might disclose information on biometric traits. The helper data in a completely secure system should not give any information about the secret. Only in this circumstance can the magnitude of the secret be used to determine security. Many FCS implementations make the assumption that the characteristics are UID without thoroughly evaluating the actual distribution of biometric features, or they make the assumption that genuine security is just slightly lower than the size of the secret (Riaz et al., 2017; Zhou, 2012).

2.5.2 Effect on Privacy Leakage: Reversible Attack

It is now widely accepted that it is possible to reconstruct an artificial sample that is identical to a real sample from an unprotected template. This process of reverse engineering, known as inverse biometrics, raises a major warning to biometric systems in two ways. Firstly, it enables the extraction of delicate personal data such as biometric data from compromised exposed templates. Secondly, the compromised exposed templates are utilized to produce artificial specimens, and this is referred to as reversible attacks. These synthetic samples can then be used to start masquerade attacks (i.e., impersonate a subject) (Gomez-barrero & Galbally, 2019). Figure 2.14 shows the reconstructed sample that results from synthetic sample generation (Gomez-barrero & Galbally, 2019).

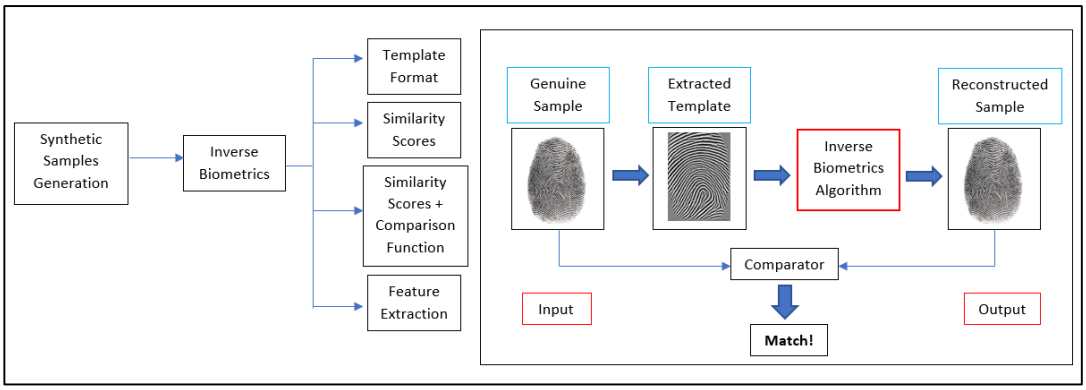


Figure 2.14. Inverse biometric method for synthetic sample generation

The basic purpose of inverse biometric techniques is to construct an artificial biometric sample which will be confidently matched to the sample that created it, starting with an apparently safe portrayal of the subject's biometric features (i.e., the biometric template). As a result, these tactics supply the attacker with important biometric data that they did not have formerly (i.e., the biometric sample). The reverse-engineered samples can then be utilized to imitate a certain subject and launch masquerade or presentation attacks (Gomez-barrero & Galbally, 2019; Yang et al., 2015). Figure 2.15 shows an case where a compromised leakage template is employed to regenerate the fingerprint that leads to other threats.

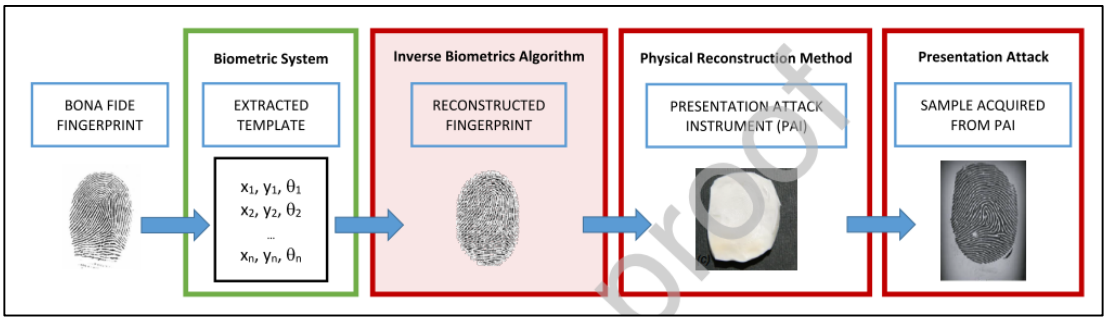


Figure 2.15. An inversion method is used to recreate a biometric sample from a compromised template, that can cause various types of greater danger like presentation attacks (Gomez-barrero & Galbally, 2019).

2.6 Privacy Requirements

The previous section introduced the concept of privacy in terms of helper data not revealing information about biometric templates. This concept gives recommendations for securing biometric templates in accordance with different security, integrity, availability, and renewability or revocability criteria. The standard specifically suggests the following privacy rules for biometric information (Yasuda et al., 2016):

1) *Irreversibility*: Preventing the application of biometric data for a purpose apart from initially designed. In other words, it shows the complexity of retrieving biometric features.
2) *Unlinkability*: Making it impossible to link stored biometric references across software or databases.
3) *Security*: Defining how hard it is to get accepted by the system illegally.
4) *Data minimization*: Reducing uncalled-for and/or unwanted processing of private data, such as when authenticating a person's identification.

The standard does not specify procedures for meeting these requirements, but it is applicable to a wide range of approaches that extend far beyond BC techniques, such as conventional template encryption. Although the standard does not explain in detail the difference between 'irreversibility', 'security', and 'data minimization', it can act as a backing when assessing the privacy of practical systems. For the purposes of this research, 'security' is interpreted to mean that outside observers copying biometric references should not be able to use the biometric information. When using FCS, this means 'irreversibility'. The requirement of 'data minimization' is defined in the context where biometric data and biographic data are combined, and this is considered outside the scope of this research. In summary, in the context of BC systems, 'irreversibility'

and 'security' are the most important requirements to be assessed (Nandakumar & Jain, 2015).

Apart from that, the renewability criteria are predicated on the ability to create several protected templates from a single biometric sample. The renewability criteria are determined by the quantity of various keys that may be drawn on in the binding operations; hence key size is also important. To summarize, two critical characteristics to study are (1) key size and (2) the type and quantity of information leakage from the protected template that influences the irreversibility criteria.

2.6.1 Maximizing the Key Size

The previous section discussed how the size of the key affects the template defence system's irreversibility and renewability. Assuming that the key's bits are evenly random and independent, the key's size is an indicator of its entropy. As a result, by increasing the key size, the irreversibility and renewability criteria may be improved.

Previous research by Chauhan and Sharma (2019) and Zhou et al. (2011) demonstrates that irreversibility grows with security. As the amount of secrets rises, irreversibility increases, and privacy leakage decreases for layings with the same feature size and codeword length. This demonstrates how a large secret size may increase irreversibility and prevent privacy leaks.

Furthermore, the secret-key length (also known as the secret-key rate) should be long in order to reduce the likelihood of guessing the secret key and providing illegal access (Jin et al., 2016).

2.6.2 Tradeoff Between Performance, Privacy and Security

The effort of reversing the key binding technique is also determined by FCS classification performance. System performance is demonstrated in terms of the false acceptance rate (FAR) and the false rejection rate (FRR). The FAR is the likelihood that two separate persons' biometric samples may be misclassified as comparable and authentic, resulting in a false match. As a result, the FAR also shows the possibility of locating a random biometric sample from an existing database, resulting in a counterpart and hence a security breach, also known as a FAR attack.

On the contrary, the FRR is the probability of a false non-match. It is also known that an increase in FRR normally leads to a drop in FAR, and therefore a possible rise in key size. In addition, the performance is improved by the acquisition of multiple biometric samples. As a result, FRR together with the sample count impact the key size, indicating a commutation between template protection system privacy and security (Al-Assam & Jassim, 2012; Riccio, Manzo, et al., 2016).

Apart from that, research by Kelkboom et al. (2012) evaluating the achievement and key size of template protection schemes based on various ECC implementations, databases, biometric modalities, or feature extraction algorithms are challenging to comprehend. Varying ECC execution can result in inconsistent error-correcting capabilities and, as a result, a probable variation in system performance and key size. The standard of the derived characteristics, and hence the system performance, is influenced by inconsistent databases, biometric modalities, or feature extraction techniques. The authors endeavored to minimize these disparities in the preceding

comparisons. They conclude that there does not appear to be a clear link between system performance and key size based on the large variances discovered between claimed system performance (particularly the FAR).

2.7 Related Works

Several significant literatures related to biometric fingerprint encryption techniques, grouped as follows.

Priya et al. (2014) created biometric and binding keys by combining biometric templates and random number keys. FCS is used to achieve this combination. They also used randomness tests to check the unpredictability of the created mixed key. The results of the experiment clearly reveal that the combined key is 50% more random than the biometric key. However, they do not provide any information about the accuracy and performance of fingerprint recognition.

Another related work on key binding scheme is by Jin et al. (2016), who put forward a new key binding scheme which is free of ECC and works together with cancelable transforms for minutia-based fingerprint biometrics. The ECC is used to mitigate biometric variations within a user in FCS, and it has some drawbacks, such as the tradeoff between security and performance. This is because the greater key size, tighter security is consistently countered by low level of key release success. Their experiment shows that the precision level is close to the latest product. Nevertheless, their proposed method may lack data integrity.

A previous study by Chauhan and Sharma (2019) proposed a technique that uses two additional matrices to secure the helper data from the previous FCS. The feature vectors along with the encoded key are encrypted using the invertible matrix. Finally, the permutation vector permutes the encrypted data to the extent that an intruder would find it very puzzling to derive relevant biometric information about the user from the data. They evaluate their proposed approach on an iris dataset.

To boost the security and privacy of biometric templates, Yasuda et al. (2016) proposed an extension of FCS that is both more secure and more private using palm veins. They use two-layer error-correcting codes and have developed a new technique that is able to provide practical performance in biometric authentication applications. Since their enhanced approach does not reveal biometric information even if matching is successful, it is befitting for remote authentication over public networks, enabling secure matching over an untrusted server (e.g., the cloud).

Wang et al. (2014) proposed a unique technique to protect multibiometric templates based on FCS and chaotic systems, as well as a security analysis approach for unimodal biometric leakage. The verification performance is reduced but the security of the multibiometric template is improved. Apart from that, Zhou et al. (2011) presented a detailed security and privacy evaluation of FCS. In the prevailing protection system for 3D face recognition, the criteria representing the requirements in real applications are explored and quantified. Owing to the reliance on to the dependency on biometric features, the evaluation results showed that the security decreased significantly, and the privacy leakage increased.

Sandhya et al. (2020) use FCS to propose a fingerprint BC with cancelability. A Delaunay triangulation net generated from fingerprint minutia is used to convert fingerprint minutia. These altered characteristics are then encrypted with convolutional coding. The Viterbi technique is used to retrieve the codeword during the decoding phase. The proposed method was experimentally evaluated using FVC 2002 database. For FVC 2002 DB1, DB2, and DB3, the EER achieved using the suggested technique is 1.66%, 1.89%, and 6.87%, respectively. A detailed comparison between this technique and the proposed approach is analyzed later in Chapter 3.

2.8 Summary

This chapter has highlighted the relevant literature that has been addressed to further elaborate the elements that form the body of knowledge for this research based on previous works. It includes the discussion of biometric fingerprinting systems, biometric template protection techniques, and biometric cryptosystems. This chapter then narrowed the discussion to key binding techniques with a focus on FCS and highlighted the elaboration of the research problem that calls for solutions.

In addition to the discussion, existing work was also presented to justify the purpose of the research. Towards the end of this chapter, further insights on the significance of the problems to be solved were highlighted as the chapter terminates and it was pointed out that a proposed approach is urgently needed to resolve the dependency issues of the helper data in FCS.

By setting the conducted literature review as the base, it should be emphasized that the dependency on biometric features in helper data is caused by the fact that the bits

extracted from biometric features are not uniformly independently distributed. The fingerprint features reflect the local properties of the attributes represented by the features. This is because real-valued biometric features must be binarized. Binarization is an important component of the helper data structure. From a security perspective, the binarized features should be provided consistently and independently. This explanation answers the research objective number one.

In the following chapter, the methodology that was used during the research is explained in more detail. It presents a stronger perception of the 'how' of the research, which includes the research activities that were conducted to achieve the research aims and objectives.

CHAPTER THREE

RESEARCH METHODOLOGY

3.1 Introduction

This chapter discusses the research methodology that was used during the study. In order to facilitate the discussion, a research framework was developed and used as a roadmap to guide the entire research process. The introduction of this research framework is provided in the following section, namely Section 3.2. Since the research was divided into four major phases, separate subsections were allocated to explain each of these phases, spanning from Subsection 3.2.1 to 3.2.4. Finally, the chapter ends with Section 3.3 to summarize the entire research processes conducted.

3.2 The Research Framework

Currently, BC is of great interest for fingerprint template protection schemes. As mentioned in the previous chapter, FCS is one of the key binding schemes that protects biometric fingerprint template from various attacks. Due to its simple and effective solution for biometric security, FCS is one of the well evaluated schemes in recent years.

Given the increasing interest in biometric template protection, especially in FCS, this research was conducted in the hope of solving the problem of dependency on helper data that compromises user privacy.

Therefore, to conduct this research, several phases were developed to strategically plan the research activities and achieve the previously established objectives. Figure 3.1

shows the research framework that includes the phases, research activities, and outcomes, respectively.

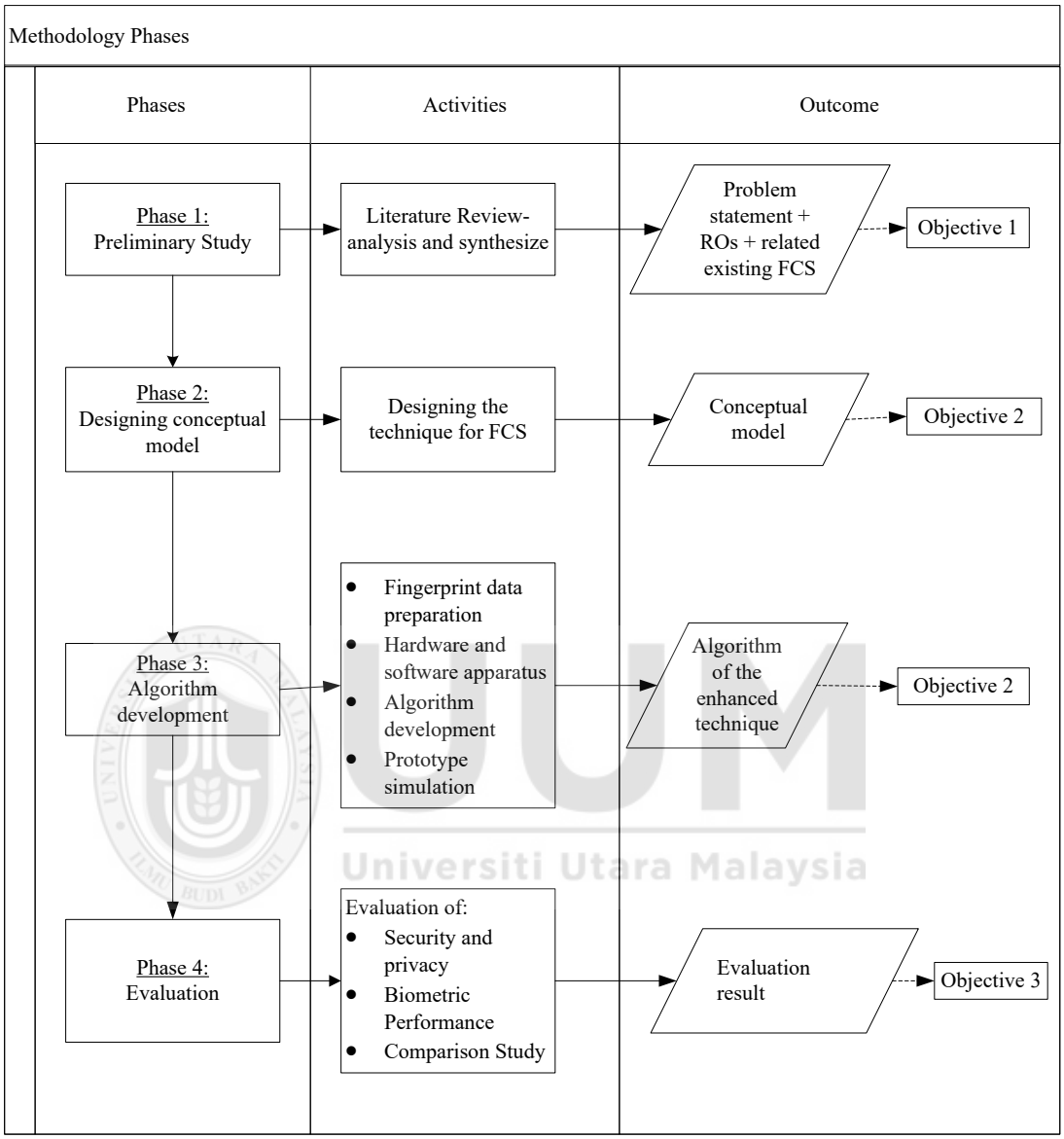


Figure 3.1. The research framework

3.2.1 Phase 1 – Research Review and Analysis

The first phase involved the research review and analysis, which was essentially conducted in an iterative mode. The iterative process included two main activities, namely (1) literature review and (2) literature analysis. These two activities are summarized as follows:

1. **Literature review:** In this research activity, the relevant literature was gathered and reviewed to ensure a comprehensive understanding of the domain area. It also ensured that the problem or gap in the literature was identified, and interesting research questions were posed for further investigation.
2. **Literature analysis:** Literature analysis was conducted in parallel with the literature review. This was to ensure that the previous works of other researchers were thoroughly analyzed in order to gain a deeper understanding of the domain area. The outcomes of this activity are summarized as follows:

    a. There are many works on key binding techniques, especially FCS, which have been proposed to improve the security and performance of existing FCS using various techniques, but few focused on helper data.
    b. Helper data in existing FCS is not secure because it depends on biometric features that could reveal important data, which could lead to privacy leakage and cross-matching attacks.
    c. Securing biometric features in the first place before XORing to generate helper data can help reduce privacy leakage.

Based on the key points above, this study proposed to enhance the existing FCS scheme by introducing a concealment module to reduce the dependency on helper data to improve security and privacy. With the completion of this phase, the first objective of this research was achieved. The next phase is to translate the proposed idea into an appropriate design.

3.2.2 Phase 2 – Designing the Conceptual Model

As discussed in Section 2.4, the existing FCS uses a key binding method that links an encoded key with biometric features. The biometric features collected from the biometric sample are XORed with the encoded data (codeword *C*) during the enrollment phase. A linear ECC *(n, k)* with an error correction capacity of *t* bits is used to encode the random key *Key* of length k into a codeword *C*. Thus, the *HD* (helper data) is obtained from the codeword *C* and the biometric features via the XOR technique.

HD is then stored in the biometric database along with the hash value of the key *h(Key)*. As a result, the biometric template consists of two pieces of information *(h(Key), HD).* The requested biometric data is XORed with the stored helper data during the matching step. The XOR operation generates the codeword *C'*, which has a length of *n.* The key *Key'* is obtained by decoding the obtained codeword *C'* using the ECC decoder. Finally, the stored key hash is compared with the hash of the decoded key *K'*. If *h(Key)= h(Key'),* the comparator module returns a match.

The design of the conceptual model was based on the analysis performed in Phase 1. It was found that the current FCS has several limitations, namely straightforward binarization and low entropy.

Based on these findings, the concealment module was proposed to strengthen the protection of biometric data. The additional keys could help to extend the exhaustive search. This will reduce the possibility of attackers successfully attempting to obtain

information about the user's biometric data. Figure 3.2 illustrates the conceptual model of the proposed approach.

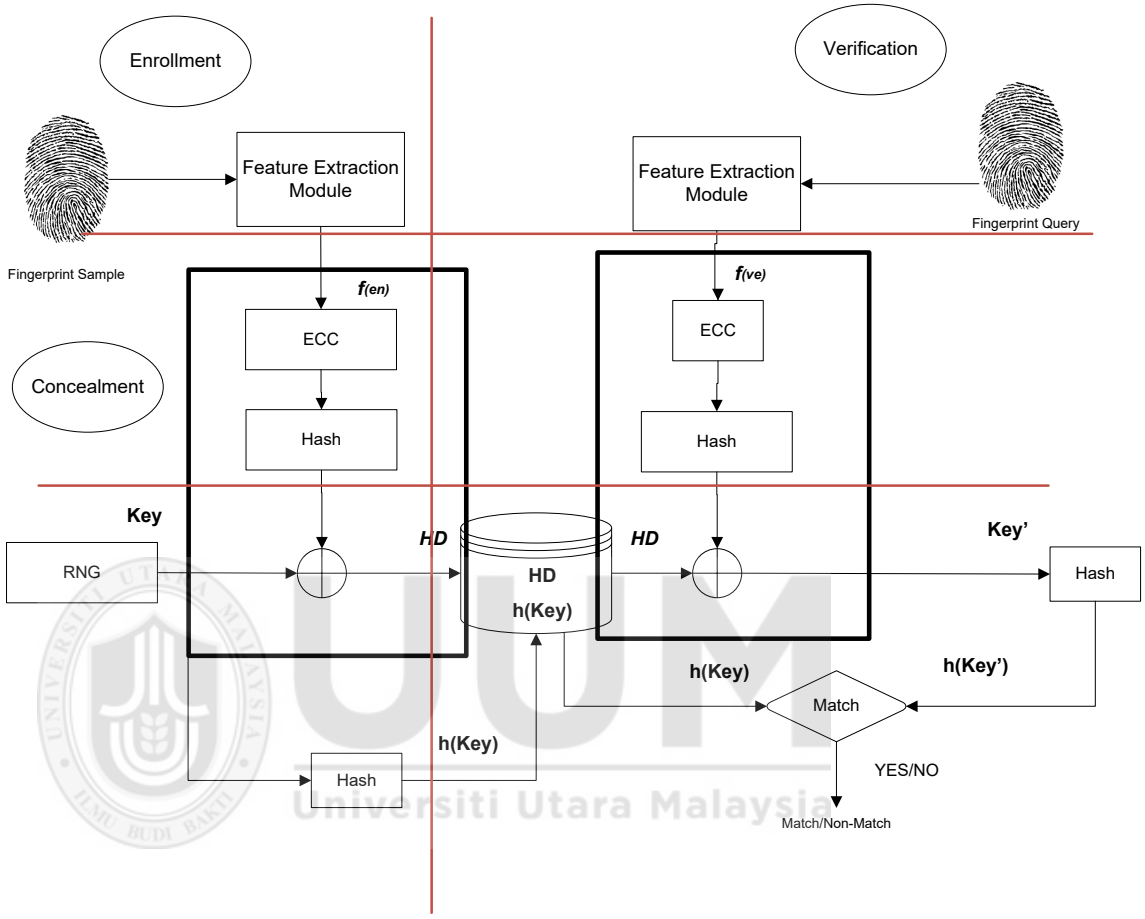


Figure 3.2. The conceptual model of the enhanced FCS

Based on the conceptual model shown in Figure 3.2, the two bold boxes (inside the red line region) illustrate the enhancement of the processes involved in the proposed approach. The proposed approach consists of three modules (shown in ovals): (1) enrollment (2) concealment (3) and verification. The modules are summarized in Table 3.1.

Table 3.1

The main modules in the enhanced FCS

Enrollment Phase		Verification Phase	
(1) Enrollment	This module was adopted from the existing FCS. It extracts the feature vectors from the enrolled biometric samples using Gabor filter, a feature extraction algorithm.	(3) Verification	This module is indirectly affected by the modification introduced in the enhanced FCS. This module calculates the Euclidean distance between the corresponding feature templates to find the nearest distance and decodes it with the ECC and is then hashed with SHA-256.
(2) Concealment	This module is proposed in the enhanced FCS, where the binary string of the feature extraction template is then encoded into a convolutional error-correcting code and then hashed using the cryptographic hash function SHA-256.		

The outcome of Phase 2 was the conceptual model, which was used to conceptually illustrate the idea of the proposed approach. The research then moves to the next phase to develop and realize the conceptual design of the proposed approach.

3.2.3 Phase 3 – Algorithm Development

Based on the outcome of Phase 2, this phase continues with the development stage. Several procedures were performed to develop the enhanced FCS. The procedures included:

1. fingerprint data collection
2. hardware and software apparatus
3. algorithm design and development
4. prototype simulation

Each of these procedures is discussed in more detail in the following subsections:

3.2.3.1 Fingerprint Data Preparation

The biometric fingerprint dataset was obtained from the Fingerprint Verification Competition (FVC) year 2002 (FVC2002, 2002) an international competition that focused on fingerprint verification software assessment. The objective of FVC 2002 is to track current advances in fingerprint verification in both the scholarly and commercial fields. In addition, the dataset is also a well-known efficient standard for fingerprint technology.

The FVC2002 consists of four datasets, whereby three of which contain three real fingerprint templates and one of which contains synthetic templates. Each dataset contains 100 fingers and 8 samples per finger (800 fingerprints in total). Each dataset was collected using different sensors. For example, Dataset 1 and Dataset 2 were collected using optical sensors, but from different brands. Meanwhile, Dataset 3 was collected using a capacitive sensor and dataset 4 was collected using synthetically generated fingerprint. For this study, fingerprint templates acquired with optical

sensors are used, i.e., Dataset 1. This is because this type of fingerprint template has been used in many previous studies due to its simplicity and reliability.

3.2.3.2 Hardware and Software Apparatus

This study used the "MATLAB R2017a environment on a machine with Intel (R) Core (TM) i7 ~2.50GHz CPU and 8-GB RAM". The proposed approach was developed using MATLAB Programming Language.

3.2.3.3 Algorithm Design and Development

As mentioned in Section 3.2, the primary goal of this research is to propose an concealment module based on the key binding technique in FCS to enhance the security and privacy aspects of fingerprint template protection. Thus, a suitable algorithm needs to be designed and developed to make the required modification. The steps are described below.

1. *Algorithm design*: In this step, the design of the proposed approach was carried out in three major phases, namely (1) Enrollment (2) Concealment and (3) Verification.

   The first module is the enrollment, which involves scanning the fingerprint to obtain a raw digital representation. A feature extractor was then used to build a feature template in the form of a binary string.

   The second module is concealment. The feature template was first encoded with an error-correcting code (ECC) to generate the *fingercode.* A hash value was

generated from *fingercode*, which was computed using the SHA-256, a hash function *h(fingercode*). An XOR operation was performed between the randomly generated *key* and *h(fingercode*) to generate helper data. Again, a hash value was generated from *key* is computed using SHA-256, *h(key*). The *h(key)* is stored with the helper data.

The third module performed the verification procedure, which verified the user's fingerprint based on the Euclidean distance. For each input fingerprint and the query template fingerprint, the system computed the match score for the features based on the closest distance by the ECC encoder and the cryptographic hash function. This was validated by comparing the corrected hash function value of *h(Key')* with the stored *h(Key)* to determine whether the verification was successful or not. If the user was valid, access was granted.

2. *Algorithm implementation*: In order to verify and measure the accuracy of the proposed approach, each algorithm design was followed by its implementation. The validation and measurement procedures were performed mathematically so that the accuracy of the generated keys can be checked and verified. The accuracy of the encryption and decryption keys is the most essential aspect in the design of cryptographic algorithms. Therefore, the algorithm design of the proposed approach must be verified by actual implementations and numerical tests.

3.2.3.4 Prototype for Simulation Purposes

Once the algorithm was designed and developed, the prototype was constructed using MATLAB software to simulate the proposed approach. This was to evaluate the proposed approach. The interfaces of the prototype are attached in Appendix A while the source code in Appendix B.

At the end of this phase, the proposed approach should be modified and standardized accordingly. The completion of Phases 2 and 3 indicated that objectives 2 and 3 of this research were achieved. The next phase is further elaborated in the following section.

3.2.4 Phase 4 – Evaluation

The development of the proposed approach was described in the previous section. The use of evaluation methods is required in order to quantify the third objective of this research. This part presents potential evaluation methods and explains what they signify in terms of security and privacy and their roles in the evaluation procedures.

Evaluation procedures are considered as steps that are developed to check and ensure that something proposed (i.e., techniques) meets specific requirements and provides functionality to users. These procedures are very important to confirm that the technique works correctly for its intended purpose, to ensure that no unintended tasks are executed, and to determine its quality and accuracy (Dasso & Funes, 2007; Wallace & Fujii, 1989). The technique does not have to be completely free of errors, but its intended use is sufficient and acceptable (Alves et al., 2011).

In this last phase, the performance of the proposed approach was evaluated. It is a fundamental requirement to evaluate the accuracy of the result in a computational environment. Therefore, this process justifies the performance of the proposed approach. The evaluation was computed in the MATLAB environment.

The evaluation of the proposed approach included two main aspects: (1) security and privacy (irreversibility and privacy leakage), and (2) biometric performance. In the following subsections, these two aspects are discussed in more detail.

3.2.4.1 Security and Privacy

The proposed approach was evaluated in terms of analysis methods for adversarial attacks from various scenarios that may be directed against FCS. There are two evaluation criteria for the proposed approach, namely (1) security and (2) privacy. First, the security aspect was measured by the difficulty of finding out the secret (key) from a secure template. Therefore, the security analysis can be analyzed based on the stored information, i.e., helper data and the hash key.

Second, the privacy protection capability consisted of two aspects, irreversibility and privacy leakage. Irreversibility indicates how complicated it is to retrieve the biometric features or difficult it is to guess the true pre-image. On the other hand, privacy leakage indicates how much information about biometric data is included in a secure template.

Apart from that, privacy leakage can cause other security issues, such as linkability or cross-matching attacks. This is because two fingerprint biometric templates are considered to be fully linkable if there exist some methods to determine whether they

come from the same biometric entity. Therefore, it is important to analyze privacy protection. Privacy protection can be measured by unlinkability property. The unlinkability property ensures that it is difficult for an adversary to determine whether the templates belong to the same person or to a different person. Unlinkability can be determined by analyzing the cross-matching attack. By preventing an adversary from performing a cross-matching attack, unlinkability ensures privacy protection.

3.2.4.2 Biometric Performance

False acceptance rate (FAR) and false rejection rate (FRR) are two of the most important parameters to consider when assessing biometric systems. FAR is the proportion of impostors that are accepted by the biometric system, whereas FRR is the proportion of genuine individuals that are falsely rejected. These measures are valuable in characterizing the accuracy of fingerprint technology. Below are the components for calculating FAR and FRR (Precise Biometrics AB, 2014):

1. FAR – the proportion of fraud attempts that are falsely declared as matching a template from another object.
2. FRR – the proportion of true attempts that are falsely declared as not matching a template of the same object.
3. True positive rate (TPR) – the proportion where the test result is positive for a known positive condition.
4. Equal error rate (EER) – the point at which the proportion of falsely accepted attempts is equal to the proportion of falsely rejected attempts (FRR = FAR).
5. Accuracy – the proportion of true results (either true positives or true negatives) to the total attempts.

Given*, "FP: False Positive, FN: False Negative, TN: True Negative, and TP: True Positive*

FRR is calculated as the proportion of positive results that fall below the threshold.

FRR = genuine scores that exceed the threshold/all genuine scores

genuine scores that exceed the threshold = FN

all genuine scores = TP+FN

Therefore, **FRR = FNR = FN/(TP+FN)** (3.1)

FAR is calculated as the proportion of negative scores that exceed the threshold.

FAR = imposter scores that exceed the threshold/all imposter scores.

imposter scores that exceed the threshold = FP

all imposter scores = FP+TN

Therefore, **FAR = FPR = FP/(FP+TN)** (3.2)

TPR is the proportion where the test result is positive for a known positive condition. The equation to calculate the TPR is:

TPR = TP / TN+FP = 1-FRR **(**3.3)

The calculation of the accuracy of the proposed approach is the proportion of true results that are either true positive or true negative to the total results. The equation to calculate the accuracy is:

Accuracy = TP+TN / TP+TN+FP+FN" **(**3.4)

As the FAR decreases, FRR increases, and vice versa. Figure 3.3 shows the percentages of FAR, FRR, and EER compared to sensitivity and the effects on each

other. The equal error rate (EER) is the value at which the lines cross each other. This is the point where the percentage of false acceptance and false rejections are equal (Giot et al., 2013).

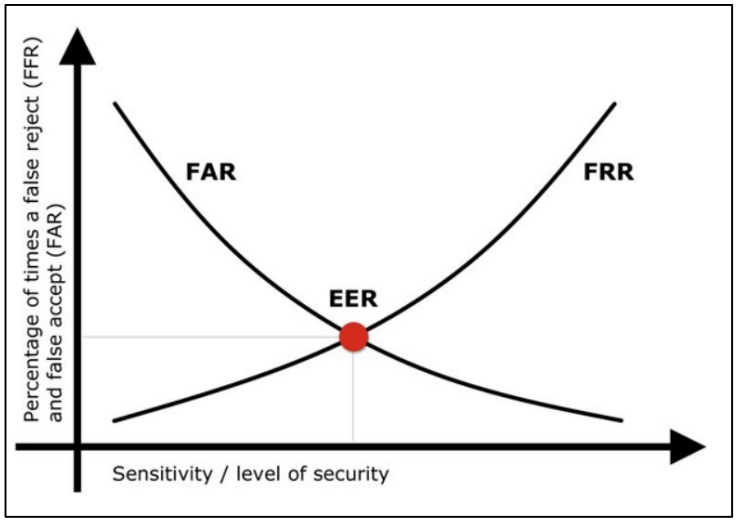


Figure 3.3. Percentage of FAR, FRR and EER versus sensitivity (Recogtech, 2022)

FRR is likely to rise sharply if FAR is reduced to its lowest possible level. In other words, more security for access control makes it less convenient, as the system rejects users who are not false. The same holds true in reverse. If the FRR is reduced to improve user convenience, the system will most likely be less secure (higher FAR) (Recogtech, 2022).

FAR and FRR can usually be configured by adjusting the corresponding threshold to be more or less stringent. The threshold has a 'score' that ranges from 0 to 1. 0 indicates no match at all, while a 1 indicates a complete match. As a result, the threshold is set between 0 and 1 for real-time systems. From the above information, it can be concluded that this leads to a more secure system (but less user-friendly or less powerful) or a less secure system (but more user-friendly or more performance) (Yasuda et al., 2016).

3.2.2.2 Comparison Study

A comparison study was selected for this research, namely the work of Sandhya et al. (2020). The overall flow of the approach proposed by Sandhya et al. (2020) is shown in Figure 3.4. It includes the following steps:

1. Construction of a Delaunay triangulation from minutia points. A fingerprint image representing minutia points and a Delaunay triangulation formed from these points.
2. Use of the Delaunay triangulation to find the neighbors for each tiny point.
3. Conversion of the neighbors into a three-dimensional array and generation of a bit string.
4. Encodement of the bit string with convolutional coding. A codeword *c* is chosen from random error-correcting codes. A key is generated at random and encoded with the convolutional code, say *c*. Between b and c, an XOR operation is performed, resulting in an encrypted template *e*. A hash value is generated from *c* using the SHA-1, say *h(c).* This *h(c)* is stored alongside *e* and is referred to as helper data. This is known as the encoding phase.
5. Matching. During decoding, a query image gives the binary feature vector b^l. b^l and $'e$ are XORed. The hash of c^l is calculated. The hashes are matched to decide if *e* should be accepted or rejected and decrypted.

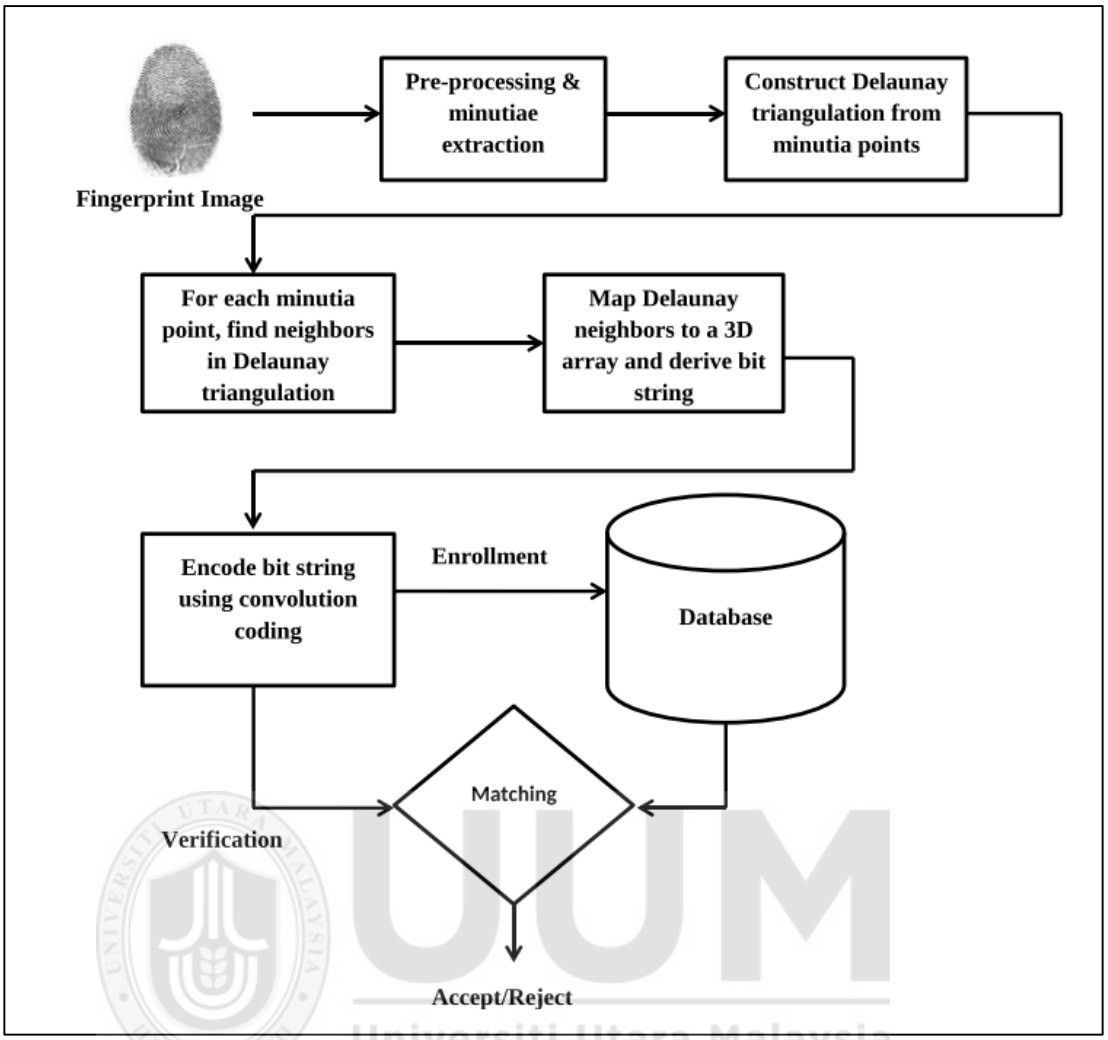


Figure 3.4. Flow diagram of the approach proposed by Sandhya et al. (2020)

Based on Figure 3.4, using a Delaunay triangulation net, the calculated transformed features are generated from the minutia of the fingerprint. Then, a bio-cryptosystem was created that employs convolutional coding to generate a codeword for use in FCS. The Viterbi algorithm was used to decode the codeword, ensuring the system's security and accuracy. They also compared the hash value of the encoded and decoded codewords using the cryptographic hash algorithm SHA-1. Their experimental evaluation assesses the needs of biometric template protection methods, including diversity, changeability, irreversibility, and accuracy.

3.3 Summary

This chapter was intended to highlight all research activities that were conducted in four main phases throughout the research. The research framework was used as the basis for explaining the phases with their activities and outcomes. Each research activity was explained individually in separate sections of this chapter to show what procedures were carried out to achieve the research objectives. This chapter also explains the proposed approach conceptually and presents various parameters used in the evaluation procedures. In the next chapter, the proposed approach is further discussed in terms of the design and development procedures, while the evaluation of the approach is presented in Chapter 5.

CHAPTER FOUR

THE ENHANCED FUZZY COMMITMENT SCHEME

4.1 Introduction

The previous chapter presented the research methodology that was used for this study. This chapter focuses on the discussion of the proposed approach in relation to the dependency on biometric features in helper data, which is discussed in Section 4.2. The proposed approach consists of three modules: (1) enrollment (2) concealment, and (3) verification. Each module is discussed separately in subsection 4.2.1 – 4.2.4. Then, in Section 4.3, the algorithm is presented along with the prototype to implement the proposed approach. Finally, this chapter ends with a summary highlighting the essence of this chapter in Section 4.3.

4.2 The Proposed Approach

In order to discuss the proposed approach, the problems of this research are revisited. There are two problems with traditional FCS biometric template protection. The first problem concerns the required entropy, while the second problem concerns the characteristic of the bit string (i.e., the feature) to be distinguishable enough among users. As for the first concern, the *HelperData* (*HD)* should not reveal any information about the Key (*K)* or the feature enrolment template *f(en)* to ensure the protection of the authentic template. In the existing technique, the randomness of *K* is not problematic. Using a crypto-secure random number generator (RNG), bit strings with high entropy can be easily produced. But the main problem is the transformation of the biometric template into a bit string from the feature space. In this

transformation process, the bit string is converted into a binary string, which makes it vulnerable to cross-matching (Gilkalaye et al., 2019).

Secondly, the bit string that is transformed should be different for the users. The current technique uses ECC to create a unique biometric bit string, which cannot recover *K* because of how XOR is implemented in FCS (Belyaev et al., 2018). To address these issues, both *f(en)* and *K* need to be uniformly random, including for *HD*. This can be achieved by strengthening the transformation process of the biometric feature space templates into bit strings.

Therefore, the proposed approach was designed to modify the transformation process by executing ECC with an additional hash function to secure the bit string. This modification was made immediately after the transformation process to solve the problem of entropy and distinguishing characteristic. In this way, it is possible to generate a bit string with high entropy and security that meets the requirements of FCS. Hence, when it comes to biometric template protection, it seems that the optimization of the transformation process in the enrollment module is crucial to achieve acceptable system security.

To better understand the proposed approach, a full illustration is shown in Figure 4.1 with the notations listed in Table 4.1. To maintain consistency, the same notations are used throughout the chapters.

Table 4.1

Notations

Notation	**Description**
$f \in \mathrm{R}$	Real-valued feature vectors
f(en)	Binary-string features vectors for enrollment
$Key \in \{0,1\}^k$	Random number generator to produce Key
Key	Key (used in hash function)
ECC *n,k,K*	Convolutional code
FingerCode	Results of binary-string features vectors are encoded into ECC
h(FingerCode)	Result of the hash function SHA-256 of *FingerCode*
h(Key)	Result of the hash function SHA-256 of *Key*
g(f),	Gabor filter
f	Frequency in Gabor filter technique
Φ	Orientation in Gabor filter technique
$\oplus$	Exclusive OR (XOR)
I	Image
R	Response of Image
r	Mean response of Image R
Nr	Number of pixels

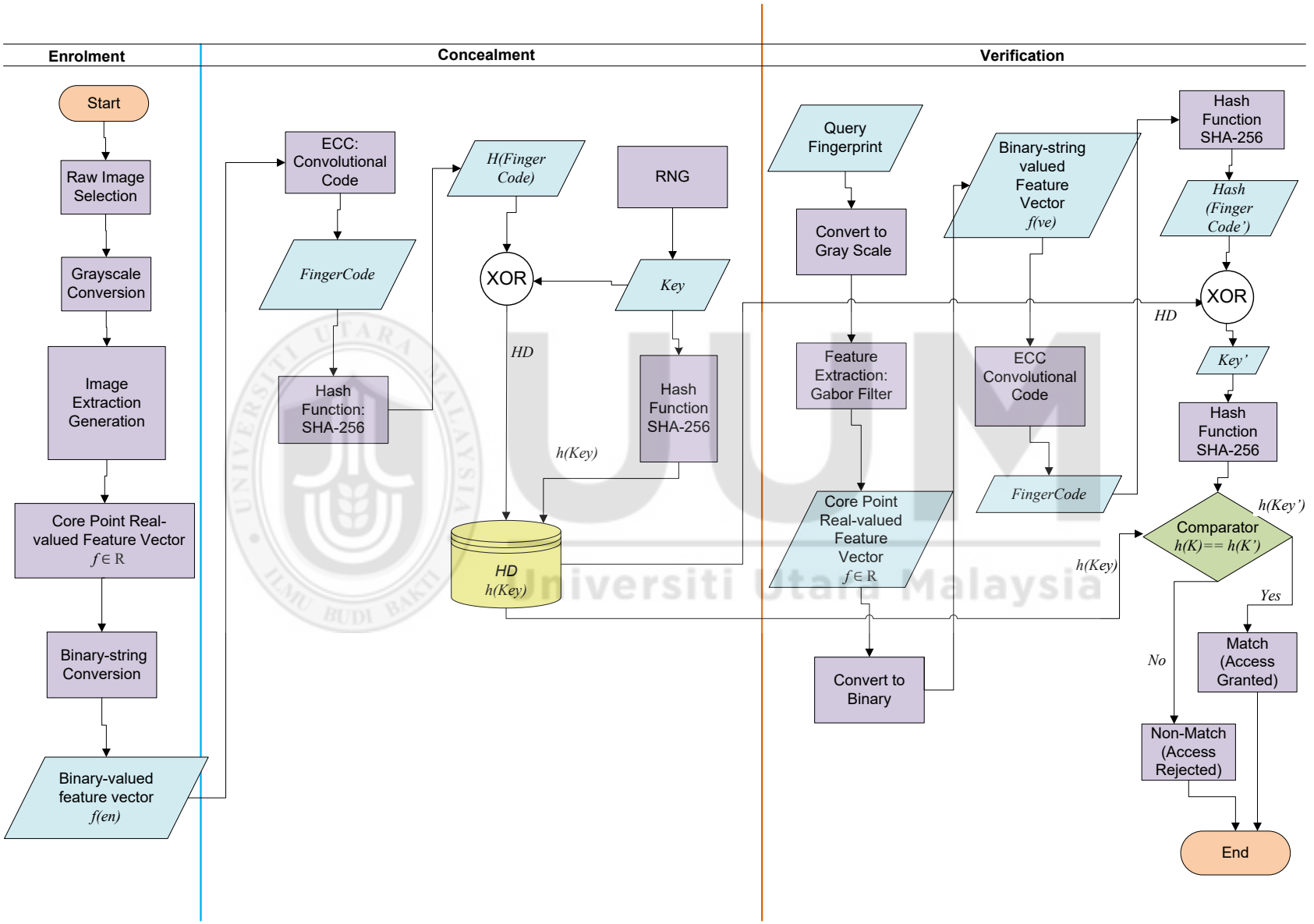


Figure 4.1. Flowchart of the proposed approach

Based on Figure 4.1, the proposed approach consists of three modules: (1) enrollment, (2) concealment, and (3) verification. As mentioned previously a prototype has been developed to consisted of these three modules (refer to Appendix A). The following subsection explains these modules in more detail.

4.2.1 Enrollment Module

The enrollment module consists of four processes: (1) raw image selection, (2) grayscale conversion, (3) image extraction generation, and (4) binary string conversion. Each process has its own detailed steps, which are elaborated as follows;

1. Raw Image Selection

In this process, the user selects a raw input image. In the context of this research, the raw input image is a fingerprint randomly selected from the FVC 2002 DB1 dataset. The size of the raw fingerprint image is 142 kilobytes. A raw fingerprint is selected to compute the feature extraction method. Table 4.1 shows the information about the sample of a raw fingerprint.

Table 4.1

Information about the raw fingerprint sample

Database No.	Filename	Competitions	Image Size	Resolution	Sensor Type
DB1_A	1_1.tif	FVC2002	388 x 374	96 dpi	Optical sensor

The raw fingerprint is then converted to grayscale in order to be processed by the Gabor filter extraction method to produce a core point that is later converted to a binary string.

2. Grayscale Conversion

Once the raw image is selected, it is converted to a grayscale image. The reason for converting the raw image to a grayscale image is that it must be extracted using the Gabor filter extraction technique, which only allows grayscale images.

3. Image Extraction

The steps in this process occur after the raw image has been converted to a grayscale image. The grayscale image is extracted using the Gabor filter extraction technique. The technique involves image enhancement, which is performed to improve the image quality using Fast Fourier Transform (FFT). The image enhancement step is used to minimize noise and increase the separation of ridges and valleys. Although fingerprint image quality cannot be objectively determined, it is basically similar to the sharpness of the ridge structure in the fingerprint image. As a result, it is required to improve the fingerprint image.

To address the low variation typically observed in the background, a simple blockwise variance technique can be employed. This involves applying binary closing (using the MATLAB command "imclose") and subsequent erosion (using the MATLAB command "imerode") to the image. These operations help eliminate gaps in the fingerprint image and minimize undesired effects occurring at the boundary between

the fingerprint and the background. Image segmentation continues until the target condition is satisfied. This is done to prevent the fingerprint from causing undesirable boundary effects with the background. The following requirement must be met: the improved image is separated into non-overlapping pieces of a specific size (usually 32 x 32 or 64 x 64).

A comprehensive filter is used to filter the entire improved picture. Let the current region of interest (previously calculated using some initial parameters) have a maximum value of the filtered image as *Cf_max*. The relative maximum *Cf_rel* is calculated for each non-overlapping block. Finally, a logical matrix F has an element (I, J) that is 1 if (I, J) is a block relative maximum. This value is at least a threshold value; F (I, J) is 0 in all other cases (i.e., when F (I, J) is not a block relative maximum or a block relative maximum below the threshold value). If the logical matrix F has more non-zero elements than a threshold value, the parameters for image segmentation are recalculated and the whole process is done again (Jain et al., 2000).

From the orientation field, a logical matrix is derived, where a pixel (I, J) is 1 if the orientation angle is <= PI/2 (PI 3.1415926535897…). After that, the border of this logical matrix is located in the region of interest of the fingerprint image. Later, the complex filtering result of the enhanced fingerprint image is computed.

The next step in the Gabor filtering technique is to determine the center by calculating the core point. The core point is the convex ridge's point of greatest curvature, which is normally placed in the center of the fingerprint (Azzoubi & Ibrahim, 2015). By

identifying the greatest curvature with comprehensive filtering procedures, the position of a core point may be reliably detected.

Once the core point is determined, it serves as a reference point for the image processing. In particular, it is required for precise sectorization. The picture is then cropped and separated into 80 sections. These sectors are filtered with eight Gabor filters, and the final template is made up of the standard deviations of the gray values included in each sector for each filter (Barrero, 2016). Sectorization is significant because it belongs to the feature extraction component (Patel & Ramalingam, 2019). Finally, the technique undergoes normalization and filtering of the image to produce the final feature vectors, known as binary-valued feature vectors *(f(en))*.

4. Binary string Conversion

The real-valued feature vectors ($f \in R$) generated in the enrolment module are converted into a binary string feature vector, denoted as *f(en)*. The process of converting the binary string is also called binarization and is an important component of the helper data structure.

Most biometric protection systems have been developed to work with binary strings. Thus, the feature vector resulting from feature extraction must be translated into a binary string before further processing can occur. It must also be ensured that the binary feature is provided consistently and independently, so that it is impossible to recover either the biometric template or the random bit string from the different vector without knowing the user's biometric data.

Binarization is used to extract a long, uniformly distributed bit string from biometric templates without significantly affecting the verification performance (Gilkalaye et al., 2019). The binarized features should be provided consistently and independently from a security perspective. The difference vector cannot recover the biometric template or the random bit string without knowing the user's biometric data.

In addition, these properties should be resilient to noise, as the error correction algorithm is limited in its ability to correct errors. Finally, most biometric protection systems were designed to work with binary strings. Thus, the feature vector resulting from feature extraction must be translated into a binary string before further processing can occur.

Figure 4.2 below shows the flowchart of the enrollment module followed by the Algorithm 4.1.

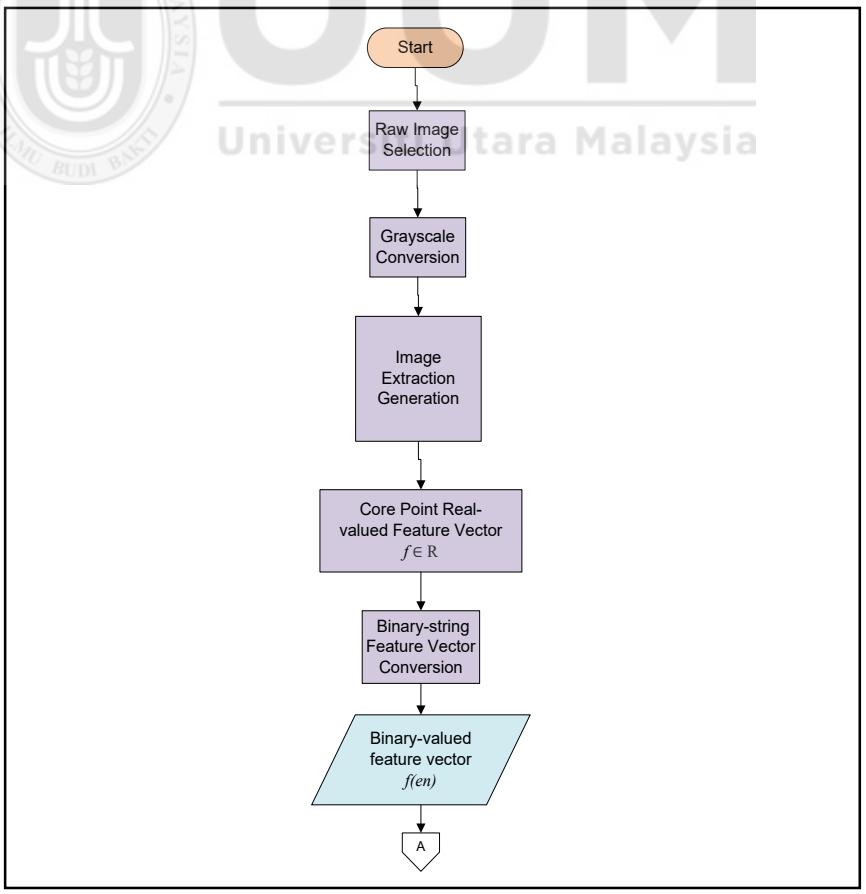


Figure 4.2. Flowchart of the enrollment module

Algorithm 4.1: Enrollment Module

BEGIN

SELECT **Raw fingerprint image**

CONVERT **Grayscale fingerprint image**

GENERATE **"For each frequency *f* in the frequency list *flist*:**

For each orientation Φ in the orientation list *olist*:

Construct: Gabor filter *g(f)*,

Convolve: *g(f)* with original image *I*, get response image *R*

Compute: the mean response in *R*, denote as *r*

Count: # of pixels *Nr* that have a larger value than *r*

Divide: *R* into n *x m* frames

For i = 1 to n:

For j = 1 to m:

Count the # of strong responses $N_{i,j}$ and compute the ratio *r:*

r =Ni,j / Nr;

Append *r* to the feature vector *f(en)*:

OUTPUT ***f(en)=[r1,r2,…r|flist|*|olist|*n*m]".***

END

4.2.2 Concealment Module

After the enrollment module is completed, the concealment module follows. This module consists of four processes: (1) *FingerCode* generation (convolutional code),

(2) hash function generation, (3) random number generation and (4) *HD* generation. Table 4.2 shows the steps of the concealment process.

Table 4.2

Concealment process steps

Process Name	Process Detail
f(en)	A binary string of the feature extraction method with a size of 20-bits.
ECC	The ECC function encodes *f(en)* to produce the *FingerCode* with a size of 40 bits.
Hash	The hash function of SHA-512 encrypts the *FingerCode,* producing *h(FingerCode).* The size is 32 hex codes = 256 bits of binary hash string.
Key	A random number generator produces a *Key*, a binary string of 256 bits.
HelperData HD	*h(FingerCode)* is XORed with *Key,* a binary hash string of 256 bits.
Hash(Key)	The hash function of SHA-256 is used to encrypt the Key that generates *h(Key).* The size is a binary hash string with 256 bits.
Add to Database	*HelperData HD* and *Key* are stored in the database. Therefore, a message box appears when the 'Add to Database' button is clicked, as shown in Figure 4.16.

The concealment module is described in detail below.

1. *FingerCode* Generation

The binary string feature vector *f(en)* is then encoded with the ECC function, which in the context of this research is the convolutional code, to generate the *FingerCode.* The message in convolutional codes is made of output bits that are generated by applying Boolean functions to the data stream as it slides. The input data bits are not divided into blocks but are input as data streams that are convolved into output bits according to the logic function of the encoder (Den, 2020).

The information is passed sequentially through a linear finite-state shift register to generate a convolutional code. The shift register consists of a specified number of bit stages and Boolean function generators.

A convolutional code can be represented as *(n,k, K),* where the components are (Den, 2020):

1. *"k* is the number of bits shifted into the encoder at one time. In general, $k = 1$.
2. *n* is the number of encoder output bits corresponding to *k* information bits.
3. The code rate, $R_c = k/n$.
4. The encoder memory, a shift register of size *k*, is the constraint length.
5. *n* is a function of the present input bits and the contents of *K*.
6. The state of the encoder is given by the value of (*K - 1)* bits".

3. Hash Function Generation

The resulting *FingerCode* from the ECC is then hashed using cryptographic hash functions SHA-256, producing *h(FingerCode)*. The size of the hash SHA-256 is 32 bytes, which corresponds to 32 hex codes. The hash functions convert input data of arbitrary size (e.g., *FingerCode* in this context) into a fixed-size result (e.g., 256

bits) called a hash value (or hash code, message digest, or hash). Our proposed approach introduces a hash function to strengthen the binary features such that it becomes impossible to recover either the biometric template or the random bit string without knowing the user's biometric data.

Besides, the main advantage of the hash function is speed**.** The access time to an element is constant time, so lookup can be performed very quickly. Hash tables are especially useful when the maximum number of entries can be determined ahead of time. In addition, data can also be scrambled in such a way that it is impossible to be unscrambled. Therefore, SHA-256 is one of the most secure hash functions (Maetouq et al., 2018), which justifies its inclusion in this proposed approach. Moreover, a Merkle-Damgård structure from a one-way compression function that uses the Davies-Meyer structure from a specialized block cipher is used to build SHA-256 (Maetouq et al., 2018).

4. Random Number Generation

Once the *FingerCode* has been hashed, the following step is to create a random number using a random number generator (RNG). A random number generator (RNG) is a mathematical device that is designed either as a hardware device or computationally to produce a random series of numbers that should not have any noticeable pattern in their generation or appearance. This random number becomes the *Key* which is later hashed using SHA-256 to strengthen its integrity before being stored in the database. This means that hash functions serve as a checksum or allow someone to see if data has been tampered with after it has been matched. They also serve as a means of identity verification.

5. *HelperData* Generation

In this step, *HelperData* (*HD)* is computed by the Exclusive-OR operation performed between the *h(FingerCode)* generated in step three and the *Key* generated in step four. The values of *Key* and *h(FingerCode)* are deleted from the records; only the value of *HelperData* is stored in the database.

In our proposed approach, the helper data stored in the database is now in the form *h(FingerCode)* $\oplus$ *Key*, which reduces the dependency on helper data and thus strengthens the protection against privacy leakage.

Figure 4.3 shows the flowchart of the concealment module, followed by the algorithms of the concealment module, as shown in Algorithms 4.2 – 4.4.

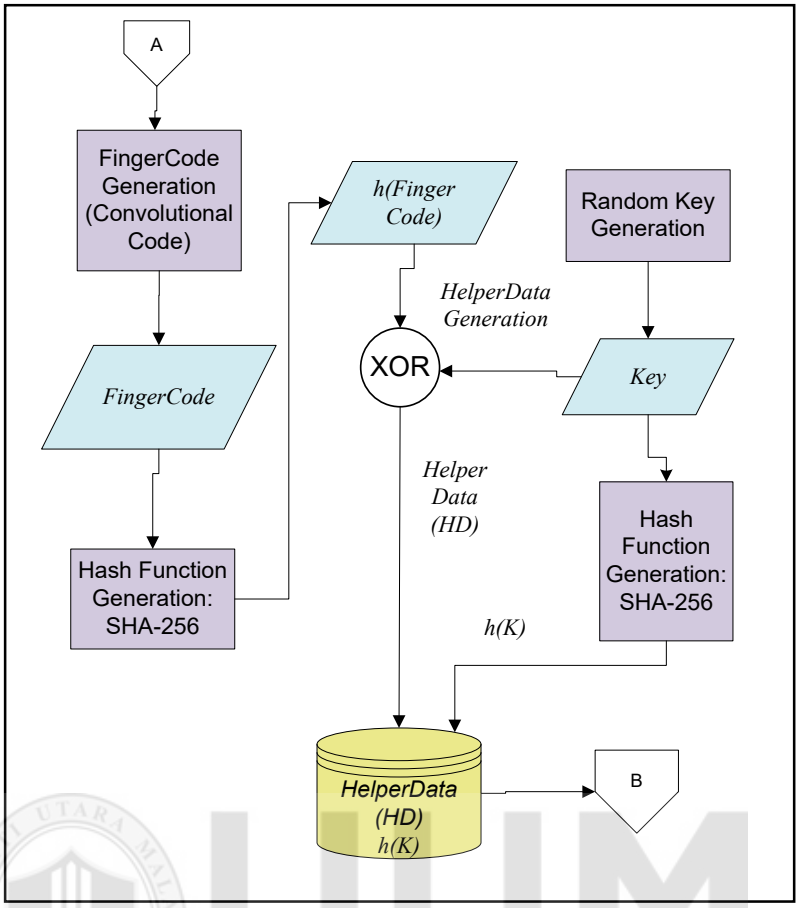


Figure 4.3. Flowchart of the concealment module

Algorithm 4.2: FingerCode Generation

BEGIN

GET The binary string feature vector *f(en)*

ENCODE The binary string feature vector *f(en)* into ECC ()

RETURN Result for encoded ECC *FingerCode*

END

Algorithm 4.4: Hash Function Generation *h(FingerCode)*

BEGIN

GET Encoded ECC, *FingerCode*

CALL Hash function *SHA-256* to hash the *FingerCode*

RETURN Result of hash *h*(*FingerCode)*

DISPLAY *h*(*FingerCode)* in 32-byte hex codes and 256-bits binary string

END

Algorithm 4.5: ***HelperData*** **Generation**

BEGIN

GET *h(FingerCode).*

GENERATE Random-Number-Generator RNG

RETURN Result of hash fingercode *h*(*FingerCode).*

DISPLAY *h*(*FingerCode)* in 32-byte hex codes and 256-bits binary strings.

GET Hash fingercode *h(FingerCode)* in 256-bits binary string.

GENERATE Key *K* from RandomNumberGenerator *RNG* ($K \in \{0,1\}^K$) 256-bits binary string.

RETURN Key *K* 256-bits.

DISPLAY Key *K* in 256-bits binary string.

CALCULATE Key XOR with *h(FingerCode)* to produce *HelperData*, $HD=K \oplus h(FingerCode)$.

GENERATE Hash function *SHA-256* to hash the Key *Key.*

RETURN Result of hash *h*(*Key).*

DISPLAY *h*(*Key)* in 256-bits binary string.

INSERT Store *HD* and *Key* into Database.

END

4.2.3 Verification Module

The verification module seeks to estimate the similarity between two given fingerprint images, *f(en)* and *f(ve).* Fingerprint verification is based on finding the Euclidean distance between the related fingerprint feature templates. The core point established the translation invariance in the fingerprint feature template. Then, the features in the fingerprint feature template itself were cyclically rotated to achieve approximate rotational invariance. Verification was done between the feature vectors of the input fingerprint *(f(en))* and the query template fingerprint *(f(ve))* based on Euclidean distance, where the system calculates the match score for the features as a function of the closest distance. The match score is then converted to a binary string for matching. Table 4.3 shows the steps of the verification process.

Table 4.3

Verification process steps

Process Name	Process Detail
f(ve)	Binary string of the query feature extraction method in a size of 20 bits.
ECC	ECC function to encode *f(ve)* to produce the *FingerCode'* with a size of 40 bits.
Hash	Hash function of SHA-256 to encrypt the *FingerCode',* producing *h(FingerCode').* The size is 32 hex codes = 32-byte binary hash string.
HelperData HD	*HelperData HD* is retrieved from the database.
Key'	*HelperData HD is* XORed with *h(FingerCode')*, a 256-bit binary hash string, to produce the *Key'*.
Hash(Key')	Hash function of SHA-256 to encrypt the *Key'*, which generates *h(Key').* The size is a binary hash string of 256 bits.
Comparator	*h(Key)* from the database is compared with *h(Key').* If *h(Key)== h(Key'),* a message box is displayed indicating that the match was successful, or a warning box indicating a rejection, as shown in Figures 4.18 and 4.19.

Figure 4.4 shows the flowchart of the verification module. The algorithm for the verification module is the same as Algorithm 4.1.

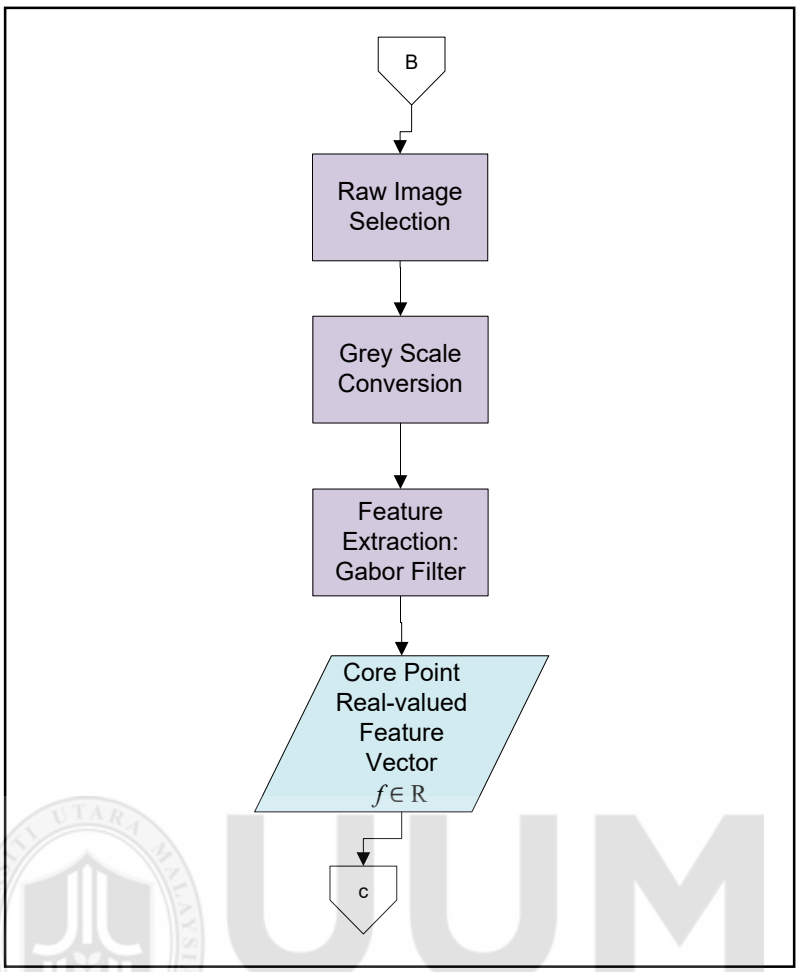


Figure 4.4. Flowchart of the verification module

Furthermore, the match score generated in the verification module is processed by the ECC to produce *FingerCode'* and the same hash function is implemented to produce *h(FingerCode')*. The *HelperData (HD)* is then retrieved from the database, where it is computed by XORing with *h(FingerCode')* to retrieve the *Key'*. For matching, the hash of *h(Key')* is calculated and compared with the stored *h(Key)* to determine whether the matching is successful Figure 4.5 shows the continuation of the flowchart of the verification module followed by the algorithm shown in Algorithms 4.6, 4.7, 4.8, and 4.9.

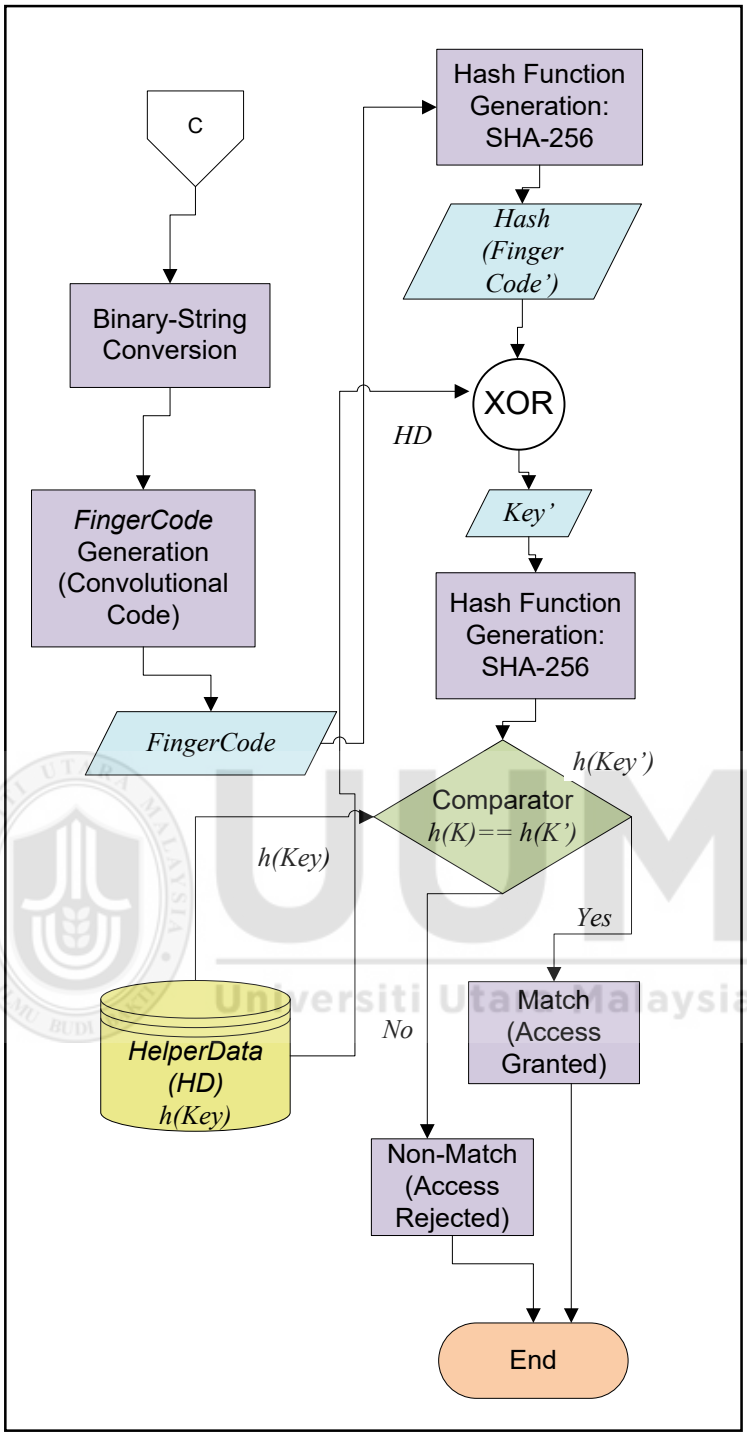


Figure 4.5. Flowchart of the verification module (continued)

Algorithm 4.6: Binary-string Conversion

BEGIN

GET Gabor filter core point

CONVERT Core point into binary string

RETURN Binary string feature vector *(f(ve))*

END

Algorithm 4.7*: FingerCode* Generation (Convolutional Code)

BEGIN

GET the binary-string feature vector *f(ve)*

ENCODE the binary-string feature vector *f(ve)* into ECC ()

RETURN result for encoded ECC *FingerCode'*

END

Algorithms 4.8: Hash Function Generation *h(FingerCode')*

BEGIN

GET encoded ECC, *FingerCode'*

CALL hash function *SHA-256* to hash the *FingerCode'*

RETURN result of hash *h*(*FingerCode')*

DISPLAY *h*(*FingerCode')* in 32-byte hex codes and 256-bits binary string

END

Algorithm 4.9: Comparator

BEGIN

GET ***HelperData HD*** **from database, 256-bits binary string** ***h*(*FingerCode'*)**

CALCULATE ***HelperData HD*** **XOR with** ***h(FingerCode')*** **to retrieve Key',** ***Key'=HD⊕h(FingerCode')***

RETURN **result of XOR,** ***Key'***

GENERATE **hash function** ***SHA-256*** **to correct** ***Key', h(Key')***

GET ***h(Key)*** **from the database**

COMPARE **if** ***h(Key') == h(key)***

RETURN **Matching Successful**

ELSE **Matching Not Successful**

DISPLAY **Message Box**

END

4.3 Summary

This chapter presented the proposed approach, which can be generally divided into three modules. The first module is enrollment, followed by the concealment and verification modules. The proposed approach aims to modify the existing FCS by implementing convolutional code as ECC function and hash function (SHA-256) in order to solve the problem of dependency on helper data as well as privacy leakage. The modification was made in the concealment module, specifically in Processes 2 and 3, as explained above. As a result of the modification, the helper data no longer

depends on the binary-valued feature vectors, which addresses the problem of dependency on helper data. Moreover, the hash function helps to strengthen the binarization process, which reduces the privacy leakage issue. The proposed approach makes it more difficult for the intruder to extract relevant biometric information about the user. This is because no information about the feature template or the Key has been stored. The proposed approach also shows uniqueness in that the hash (*FingerCode*) bit strings are different from one user to another, which should be approximately independent. This is to prevent an adversary from easily creating collisions, since he could predict the response of *h(Key)* to recover the Key. Thus, it can be assumed that the proposed approach uses two layers of security before converting to *HelperData*. The completion of this chapter shows that the Research Objective 2 has been achieved, and the next chapter focuses on the evaluation of the proposed approach.

CHAPTER FIVE

SECURITY, PRIVACY AND PERFORMANCE EVALUATION OF THE PROPOSED APPROACH

5.1 Introduction

In the previous chapter, the problem of privacy leakage was discussed, and the concern for the transformed bit string of the biometric should be such that the privacy leakage problem can be solved. After all, current biometric environments must manage a growing number of users, huge amounts of data, and an increasing variety of end-user devices. Inadequate protection can therefore lead to a massive leakage of sensitive data, and the loss of trust can significantly damage an organization's reputation.

This chapter discusses the evaluation studies that were conducted to determine the security, irreversibility, and privacy leakage of the proposed approach. The evaluation was conducted in three parameters; the first parameter analyzes the security perspective in Section 5.2, followed by a second parameter (irreversibility) focusing on the privacy perspective in Section 5.3, and the third parameter is the performance of the proposed approach in Section 5.4. Details of each evaluation, such as the procedures, apparatus, and measurements, are discussed in the next respective subsections. This is followed by the comparative analysis in Section 5.6, and the chapter ends with Section 5.7, which summarizes the essence of the chapter.

5.2 Security Evaluation

This section presents the security evaluation of the proposed approach. In the context of FCS, security is defined as the difficulty of retrieving the secret key from the helper data. In order to evaluate the security of the proposed approach, several sets of

experiments were conducted. The dataset used for the experiments was FVC2002 DB1 (FVC2002, 2002). This dataset consists of 100 users, each with eight samples available (100 users x 8 samples = 800 samples). The images of the dataset have a size of 388 x 374 pixels and a resolution of 96 dots per inch (96 dpi).

Analysis and Result of the Security Evaluation

The proposed approach stores only the helper data $HD = \{ HD_1, HD_2, HD_3 \ldots, HD_N\}$ and the hash of the key $h(Key) = \{h(Key_1)\ h(Key_2),\ h(Key_3),\ \ldots..\ h(Key_N)\}$ where N is the number of sectors created on the fingerprint image, in the database and the rest of the information is eliminated. This section contains the security analysis of the stored information, i.e., (1) the hashed key *h(Key)* and (2) the helper data in the proposed approach.

h(Key): The proposed approach uses the SHA-256 hash function to secure the *h(Key)* stored in the database. The adversary learns about the hash function used in the proposed approach.

The hash functions are one-way functions that are computationally complex to invert, so *h(Key)* is secure in the database. However, given a *q*-bit hash function requiring $2^{q/2}$ operations, the birthday attack is the most efficient way to discover a collision. (Menezes et al., 1997). The security parameter for the proposed approach, that uses SHA-256, is $q = 256$, which corresponds to the necessary minimum of 2^{128} operations to locate a pair of colliding codewords. The adversary needs $2^{128 \times 80}$ colliding pairs of codewords for the proposed approach since it creates 80 sectors in the fingerprint image.

Helper data: The helper data $HD = \{HD_1, HD_2, HD_3 \ldots, HD_N\}$ stored in the database are 256 bits each. This helper data is a combination of *h(fingercode)* and *Key*, and the helper data alone is not sufficient to provide information about *h(fingercode)* and *Key* of the user. For successful matching of the user, the query binary-valued feature vector $f_{ve} = \{ f_{ve1}, f_{ve2}, f_{ve3}, \ldots f_{veN}\}$ should be sufficiently close to enrollment binary-valued feature vector $f_{en} = \{\{ f_{en1}, f_{en2}, f_{en3}, \ldots f_{enN} \}$, where both f_{en} and f_{ve} 0 for $(1 \leq i \leq N)$ are 256 bits each. The proposed scheme builds 80 sectors on a fingerprint template, so the adversary needs $2^{256 \times 80}$ tries to brute force *h(fingercode)* from the helper data. Hence, it is difficult for an adversary to learn *h(fingercode)* from the helper data stored in the database.

The proposed approach ensures that the stored data does not, by itself, expose any major information about the biometric templates, protecting the privacy of the rightful owner. Additionally, the security analysis examines how challenging a brute force attack would be against the data kept in the database.

5.3 Privacy Evaluation

This section is about the privacy evaluation of the proposed approach. In the context of this research, privacy refers to the ability to protect personal information and can be evaluated using an adversarial attack, namely a cross-matching attack.

As the method for a cross-matching attack, consider two distinct biometric-based applications that both employ the same template protection scheme. Each application has its own database, which stores the user's templates. Moreover, an adversary gains access to both databases. The protected template is stored along with the helper data and the hash key. The adversary attempts to determine who is enrolled in both

applications. To determine this, he uses a cross-matching classifier to compare two templates from each database. The classifier provides a cross-matching distance score, which is then used to determine if the templates are genuine (from the same individual) or not (imposter). The apparatus for evaluating privacy is similar to that for evaluating security (see Section 5.2.2).

Analysis and Result of the Privacy Evaluation

The cross-matching attack seeks to identify the user from whom two sets of helper data are derived. The correlation attack attempts to estimate the biometric traits or secret key utilized in the protection procedure by connecting various helper data from other sources (the same biometric trait is employed by both systems.). In the proposed approach, the cross-matching CR can be assessed by the probability P that the distance DH between different helper data HDS1 and HDS2 is lower than a threshold t:

$$CR(t) = P(D_H(HD^{S1}, \mathrm{HD}^{S2}) < t) \quad (5.1)$$

Assume the adversary is aware of both the HD^{S1} and HD^{S2} of the two systems S_1 and S_2 helper data. The attacker can calculate the separation between the user's two biometric traits in both systems by using a correlation attack. Thus, the adversary can recover the original biometric features f_{en} by decoding the *h(fingercode).* If the adversary knows the secret key Key^{S1} and *h(fingercode)'* such as *h(fingercode)'* = $XOR(h(fingercode)^{S1}, h(fingercode)^{S2})$, the secret key Key^{S1} can be recovered by the adversary using Key^{S1} and $h(fingercode)^{S2}$ (i.e., $h(fingercode)^{S2} = XOR(h(fingercode)^{S1}, h(fingercode)')$, where $h(fingercode)^{S1}$= encode (Key^{S1}) and $h(fingercode)^{S2}$ = encode (Key^{S2})). Consequently, the original biometric features of the

second system can be restored by the adversary (i.e., $f_{en}^{S2} = XOR\ HD^{S2}$, $h(fingercode)^{S2}$)).

The adversary's objective is to retrieve the second system's S_2 biometric characteristics or secret key if *h(fingercode)'* is unknown. The adversary can then determine the correlation between the two helper datasets and calculate the separation between the two biometric features, f_{en}^{S1} and f_{en}^{S2}, where f_{en}^{S2} can be estimated by locating the nearest $h(fingercode)_n$ (i.e., $D_H(f_{en}^{S1}, XOR(HD^{S2}, h(fingercode)_n))$ is minimal).

The attacker can only approximate the distance between the original biometric characteristics of both systems without knowing their actual values if Key^{S1} and Key^{S2} are unknown and cannot be determined. In addition, if the opponent does not have access to the second secret key, he cannot proceed with the cross-matching without prior knowledge of the key and biometric features. Thus, the proposed approach has the aspect of unlinkability and prevents cross-matching.

5.4 Performance Evaluation

In this section, the performance evaluation of the proposed approach is presented. In evaluating the correctness of a biometric system, e.g., to measure its biometric performance, genuine and impostor attempts were calculated with the system and all similarity scores were recorded. By calculating a variable score of thresholds to the similarity scores, sets of FRR and FAR can then be calculated. During the comparison, the more accurate performance is the one that would have a lower FRR at the same level of FAR. The best biometric performance is at the top of the plot (Precise Biometrics AB, 2014).

The FAR and FRR are used to evaluate the performance of the proposed approach. The FRR can be defined as the efforts of impostor that falsely indicated to match a template of another object represented by this percentage. FAR can be defined as the percentage of efforts of genuine users that are falsely classified as not matching.

Therefore, the Receiver Operating Characteristic (ROC) curve, which shows the FRR against the FAR at different thresholds for the match score, is used to evaluate the performance analysis of the proposed approach. The performance is also evaluated using the EER, which is the error rate where the FAR and the FRR are equal. The EER indicates the minimum verification error, and the threshold value is selected according to the minimum error.

Depending on the selection of the threshold score, the system may incorrectly accept any number of imposter patterns. The FAR is the threshold value that depends on the fraction of mistakenly accepted patterns divided by the total number of imposter patterns. On the other hand, if the threshold for classification scores is set too high, a portion of users will be falsely rejected, which is called the FRR.

The performance evaluation calculates the TPR of the genuine fingerprint, the TNR of the imposter fingerprint, and the accuracy at a threshold of 40%. The proposed approach's accuracy is defined as the proportion of real findings that are either true positive or true negative in comparison to the total outcomes.

In addition, the FAR and FRR values are obtained from the TPR and TNR data. The genuine (matching) distribution is defined as the match scores obtained between pairs of samples from the same person. In comparison, the imposter (non-matching)

distribution is defined as match scores generated between pairs of samples from different persons.

Assume that the proposed approach's match attempts have a ‘score’ within the closed intervals [0, 1], where 0 denotes a complete match and 1 denotes no matches at all. All users, genuine (positive) and imposters (negative), are matched when the threshold is set to 0. On the contrary, there is a strong likelihood of no match if the threshold is set to 1. Therefore, the threshold is kept somewhere between 0 and 1 in real-time systems (Maltoni et al., 2009). Consequently, this threshold setting may occasionally prevent the authentication of the legitimate users, which is referred to as FRR, but also the imposters can be authenticated, which is indicated by FAR.

Analysis and Result of the Performance Evaluation

The similarity score of the genuine templates is calculated and shown in Table 5.1. The data shown in Table 5.1 is analyzed to calculate the number of true positive and false negative matches.

Table 5.1

The match score sample for the calculation of the true positive rate of the proposed approach

Finger print	Match Score with 1_1.tif	Finger print	Match Score with 2_1.tif	Finger print	Match Score with 3_1.tif	Finger print	Match Score with 4_1.tif
1_1.tif	100	2_1.tif	100	3_1.tif	100	4_1.tif	100
1_2.tif	60.09	2_2.tif	48.49	3_2.tif	48.25	4_2.tif	68.24
1_3.tif	52.00	2_3.tif	51.15	3_3.tif	65.25	4_3.tif	65.75
1_4.tif	43.87	2_4.tif	**27.58**	3_4.tif	75.58	4_4.tif	78.23
1_5.tif	**35.57**	2_5.tif	63.25	3_5.tif	49.24	4_5.tif	75.25
1_6.tif	69.22	2_6.tif	58.56	3_6.tif	85.47	4_6.tif	68.56
1_7.tif	47.15	2_7.tif	**38.74**	3_7.tif	78.56	4_7.tif	55.12
1_8.tif	49.04	2_8.tif	57.25	3_8.tif	61.47	4_8.tif	78.25

Finger print	Match Score with 5_1.tif	Finger print	Match Score with 6_1.tif	Finger print	Match Score with 7_1.tif	Finger print	Match Score with 8_1.tif
5_1.tif	100	6_1.tif	100	7_1.tif	100	8_1.tif	100
5_2.tif	78.25	6_2.tif	52.12	7_2.tif	56.23	8_2.tif	45.23
5_3.tif	65.13	6_3.tif	45.23	7_3.tif	75.23	8_3.tif	70.23
5_4.tif	90.23	6_4.tif	74.23	7_4.tif	**24.02**	8_4.tif	68.56
5_5.tif	85.36	6_5.tif	83.20	7_5.tif	40.18	8_5.tif	50.12
5_6.tif	42.12	6_6.tif	63.42	7_6.tif	50.14	8_6.tif	**37.25**
5_7.tif	51.02	6_7.tif	45.20	7_7.tif	56.32	8_7.tif	48.41
5_8.tif	57.23	6_8.tif	42.05	7_8.tif	48.23	8_8.tif	50.23

From Table 5.1, the total number of TPs is 59 and the total number of FNs is 5 (bold numbers), which is below the threshold score (40%).

The similarity score of the impostor for the proposed approach is calculated and presented in Table 5.2. Finally, the data presented in Table 5.2 are analyzed to calculate the number of TNs and FPs of the proposed approach.

Table 5.2

The match score for the calculation of the true negative rate of the proposed approach

Finger print	**Match Score with 2_1.tif**	**Finger print**	**Match Score with 3_1.tif**	**Finger print**	**Match Score with 4_1.tif**	**Finger print**	**Match Score with 5_1.tif**
1_1.tif	25.14	2_1.tif	31.24	3_1.tif	18.44	4_1.tif	16.32
1_2.tif	13.25	2_2.tif	25.63	3_2.tif	37.56	4_2.tif	35.32
1_3.tif	36.15	2_3.tif	19.45	3_3.tif	25.14	4_3.tif	17.56
1_4.tif	26.21	2_4.tif	18.55	3_4.tif	17.58	4_4.tif	27.53
1_5.tif	27.23	2_5.tif	25.13	3_5.tif	19.36	4_5.tif	38.25
1_6.tif	39.45	2_6.tif	35.66	3_6.tif	26.85	4_6.tif	32.55
1_7.tif	38.24	2_7.tif	25.36	3_7.tif	29.35	4_7.tif	34.57
1_8.tif	36.14	2_8.tif	36.47	3_8.tif	30.54	4_8.tif	35.15
Finger print	**Match Score with 5_1**	**Finger print**	**Match Score with 6_1**	**Finger print**	**Match Score with 7_1**	**Finger print**	**Match Score with 8_1**
5_1.tif	12.45	6_1.tif	35.47	7_1.tif	28.64	8_1.tif	19.20
5_2.tif	19.23	6_2.tif	31.24	7_2.tif	39.14	8_2.tif	26.30
5_3.tif	25.14	6_3.tif	36.12	7_3.tif	26.44	8_3.tif	30.25
5_4.tif	**58.47**	6_4.tif	27.14	7_4.tif	14.75	8_4.tif	36.41
5_5.tif	38.23	6_5.tif	15.24	7_5.tif	27.55	8_5.tif	34.02

5_6.tif	28.25	6_6.tif	20.14	7_6.tif	23.41	8_6.tif	25.07
5_7.tif	25.45	6_7.tif	24.27	7_7.tif	32.54	8_7.tif	25.33
5_8.tif	27.56	6_8.tif	23.25	7_8.tif	36.96	8_8.tif	36.04

From Table 5.2, the total number of TNs is 63 and the total number of FPs is 1 (bold number), which is above the threshold score (40%).

The calculation of FRR, FAR, TPR and accuracy is shown in Table 5.3.

Table 5.3

Calculation of FRR, FAR, TPR and accuracy

Parameter	**Calculation**
FRR	FRR= FNR = FN / TP + FN = 5 / 59 + 5 = 0.07813
FAR	FAR = FPR = FP / TN + FP = 1 / 63 + 1 = 0.0156
TPR	TPR = TP / TN + FP = 1 – FRR = 59 / 63 + 1 = 0.9219
Accuracy	Accuracy = TP+TN / TP+TN+FP+FN = 59 + 63 / 59 + 63 + 1 + 5 = 0.9531 = 50 + 64 / 128 The accuracy of matching in the proposed approach is 95.31%.

The data shown in Table 5.1 and Table 5.2 are analyzed to evaluate the FNR, TPR, and FPR of the proposed approach at different threshold values, as shown in Table 5.4.

Table 5.4

FAR, FRR and accuracy values with different thresholds

Threshold %	FRR	TPR	FAR	Accuracy %
0	0.0000	1.0000	1.0000	50.00
5	0.0000	1.0000	1.0000	50.00
10	0.0000	1.0000	1.0000	50.00
15	0.0000	1.0000	0.9765	51.56
20	0.0000	1.0000	0.8312	58.59
25	0.0000	1.0000	0.6411	67.97
30	0.0000	1.0000	0.5378	73.44
35	0.0000	1.0000	0.3210	84.38
40	**0.0781**	**0.9219**	**0.0156**	**95.31**
45	0.2124	0.7875	0.0000	89.06
50	0.3325	0.6675	0.0000	83.38
55	0.4524	0.5476	0.0000	77.38
60	0.5620	0.4380	0.0000	71.90
65	0.6689	0.3311	0.0000	66.56
70	0.7602	0.2398	0.0000	62.00
75	0.7721	0.2279	0.0000	61.40
80	0.8214	0.1786	0.0000	58.93
85	0.8496	0.1504	0.0000	57.52
90	0.8841	0.1159	0.0000	55.80
95	0.8936	0.1064	0.0000	55.32
100	1.0000	0.0000	0.0000	50.00

Based on Table 5.5, there are two common ways to graphically display the performance evaluation results, namely

1. The ROC curve (Precise Biometrics AB, 2014). The ROC (Receiver Operating Characteristic) curve depicts the graphical representation of the correlation between the True Positive Rate (TPR) and the False Acceptance Rate (FAR). This curve illustrates the TPR (Y-axis) against the FAR (X-axis) for various threshold values. In Figure 5.1, the ideal performance curve in the ROC curve is the one positioned closer to the top.

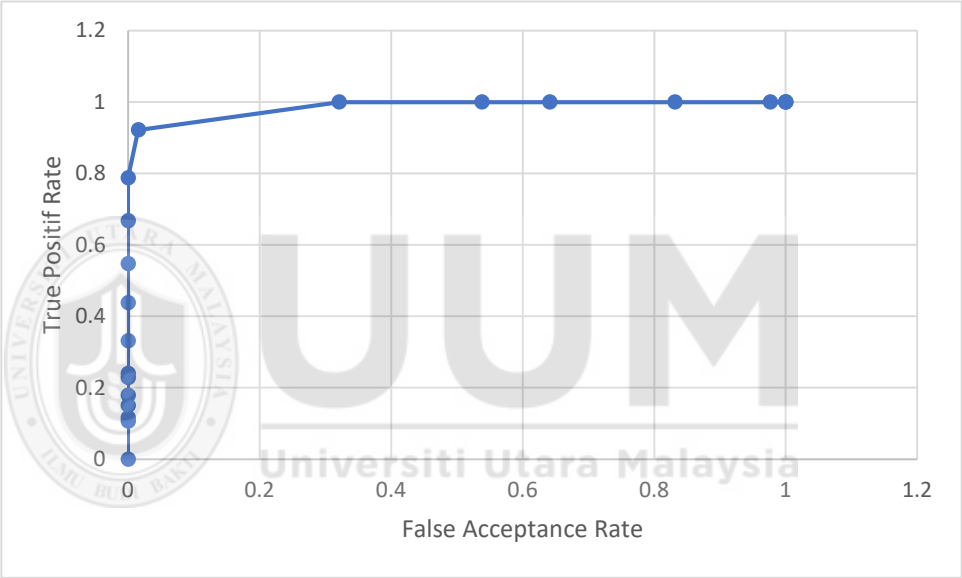


Figure 5.1. ROC curves of the proposed approach

2. The detection error tradeoff (DET) graph (Precise Biometrics AB, 2014). DET is the graphical representation of the error rates for a set of thresholds, i.e., FRR (on the Y-axis) vs. FAR (on the X-axis) for a set of threshold values. The best

performance curve in the DET graph is close to the bottom (X-axis), as shown in Figure 5.2.

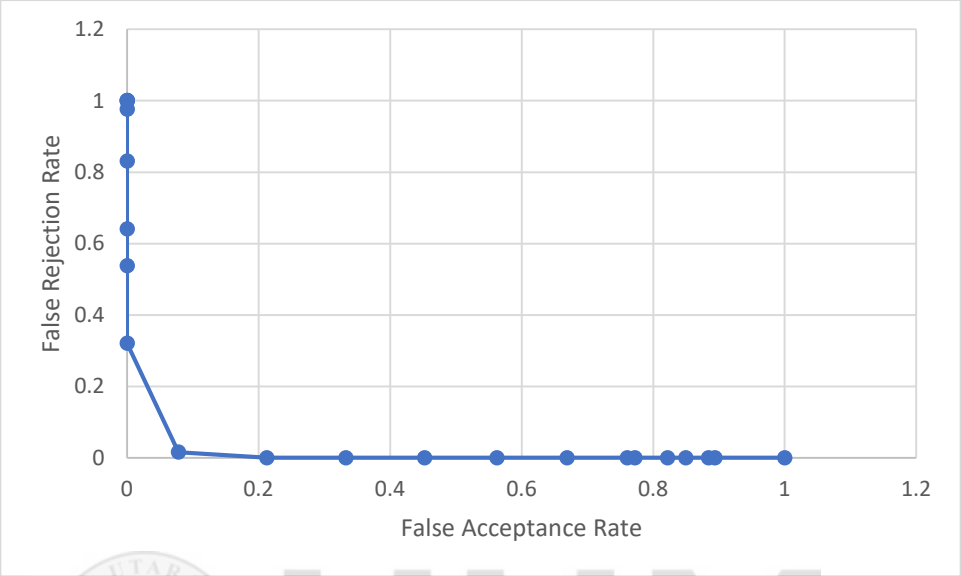


Figure 5.2. DET curves of the proposed approach

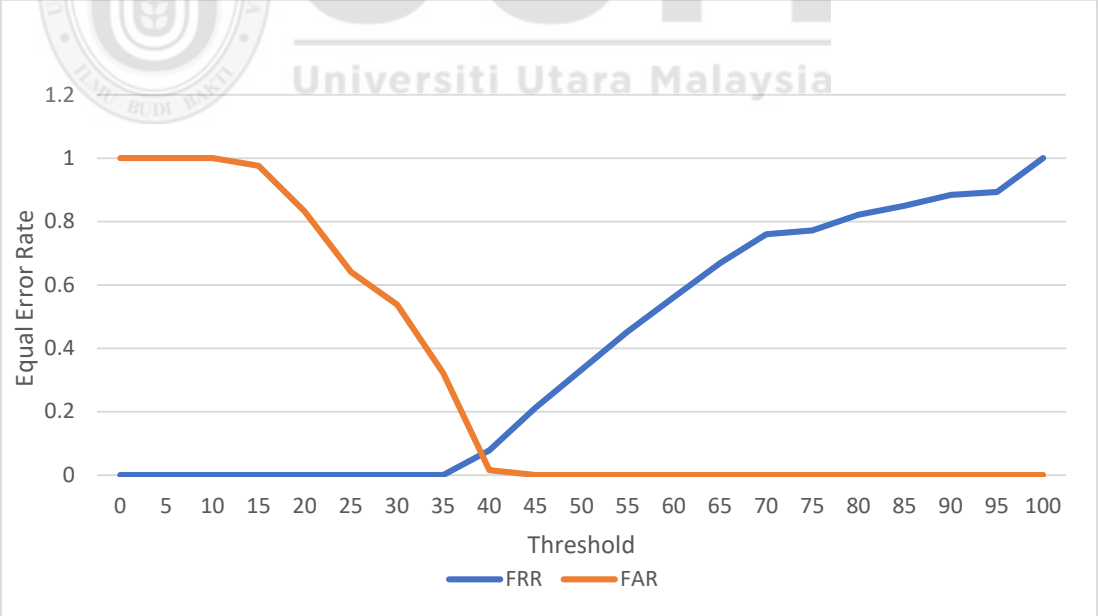


Figure 5.3. The EER of FAR and FRR for the proposed approach

The ROC curve between FAR and FRR with different thresholds is shown in Figure 5.3. The EER is the point at which FRR and FAR are equal. The ROC curve, for example, indicates that the EER point is equivalent to 1.54% at a threshold value of 0.4. Hence, the proposed approach is able to achieve a high accuracy rate of 95.31% and EER = 1.54%, indicating high performance in matching.

5.5 Comparative Analysis

In order to strengthen the evaluation procedures, the performance of the proposed approach was compared with the most similar previous work by Sandhya et al. (2020). They described a technique for choosing neighbors of a minutia point variable to create a bio-cryptosystem for fingerprints. They build a Delaunay triangulation for the minutia points in the fingerprint image so as to choose a variable number of neighbors for each one. Additionally, they obtain a bit string from their Delaunay triangulation neighbors. The bit string is encrypted using convolutional coding and FCS to safeguard it. The experimental tests are performed using the FVC 2002 databases. Table 5.2 shows the comparison between the proposed approach and the comparison study.

Table 5.5

Comparison between the proposed approach and the comparison study

	Proposed approach	Sandhya et al. (2020)
Enrollment	Obtain the core point from Gabor filter	Construct Delaunay triangulation from minutia points
	Convert to binary string	Convert to binary string
Binding	ECC - Convolutional Code (Using biometric feature core point) Cryptographic Hash Function SHA-256	ECC - Convolutional Code (Using random key) Cryptographic Hash Function SHA-1
Verification	Generating Match Score	Generating Match Score
Matching	Hash value comparator	Hash value comparator
Dataset	FVC 2002 DB1	FVC 2002 DB1, DB2, DB3

In the literature, Sandhya et al., (2020) presented their work using the ROC curve, which shows the accuracy performance of their technique. In this graph, the genuine acceptance rate (GAR) is displayed against the FAR at different matching threshold settings. GAR is the percentage ratio of successfully authorized genuine users to the total number of trials with legitimate users. It is also known as the FRR inverse (1-FRR), TPR, or true match rate. Better performance is associated with a greater area under the curve (closer to the curve towards the top of the graph).

Figure 5.4 illustrates the comparative analysis of the performance accuracy of the proposed approach against Sandhya et al. (2020).

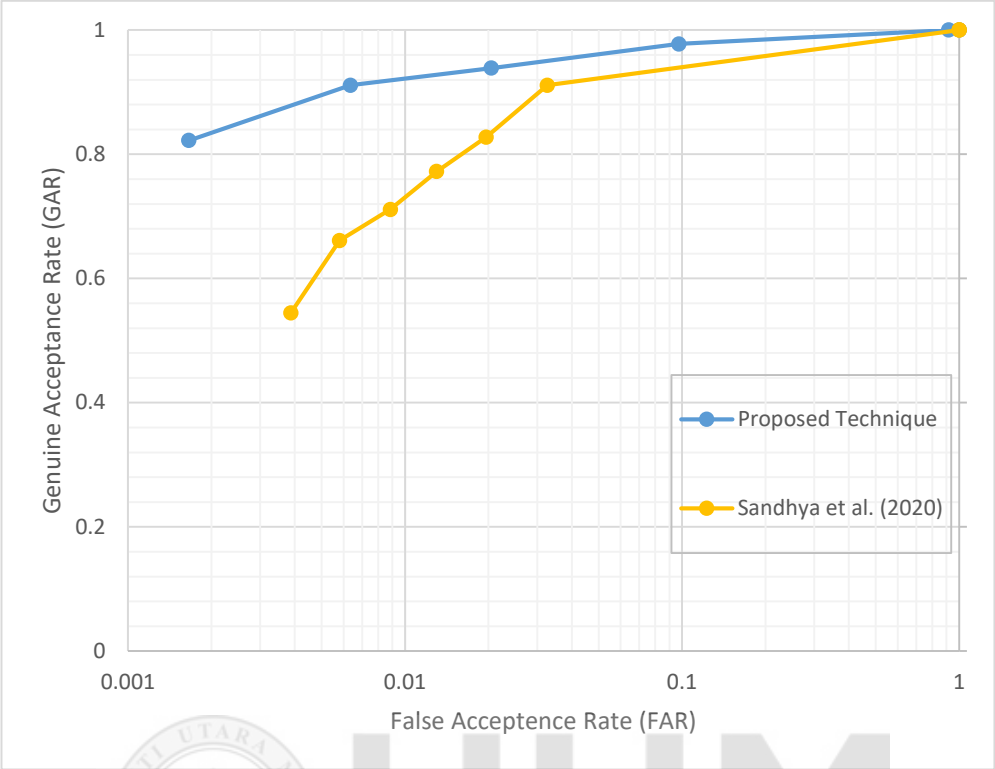


Figure 5.4. ROC curves are used to compare the performance of Sandhya et al. (2020) and the proposed approach from FVC 2002 DB1.

As observed in Figure 5.4, the accuracy of the proposed approach was slightly (1.42%) better than that of Sandhya et al. (2020). These differences seem to be justifiable considering that they emphasized security and privacy aspects as much as our proposed approach, which explains the trade-off effect between security, privacy, and performance (Mucchi et al., 2019; SQL Management Suite, 2021). It also shows that the proposed approach has better performance than Sandhya et al. (2020). The enhancement of the proposed approach in improving security, irreversibility and reducing privacy leakage shows that the performance is maintained with an accuracy of 95.31%.

Table 5.6

EER comparison between Sandhya et al. (2020) and the proposed approach

Methods	FVC 2002 DB 1 (EER%)
Sandhya et al. (2020)	2.96
The proposed approach	1.54

The comparison between Sandhya et al. (2020) and the proposed approach for FVC 2002 DB1 databases is shown in Table 5.6. EER is defined as the point at which FAR is equal to the FRR. A lower EER value implies more efficiency.

5.6 Summary

Adversarial attacks are used to evaluate the system's security and privacy. Besides, the difficulty of retrieving a biometric feature is greater than the complexity of guessing a secret because all the bits of the feature are scrambled and concealed in the proposed approach. Furthermore, there is a significant amount of privacy leakage. Therefore, increasing the amount of the secret procedures can increase both the security and the irreversibility of the system while reducing privacy leakages.

Moreover, the entropy of the helper data has a major influence on the security of the proposed approach. For instance, modifying the biometric traits to increase the entropy of the helper data can increase the security. The selection of binary features, on the

other hand, has an impact on the overall level of security. During the feature selection process, it is critical to examine not just the dependability of the features, but also their entropy. Selecting features that are less interdependent can increase security.

Therefore, this chapter addressed the Research Objectives 3, which is to evaluate the proposed approach based on parameters such as security, privacy (irreversibility and privacy leakage), and biometric performance. The next chapter contains a general discussion of the findings to conclude the research work.

CHAPTER SIX

DISCUSSION, CONCLUSIONS AND FUTURE WORKS

6.1 Introduction

This chapter presents an overall discussion of the research conducted. Section 6.2 and its subsections further discuss the research by revisiting all of the research questions raised in Chapter 1. In this way, it is also shown that the research objectives have been

achieved. In Section 6.3, the contributions of this research are discussed. The next section, Section 6.4, provides potential follow-up research that may be done as a continuation of this study. Finally, the summary of this chapter is presented in Section 6.5.

6.2 Discussion

In this section, the discussion of this thesis is presented, which directly expresses the research's general conclusion. This section is separated into three sections depending on the research questions presented at the beginning of the thesis in order to facilitate the discussion. These research questions are revisited and further elaborated to provide a critical analysis that shows, in particular, that the research has successfully achieved its objectives.

6.2.1 Research Question (1)

What are the causes of dependency on biometric features in helper data?

This research question arose when the research identified the problem related to the dependency on biometric features in helper data. In order to better understand the problem, a literature search and analysis was conducted to determine what elements cause dependency on biometric features in helper data. Based on the analysis performed, the fingerprint features reflect the local properties of the attributes represented by the features. The bits that were retrieved from the biometric features are not all distributed independently. This is due to the necessity of binarizing real valued biometric features. Binarization is an important component of the helper data structure. For security reasons, the binarized features should be provided consistently and independently. In addition, these features should be robust against noise, as the

error correction algorithm's capacity to correct errors is limited. Binarization is used to extract a lengthy, uniformly and independently distributed bit string from biometric templates without significantly impacting verification performance.

Most biometric protection systems were developed to work with binary strings. Thus, the feature vector resulting from feature extraction must be translated into a binary string before further processing can occur. Then the feature extraction module takes binary features as input. As a result, the binary features inherit the dependency on the real-valued features, lowering the system's security.

When it comes to security and privacy evaluation, the probability distribution of biometric data is very important. Secure references should be randomly generated so that they cannot be predicted or linked. This is done for security reasons. Before the templates are derived, they go through the feature extraction method to generate the binary features. These are then bound by a biometric cryptography technique that generates the protected template.

In contrast, protected templates are produced from biometric data that is dependent on the template. Consequently, it is likely that the derived references include user-specific data and that the template protection is susceptible to linkage attacks, both of which are possible. On the other hand, biometric data distribution can aid an attacker in retrieving the original biometric data from protected templates.

There are numerous applications where the dependency on biometric traits is often overlooked, leading to the belief that the perceived level of security is significantly overstated. Hence, it is critical to conduct a thorough evaluation to further investigate whether the dependency issue can be reduced to increase security and privacy.

Answering Research Question (1) also contributes to achieving Research Objective (1), which has been discussed in detail in previous chapters. It also leads us to the next research question, which is discussed in more detail in the next section.

6.2.2 Research Question (2)

How can the dependency on biometric features in helper data be reduced?

This research has provided direction to further investigate how to reduce dependency on biometric features in helper data. In our proposed work, an enhanced keys binding technique based on the FCS has been proposed, which uses an additional cryptographic hash function to secure the provided biometric data. The advantage is that the scope of the comprehensive search can be extended. This is due to the inability of an attacker to employ the decoding techniques to find information about the user's biometric features in the helper data.

To strengthen the protection, the ECC encodes the binary feature vector. Combined with the additional cryptographic hash function used to hash the encoded binary feature vectors, the encrypted data also makes it difficult for an attacker to extract important biometric information about the individual. This also means that the dependency on biometric features in the helper data has been reduced. As a result, the proposed approach uses two levels of security: one through the hash binary feature vectors and another through the XOR function with the key. The details of the operation of the proposed approach were discussed in detail in Chapter 4, including the flow of the process and the algorithms. By answering this second research question, Research Objective (2) was achieved. This led the research to continue with the third

research question, which was meant to evaluate the proposed work. In the following subsection, Research Question (3) is discussed accordingly.

6.2.3 Research Question (3)

Can the proposed approach enhance the existing FCS to reduce the dependency on biometric features in helper data to decrease privacy leakage?

The last research question provides the direction of the research to validate the proposed work. In general, the proposed approach was evaluated from three perspectives: (1) security, (2) privacy, and (3) performance. From the security perspective, it is stated how difficult it is for an adversary to construct a datum that can be used to trick the verification process. The capacity to safeguard privacy shows how challenging it is to get biometric data and information on biometric data stored in a secured template. Two parameters are used to measure privacy: (1) irreversibility and (2) privacy leakage.

The proposed approach of protecting templates was evaluated experimentally with a focus on security and privacy. The details of the experimental work results were discussed in detail in Chapter 5. In summary, the secrecy of authentication with protected templates is based on the security of *h(Key)* and *h*(*fingercode*). In order to protect privacy, biometric data must be irreversible, since a breach of biometric data leads to a loss of biometric identity that is nearly impossible to get back. Leakage of personal information is critical to data minimization and has implications for

unlinkability as well as other factors. Both irreversibility and privacy leakage demonstrate an algorithm's ability to preserve biometric data. Based on the results, security increases with the amount of secret as the uncertainty of the secret increases.

Furthermore, the proposed approach was assessed in terms of performance. The performance seems to be acceptable as measured by FAR and FRR, which represent the security and user-friendliness of the system, respectively. Since security is more important than anything else, the setting for FAR should be 0% and the value for FRR should be as low as possible. In the proposed approach, FAR was almost 0%, but the FRR varied depending on the data and key length. Based on the experimental findings, the proposed approach performs better with an EER of 1.54% (a low EER value indicates better performance). Answering the third research question shows that the Research Objective (3) has been achieved. In the next section, the contribution of this research is highlighted.

6.3 Research Contribution

This section highlights the various contributions of this research. The contribution of this research can be divided into several perspectives based on the chapters and objectives. Chapter One has contributed to the identification of the gap and problem related to biometric template protection, whereas Chapter Two has added to the corpus of knowledge in the field of biometric template protection. Chapter Three has contributed to the key binding technique for biometric fingerprint template protection based on FCS, which was developed to create a secret binary feature to reduce the dependency on biometric features in helper data. Besides, this research focuses on a

BC that can be used in key binding techniques or as a template protection approach in fingerprint biometric systems by transforming templates into the secret binary domain.

Chapter Four has contributed to the concealment module. In particular, this research was able to construct FCS with a key length that can be used in other state-of-the-art cryptographic systems that meet sufficient security and privacy criteria by using an appropriate error-correcting approach that meets security standards. Based on Objective (2), this chapter has also helped to improve FCS in terms of technique and algorithm.

In addition, Chapter Five has contributed to the analysis of both security and privacy (i.e., irreversibility and privacy leakage), to illustrate the feasibility of the proposed approach to secure biometric information. Moreover, the proposed approach was also evaluated in terms of its performance, where the accuracy was found to be acceptable as measured by FAR and FRR. This chapter has also contributed to Objective (3).

Last but not least, the purpose of this exploratory study is not to confirm or disprove any beliefs or practices that already exist. But it does provide a fresh starting point for future research that may be carried out. Additionally, this study does not seek to offer a conclusive answer to the issues at hand; rather, it offers potential directions to pursue. We believe that good research expands the opportunity for interested parties to collaborate to enhance existing proposed countermeasures rather than providing a one-size-fits-all solution to a problem. The next section 6.4 talks about limitations of the research.

6.4 Limitations of the Research

There are several limitations of this study that should be acknowledged to clarify that the findings presented are interpreted in the context of this research and are not generalizable to some extent. First, the dataset for the experimental work was not collected from individual participants. Instead, it was obtained from the existing FVC 2002 dataset. The dataset contained samples fingerprint images captured by scanners. However, no detailed information was disclosed about the scanners used to capture the images for the dataset. Complete information about the dataset is necessary to avoid any ambiguity that could affect the interpretation of the findings obtained.

Second, the experimental work was performed in a simulated environment using MATLAB software. Although simulated results are widely accepted, they may vary in real-world implementation. Thus, this must be acknowledged as part of the limitations of this research. The next Section 6.5 discusses possible future work that could be done related to this research.

6.5 Future Works

This section provides recommendations for future research that can be implemented and provides a path and opportunity for researchers with similar interests. First, the proposed approach can be improved in implementation and evaluation with a real microcontroller and a fingerprint sensor. The microcontroller was designed to support the typical processing load of a processor and speed up the cryptographic functions.

Additionally, multimodal biometrics, which incorporate two distinct qualities, can be used. Examples include fingerprint and iris or fingerprint and face. One of the advantages of multimodal biometrics is the prevention of fraud mechanisms.

The proposed approach can be adapted to artificial intelligence (AI). This means that an intelligent agent can be provided to secure the biometric template. A set of rules, for example, is coded. The intelligent agent uses these rules to assess security-related events in the system. A set of information is collected and trained on a specific situation. Apart from that, AI and biometrics can precisely validate a person's identification based on their physiological and behavioral characteristics.

6.6 Summary

In this chapter, a detailed discussion was conducted to answer the research questions related to the aim and objectives of this research. Apart from that, the contribution of the research to the state of knowledge in the domain of biometric template protection was comprehensively discussed. Finally, recommendations for future work were proposed.

REFERENCES

Adamovic, S. Z., Milosavljevic, M., Veinovic, M. D., Sarac, M., & Jevremovic, A. (2017). Fuzzy commitment scheme for generation of cryptographic keys based on iris biometrics. *The Institution of Engineering and Technology IET*

Biometrics, *6*(2), 89–96. https://doi.org/10.1049/iet-bmt.2016.0061

Aithal, P. S., & Prasad, K. K. (2017). Fingerprint image segmentation: A review of state of the art techniques. *International Journal of Management, Tecnology, and Social Sciences (IJMTS)*, *2*(2), 28–39.

Al-Assam, H., & Jassim, S. (2012). Security evaluation of biometric keys. *Computers and Security*, *31*(2), 151–163. https://doi.org/10.1016/j.cose.2012.01.002

Al-Assam, H., Sellahewa, H., & Jassim, S. (2009). A lightweight approach for biometric template protection Hisham. *Proceedings of SPIE - The International Society for Optical Engineering*, *7351*(March), 1–12. https://doi.org/10.1117/12.818291

Al-Kuwari, S., Davenport, J. H., & Bradford, R. J. (2010). Cryptographic hash functions: recent design trends and security notions. *Short Paper Proceedings of 6th China International Conference on Information Security and Cryptology (Inscrypt '10)*, 133–150.

Ali, M. M. H., Mahale, V. H., Yannawar, P., & Gaikwad, A. . (2016). Fingerprint recognition for person identification and verification based on minutiae matching. *2016 IEEE 6th International Conference on Advanced Computing*, 332–339. https://doi.org/10.1109/IACC.2016.69

Ali, S. S., Baghel, V. S., Ganapathi, I. I., & Prakash, S. (2020). Robust biometric authentication system with a secure user template. *Image and Vision Computing*, *104*, 104004. https://doi.org/https://doi.org/10.1016/j.imavis.2020.104004

Ali, S. S., Ganapathi, I. I., & Prakash, S. (2018). Robust technique for fingerprint template protection. *The Institution of Engineering and Technology IET Biometrics*, *7*(6), 536–549. https://doi.org/10.1049/iet-bmt.2018.5070

Alsmirat, M. A., Al-Alem, F., Al-Ayyoub, M., Jararweh, Y., & Gupta, B. (2019). Impact of digital fingerprint image quality. *Multimed Tools Appl*, *78*(2019), 3649–3688. https://doi.org/doi.org/10.1007/s11042-017-5537-5

Alves, M. C. B., Drusinsky, D., & Shing, M. T. (2011). A practical formal approach for requirements validation and verification of dependable systems. *Proceedings - 5th Latin-American Symposium on Dependable Computing Workshops, LADCW 2011*, 47–51. https://doi.org/10.1109/LADCW.2011.14

Ankit, K., & Rekha, J. (2016). Biometrics as a cryptographic method for network security. *Indian Journal of Science and Technology*, *9*(22). https://doi.org/10.17485/ijst/2016/v9i22/95288

Arora, S., & Bhatia, M. P. S. (2021). Challenges and opportunities in biometric security: A survey. *Information Security Journal: A Global Perspective*, *31*(1), 24–48. https://doi.org/10.1080/19393555.2021.1873464

Ashish, M., & Sinha, G. (2017). Biometric template protection. *Journal of Biostatistics and Biometric Applications*, *1*(2), 1–8.

Azzoubi, E. A., & Ibrahim, R. B. (2015). An enhancement algorithm using gabor filter for fingerprint recognition. *Journal of Theoretical and Applied Information Technology 30th*, *74*(3), 355–363.

Babatunde, I. G. (2015). Fingerprint Matching Using Minutiae-Singular Points

Network. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, *8*(2), 375–388. https://doi.org/10.14257/ijsip.2015.8.2.35

Ballard, L., Lopresti, D., & Monrose, F. (2007). Forgery quality and its implications for behavioral biometric security. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *37*(5), 1107–1118. https://doi.org/10.1109/TSMCB.2007.903539

Barrero, M. G. (2016). *Improving Security and Privacy in Biometric Systems*. [Doctoral dissertation, Universidad Aut´onoma de Madrid]. EURASIP. https://theses.eurasip.org/theses/662/improving-security-and-privacy-in-biometric/

Bertoni, G., Daemen, J., Peeters, M., & Van Assche, G. (2013). Keccak. In T. Johansson & P. Q. Nguyen (Eds.), *Advances in Cryptology – EUROCRYPT 2013. Lecture Notes in Computer Science, vol 7881* (pp. 313–314). Springer, Berlin, Heidelberg. https://doi.org/https://doi.org/10.1007/978-3-642-38348-9_19

Borgianni, Y., & Maccioni, L. (2020). Review of the use of neurophysiological and biometric measures in experimental design research. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, *34*(2), 248–285. https://doi.org/DOI: 10.1017/S0890060420000062

Bose, P.-K., & Kabir, M.-J. (2017). Fingerprint : A unique and reliable method for identification. *Journal of Enam Medical College*, *7*(1), 29–34.

Bose, R. C., & Ray-Chaudhuri, D. (1960). On a class of error correcting binary. *Information and Control*, *3*(1), 68–79.

Brindha, V. E. (2012). Biometric template security using dorsal hand vein fuzzy vault. *Journal of Biometrics & Biostatistics*, *03*(04), 1000145–1000145. https://doi.org/10.4172/2155-6180.1000145

BSI. (2011). *Study of the Privacy and Accuracy of the Fuzzy Commitment Scheme BioKeyS III-Final Report*. http://www.bsi.bund.de

Cavoukian, A., & Stoianov, A. (2015). Biometric Encryption. In *Encyclopedia of Biometrics* (pp. 1–14). Springer.

Cavoukian, A., Stoianov, A., & Carter, F. (2008). Biometric encryption : Technology for strong authentication, security and privacy. *IFIP International Federation for Information Processing*, *261*, 57–77.

Chang, S., Perlner, R., Burr, W. E., Kelsey, J. M., & Bassham, L. E. (2012). Third-round report of the SHA-3 cryptographic hash algorithm competition. *National Institute of Standards and Technology*.

Chauhan, S., & Sharma, A. (2018). Securing fuzzy commitment scheme against decodability attack-based cross-matching. In S. K. D. I. Woungang (Ed.), *International Conference on Wireless, Intelligent, and Distributed Environment for Communication, Lecture Notes on Data Engineering and Communications Technologies 18*. Springer International Publishing AG.

Chauhan, S., & Sharma, A. (2019). Improved fuzzy commitment scheme. *International Journal of Information Technology (Singapore)*. https://doi.org/10.1007/s41870-018-0275-0

Dargan, S., & Kumar, M. (2020). A comprehensive survey on the biometric

recognition systems based on physiological and behavioral modalities. *Expert Systems with Applications*, *143*, 113114. https://doi.org/https://doi.org/10.1016/j.eswa.2019.113114

Dasso, A., & Funes, A. (2007). *Verification, Validation, and Testing in Software Engineering*. Idea Group Publishing.

Datta, P., Bhardwaj, S., Panda, S. N., Tanwar, S., & Badotra, S. (2020). Survey of security and privacy issues on biometric system. In B. B. Gupta, G. M. Perez, D. P. Agrawal, & D. Gupta (Eds.), *Handbook of Computer Networks and Cyber Security* (pp. 763–776). Springer International Publishing. https://doi.org/10.1007/978-3-030-22277-2_30

de Groot, J., Škorić, B., de Vreede, N., & Linnartz, J. P. (2016). Quantization in zero leakage helper data schemes. *Eurasip Journal on Advances in Signal Processing*, *2016*(1). https://doi.org/10.1186/s13634-016-0353-z

Dwivedi, R., Dey, S., Anand, M., & Apurv, S. (2019). A fingerprint based crypto - biometric system for secure communication. *Journal of Ambient Intelligence and Humanized Computing*. https://doi.org/10.1007/s12652-019-01437-5

Dwivedi, R., Dey, S., Sharma, M. A., & Goel, A. (2020). A fingerprint based crypto-biometric system for secure communication. *Journal of Ambient Intelligence and Humanized Computing*, *11*(4), 1495–1509. https://doi.org/10.1007/s12652-019-01437-5

FVC2002. (2002). *FVC2002*. Second International Competition for Fingerprint Verification Algorithms. http://bias.csr.unibo.it/fvc2002/databases.asp

Galar, M., Derrac, J., Peralta, D., Triguero, I., Paternain, D., Lopez-molina, C., García, S., Benítez, J. M., Pagola, M., Barrenechea, E., Bustince, H., & Herrera, F. (2015). Knowledge-based systems a survey of fingerprint classification part I : Taxonomies on feature extraction methods and learning models. *Knowledge-Based Systems*, *February*, 0950–7051. https://doi.org/10.1016/j.knosys.2015.02.008

Galbally, J., Fierrez, J., & Cappelli, R. (2019). An introduction to fingerprint presentation attack detection. In S. M. et Al. (Ed.), *Handbook ofBiometric Anti-Spoofing, Advances in Computer Vision and Pattern Recognition* (pp. 3–31). Springer Nature Switzerland AG. https://doi.org/10.1007/978-3-319-92627-8

Geng, S., Giannopoulou, G., & Kabir-Querrec, M. (2019). Privacy protection in distributed fingerprint-based authentication. *Proceedings of the 18th ACM Workshop on Privacy in the Electronic Society - WPES'19*, 1–13.

Gilkalaye, B. P., Rattani, A., & Derakhshani, R. (2019). Euclidean-distance based fuzzy commitment scheme for biometric template security. *2019 7th International Workshop on Biometrics and Forensics, IWBF 2019*, Cancun, Mexico 1–6. https://doi.org/10.1109/IWBF.2019.8739177

Giot, R., El-Abed, M., & Rosenberger, C. (2013). Fast computation of the performance evaluation of biometric systems: Application to multibiometrics. *Future Generation Computer Systems*, *29*(3), 788–799. https://doi.org/10.1016/j.future.2012.02.003

Gomez-barrero, M., & Galbally, J. (2019). Reversing the irreversible : A survey on inverse biometrics. *Computers & Security*.

https://doi.org/10.1016/j.cose.2019.101700

Grigorescu, A., Boche, H., & Schaefer, R. F. (2017). Robust biometric authentication from an information theoretic perspective. *Entropy*, *19*(9), 1–25. https://doi.org/10.3390/e19090480

Gunjan, V. K., Prasad, P. S., & Mukherjee, S. (2020). Biometric template protection scheme-cancelable biometrics. In A. Kumar & S. Mozar (Eds.), *ICCCE 2019* (pp. 405–411). Springer Singapore.

Hasan, H., & Abdul-Kareem, S. (2013). Fingerprint image enhancement and recognition algorithms : A survey. *Neural Comput & Applic*, *23*(2013), 1605–1610. https://doi.org/10.1007/s00521-012-1113-0

Hocquenghem, A. (1959). Codes correcteurs d'erreurs. *Chiffres*, *2*, 147–156.

Ignatenko, T., & Willems, F. M. J. (2010). Information leakage in fuzzy commitment schemes. *IEEE Transactions on Information Forensics and Security*, *5*(2), 337–348. https://doi.org/10.1109/TIFS.2010.2046984

Ilchenko, M., Uryvsky, L., & Globa, L. (2020). *Advances in information and communication technology and systems*. Springer International Publishing. https://books.google.com.my/books?id=i3r7DwAAQBAJ

ISO/IECJTC1. (2021). *Biometrics*. https://committee.iso.org/home/jtc1sc37

Jain, A. K., Nandakumar, K., & Ross, A. (2016). 50 years of biometric research : Accomplishments, challenges, and opportunities. *Pattern Recognition Letters*, *79*(2016), 80–105. https://doi.org/10.1016/j.patrec.2015.12.013

Jain, A. K., Prabhakar, S., Hong, L., & Pankanti, S. (2000). Filterbank-based fingerprint matching. *IEEE Transactions on Image Processing*, *9*(5), 846–859. https://doi.org/10.1109/83.841531

Jayapal, R. (2017). *Biometric encryption system for increased security*. [Graduate Theses and Dissertations, University of North Florida]. UNF Digital Commons. https://digitalcommons.unf.edu/etd/746

Jegede, A., Udzir, N. I., Abdullah, A., & Mahmod, R. (2017). State of the art in biometric key binding and key generation schemes. *International Journal of Communication Networks and Information Security*, *9*(3), 333–344.

Jin, Z., Teoh, A. B. J., Goi, B. M., & Tay, Y. H. (2016). Biometric cryptosystems: A new biometric key binding and its implementation for fingerprint minutiae-based representation. *Pattern Recognition*. https://doi.org/10.1016/j.patcog.2016.02.024

Joshi, M., Mazumdar, B., & Dey, S. (2020). A comprehensive security analysis of match-in-database fingerprint biometric system. *Pattern Recognition Letters*, *138*, 247–266. https://doi.org/https://doi.org/10.1016/j.patrec.2020.07.024

Jothi, R. A. (2018). Analysis of fingerprint minutiae extraction and matching algorithm. *International Journal of Advanced Research Trends in Engineering and Technology (IJARTET), 3*(20), 398-402.

Juels, A., & Wattenberg, M. (1999). A fuzzy commitment scheme. *In Proceedings of the 6th ACM Conference on Computer and Communications Security, CCS '99,* 28–36, New York, NY, USA.

Kapoor, I., & Sharma, P. (2016). Home security using biometric sensor technology. *International Journal of Control Theory and Applications*, *9*(17), 8407–8413.

Kelkboom, E. J. C. (2010). *On the performance of helper data template protection schemes*. [Doctorial Dissertations, University of Twente, the Netherlands]. doi.org/10.3990/1.9789036530743

Kumar, G., Tulyakov, S., & Govindaraju, V. (2010). Combination of symmetric hash functions for secure fingerprint matching. *Proceedings - International Conference on Pattern Recognition*, 890–893. Istanbul, Turkey. https://doi.org/10.1109/ICPR.2010.224

Laban, M., & Drutarovsky, M. (2020). Leakage free helper data storage in microcontroller based PUF implementation. *Microprocessors and Microsystems*, *October*, 103369. https://doi.org/10.1016/j.micpro.2020.103369

Lafkih, M., Mikram, M., Ghouzali, S., El Haziti, M., & Aboutajdine, D. (2016). Biometric cryptosystems based fuzzy commitment scheme: A security evaluation. *International Arab Journal of Information Technology*, *13*(4), 443–449.

Liang, B., Wu, Z., & You, L. (2014). A novel fingerprint-based biometric encryption. *2014 Ninth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, 146–150. https://doi.org/10.1109/3PGCIC.2014.48

Lutsenko, M., Kuznetsov, A., Kiian, A., Smirnov, O., & Kuznetsova, T. (2021). Biometric cryptosystems: Overview, state-of-the-art and perspective directions. In M. Ilchenko, L. Uryvsky, & L. Globa (Eds.), *Advances in Information and Communication Technology and Systems* (pp. 66–84). Springer International

Publishing.

Macek, N., Franc, I., Bogdanoski, M., & Aca, A. (2018). Biometric cryptosystems – approaches to biometric key-binding and key- generation. *The 10th International Conference on Business Information Security (BISEC-2018)*, 16–19.

MacKay, D. J. . (2005). *Information theory, inference, and learning algorithms* (4th ed.). Cambridge University Press. https://doi.org/10.1166/asl.2012.3830

Maetouq, A., Daud, S. M., Ahmad, N. A., Maarop, N., Sjarif, N. N. A., & Abas, H. (2018). Comparison of hash function algorithms against attacks: A review. *International Journal of Advanced Computer Science and Applications*, *9*(8), 98–103. https://doi.org/10.14569/ijacsa.2018.090813

Maiorana, E., La Rocca, D., & Campisi, P. (2016). On the permanence of EEG signals for biometric recognition. *IEEE Transactions on Information Forensics and Security*, *11*(1), 163–175. https://doi.org/10.1109/TIFS.2015.2481870

Malek, M. (2006). Hadamard Codes. *Coding Theory*, 1–8.

Maltoni, D. (2005). A tutorial on fingerprint recognition. *Advanced Studies in Biometrics*, *3161/2005*, 121–138. https://doi.org/10.1007/11493648_3

Maltoni, D., Jain, A. K., Maio, D., & Prabhakar, S. (2009). Handbook of fingerprint. In *Springer* (Second Edi). Springer-Verlag London.

Maltoni, D., Maio, D., Jain, A. K., & Prabhakar, S. (2003). *Handbook of Fingerprint Recognition* [First Ed.], Springer Science & Business Media. https://doi.org/10.1109/MEI.2004.1342443

Manikpuri, M. M. (2017). *Biometric security systems for beginner*. Educreation Publishing. https://books.google.com.my/books?id=Qrw6DwAAQBAJ

Marasco, E., & Ross, A. (2014). A survey on antispoofing schemes for fingerprint recognition systems. *ACM Computing Surveys*, *47*(2), 1–36. https://doi.org/10.1145/2617756

Mehmood, R., & Selwa, A. (2019). Fingerprint biometric template security schemes: attacks and countermeasures. In T. S. Singh P., Kar A., Singh Y., Kolekar M. (Ed.), *Proceedings of ICRIC 2019. Lecture Notes in Electrical Engineering* (pp. 455–467). Springer, Cham.

Memon, S., Sepasian, M., & Balachandran, W. (2008). Review of finger print sensing technologies. *IEEE INMIC 2008: 12th IEEE International Multitopic Conference - Conference Proceedings*, 226–231. Karachi, Pakistan https://doi.org/10.1109/INMIC.2008.4777740

Menezes, A. J., Oorschot, P. C. Van, & Vanstone, S. A. (1997). *Handbook of Applied Cryptography*. CRC Press. https://doi.org/10.1.1.99.2838

Mirza, M. (2014). Vulnerabilities in biometric authentication and their countermeasures: A secure and efficient authentication system. *International Journal of Information Technology and Electrical Engineering*, *3*(3), 36–41. http://www.iteejournal.org/archive/vol3no3/v3n3_7.pdf

Moreno, P., Bernardino, A., & Santos-Victor, J. (2005). Gabor parameter selection for local feature detection. *Lecture Notes in Computer Science*, *3522*(I), 11–19. https://doi.org/10.1007/11492429_2

Mucchi, L., Nizzi, F., Pecorella, T., Fantacci, R., & Esposito, F. (2019). Benefits of physical layer security to cryptography : Tradeoff and applications. *2019 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, 1–3.

Müftüoğlu, Z., & Yildirim, T. (2019). Comparative analysis of crypto systems using biometric key. *Procedia Computer Science*, *154*, 327–331. https://doi.org/10.1016/j.procs.2019.06.047

Mwema, J., Kimwele, M., & Kimani, S. (2015). A simple review of biometric template protection schemes used in preventing adversary attacks on biometric fingerprint templates. *International Journal of Computer Trends and Technology*, *20*(1), 12–18. https://doi.org/10.14445/22312803/IJCTT-V20P103

Nagar, A., Nandakumar, K., & Jain, A. K. (2010). Biometric template transformation: A security analysis. *Proceedings of SPIE - The International Society for Optical Engineering*, *7541*, Media Forensics and Security II, 1-15, San Jose, California, United States. https://doi.org/10.1117/12.839976

Nandakumar, K., & Jain, A. K. (2015). Biometric template protection: bridging the performance gap between theory and practice. *IEEE Signal Processing Magazine*, *32*(5), 88–100. https://doi.org/10.1109/MSP.2015.2427849

National Institute of Standards and Technology. (2015). SHA-3 standard : Permutation-based hash and extendable-output functions. *NIST Federal Information Processing Standard*, *August*.

Nezhad, S. Y. D., Safdarian, N., & Zadeh, S. A. H. (2020). New method for fingerprint images encryption using DNA sequence and chaotic tent map. *Optik*

- *International Journal for Light and Electron Optics*, *S0030-4026*(20), 1–15. https://doi.org/10.1016/j.ijleo.2020.165661

Nilsson, K., & Bigun, J. (2003). Prominent symmetry points as landmarks in fingerprint images for alignment. *16th International Conference on Pattern Recognition (ICPR'02)*, 395–398.

Obaidat, M. S., Rana, S. P., Maitra, T., Giri, D., & Dutta, S. (2019). Biometric security and internet of things (IoT). In M. S. Obaidat, S. P. Rana, T. Maitra, D. Giri, & S. Dutta (Eds.), *Biometric-Based Physical and Cybersecurity Systems* (pp. 577–509). Springer Nature Switzerland AG 2019. https://doi.org/https://doi.org/10.1007/978-3-319-98734-7_19

Pagnin, E., & Mitrokotsa, A. (2017). Privacy-preserving biometric authentication : challenges and directions. *Security and Communication Networks*, *2017*(7129505), 1–9.

Patel, R., & Ramalingam, S. (2019). Advances in fingerprint technology. In Obaidat et al. (Ed.), *Biometric-Based Physical and Cybersecurity Systems*. Springer.

Pellikaan, R., & Wu, X. (2012). *Error-correcting codes and cryptology*. Cambridge University Press.

Peter, S., Reddy, B. P., Momtaz, F., & Givargis, T. (2016). Design of secure ECG-based biometric authentication in body area sensor networks. *Sensors*, *16*(4). https://doi.org/10.3390/s16040570

Prashant P. Pittalia. (2019). A comparative study of hash algorithms in cryptography. *International Journal of Computer Science and Mobile Computing*, *8*(6), pg.147

– 152. https://ijcsmc.com/docs/papers/June2019/V8I6201928.pdf

Precise Biometrics AB. (2014). *Understanding biometrics performance evaluation [White paper]*. 1–4.

Premasathian, N. (2013). A multiple fuzzy commitment scheme. *International Conference on Computer Applications Technology, ICCAT 2013*. https://doi.org/10.1109/ICCAT.2013.6521958

Priya, S. S. S., Karthigaikumar, P., & Mangai, N. M. S. (2014). Mixed random 128 bit key using finger print features and binding key for AES algorithm. *Proceedings of 2014 International Conference on Contemporary Computing and Informatics, IC3I 2014*, *November 2014*, 1226–1230. https://doi.org/10.1109/IC3I.2014.7019656

Rane, M., Latne, T., & Bhadade, U. (2020). Biometric recognition using fusion. In A. Kumar, M. Paprzycki, & V. K. Gunjan (Eds.), *Lecture Notes in Electrical Engineering*. Springer Nature Singapore Pte Ltd. 2020. https://doi.org/https://doi.org/10.1007/978-981-15-1420-3_142

Ratha, N. K., Connell, J. H., & Bolle, R. M. (2003). Biometrics break-ins and band-aids. *Pattern Recognition Letters 24*(13), 2105–2113. https://doi.org/10.1016/S0167-8655(03)00080-1

Rathgeb, C., & Uhl, A. (2011). A survey on biometric cryptosystems and cancelable biometrics. *EURASIP Journal on Information Security*, *2011*(3), 1–25. https://doi.org/10.1186/1687-417X-2011-3

Recogtech. (2022). *FAR and FRR: security level versus user convenience*.

https://www.recogtech.com/en/knowledge-base/security-level-versus-user-convenience

Reed, I. ., & Solomon, G. (1960). Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, *8*(2), 300–304.

Riaz, N., Riaz, A., & Khan, S. A. (2017). Biometric template security: An overview. *Sensor Review*, *38*(1), 120–127. https://doi.org/10.1108/SR-07-2017-0131

Riccio, D., Galdi, C., & Manzo, R. (2016). Biometric / cryptographic keys binding based on function minimization. *International Conference on Signal-Image Technology & Internet-Based Systems Biometric/Cryptographic*. https://doi.org/10.1109/SITIS.2016.31

Rinaldi, A. (2016). Biometrics' new identity—measuring more physical and biological traits. *Science & Society*, *17*(1), 22–26.

Rivest, R. L. (1992). RFC 1321: The MD5 message-digest algorithm. In *MIT Laboratory for Computer Science and RSA Data Security, Inc 19*, 709–715. https://doi.org/10.20595/jjbf.19.0_3

Ross, A., Banerjee, S., & Chowdhury, A. (2020). Security in smart cities: A brief review of digital forensic schemes for biometric data. *Pattern Recognition Letters*, *138*, 346–354. https://doi.org/https://doi.org/10.1016/j.patrec.2020.07.009

Sabhanayagam, T., Venkatesan, V. P., & Senthamaraikannan, K. (2018). A comprehensive survey on various biometric systems. *International Journal of Applied Engineering Research*, *13*(5), 2276–2297.

Sadhya, D., Singh, S. K., & Chakraborty, B. (2016). Review of key-binding-based biometric data protection schemes. *IET Biometrics*, *5*(4), 263–275. https://doi.org/10.1049/iet-bmt.2015.0035

Sahani, M., Nanda, C., Sahu, A. K., & Pattnaik, B. (2015). Web-based online embedded door access control and home security system based on face recognition. *2015 International Conference on Circuit, Power and Computing Technologies [ICCPCT]*.

Sandhya, M., Dileep, M., Murthy, A. N., & Misbahuddin, M. (2020). Fingerprint cryptosystem using variable selection of minutiae points. In *Data Engineering and Communication Technology, Advances in Intelligent Systems and Computing, 1079*, 359–369. Springer Singapore. https://doi.org/10.1007/978-981-15-1097-7

Sandhya, M., & Prasad, M. V. N. K. (2017). Biometric template protection : A systematic literature review of approaches and modalities. In R. et al Jiang (Ed.), *Biometric Security and Privacy*. https://doi.org/10.1007/978-3-319-47301-7

Sapkal, S., & Deshmukh, R. R. (2016). Biometric template protection with fuzzy vault and fuzzy commitment. *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*, Udaipur, India.1-6. https://doi.org/10.1145/2905055.2905118

Sarier, N. D. (2016). Efficient biometric-based encryption for fingerprints. *The 11th International Conference for Internet Technology and Secured Transactions*. 127–132.

Sarkar, A. (2018). Cryptographic key generation from cancelable fingerprint templates. *2018 4th International Conference on Recent Advances in Information Technology (RAIT)*, 1–6.

Sarkar, A., & Singh, B. K. (2020). A review on performance, security and various biometric template protection schemes for biometric authentication systems. *Multimedia Tools and Applications*, *79*(37–38), 27721–27776. https://doi.org/10.1007/s11042-020-09197-7

Scheirer, W. J., & Boult, T. E. (2007). Cracking fuzzy vaults and biometric encryption. *Biometrics Symposium 2007*, 1–6.

Segun, O. F., Florence, B. M., & Olawale, F. B. (2020). How secured is the securer: Biometric technology overview. *International Journal of Computer Trends and Technology*, *68*(8), 39–43. https://doi.org/10.14445/22312803/ijctt-v68i8p106

Shelton, J., Dozier, G., Adams, J., & Alford, A. (2012). Permutation-based biometric authentication protocols for mitigating replay attacks. *2012 IEEE Congress on Evolutionary Computation*, Brisbane, Australia, 1–5. https://doi.org/10.1109/CEC.2012.6253011

Shirai, T., & Shibutani, K. (2003). On the diffusion matrix employed in the Whirlpool hashing function. *NESSIE Public Report*.

Shukla, S., & Patel, S. J. (2021). Securing fingerprint templates by enhanced minutiae-based encoding scheme in fuzzy commitment. *IET Information Security*, *15*(3), 256–266. https://doi.org/10.1049/ise2.12024

Simoens, K., Tuyls, P., & Preneel, B. (2009). Privacy Weaknesses in Biometric

Sketches. *30th IEEE Symposium on Security and Privacy.* Oakland, CA, USA

Siswanto, A., Katuk, N., & Ku-Mahamud, K. R. (2018). Fingerprint template protection schemes: A literature review. *Journal of Theoretical and Applied Information Technology*, *96*(10), 2764–2781.

Sklar, B. (2020). Reed-Solomon codes. *Applied Abstract Algebra with Maple™ and Matlab®*, *1*(3), 137–172. https://doi.org/10.1201/b19010-5

SQL Management Suite, I. (2021). The trade-off between database security and database performance. In *IDERA, Inc* (pp. 1–6).

Stevens, M., Bursztein, E., Karpman, P., Albertini, A., & Markov, Y. (2017). The first collision for full SHA-1. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *10401 LNCS*, 570–596. https://doi.org/10.1007/978-3-319-63688-7_19

Suzuki, H., Takeda, M., Obi, T., Yamaguchi, M., Ohyama, N., & Nakano, K. (2014). Encrypted sensing for enhancing security of biometric authentication. *2014 13th Workshop on Information Optics, WIO 2014*, 1–3. https://doi.org/10.1109/WIO.2014.6933292

Tantubay, N., & Bharti, J. (2020). A survey of biometric key-binding biocrypto-system using different techniques. *International Journal on Emerging Technologies*, *11*(1), 421–432.

Teoh, A. J., & Kim, J. (2015). Error correction codes for biometric cryptosystem : An Overview. *오류정정부호의 응용 Yonsei University*. 39–49.

Trivedi, A. K., Thounaojam, D. M., & Pal, S. (2020). Non-invertible cancellable fingerprint template for fingerprint biometric. *Computers & Security*, *90*(101690), 1–11. https://doi.org/10.1016/j.cose.2019.101690

Turakulovich, K. Z., Ugli, I. S. Z., Menglimuratovich, A. O., & Mardiyev Ulugbek Rasulovich. (2018). A practical implementation of fingerprint based fuzzy commitment scheme. *Section 10. Mechanical engineering.* (pp. 105–109).

Van De Haar, H., Van Greunen, D., & Pottas, D. (2013). The characteristics of a biometric. *2013 Information Security for South Africa - Proceedings of the ISSA 2013 Conference*, Johannesburg, *South Africa*, (pp. 1–8). https://doi.org/10.1109/ISSA.2013.6641037

Vats, S., Kaur, H., & Geetu. (2016). A comparative study of different biometric features. *International Journal of Advanced Research in Computer Science*, *7*(6), 169–171.

Velciu, M. A., Patrascu, A., & Patriciu, V. V. (2014). Bio-cryptographic authentication in cloud storage sharing. *SACI 2014 - 9th IEEE International Symposium on Applied Computational Intelligence and Informatics,* Timisoara, Romania, (pp.165–170). https://doi.org/10.1109/SACI.2014.6840054

Verlinde, P. (2003). Error detecting and correcting codes. *Encyclopedia of Information Systems*, *2*, 212–228. https://doi.org/10.1201/b17011-26

Vorobyeva, I., Guriel, D., Ferguson, M., & Oladapo, H. (2014). Benefits and issues of biometric technologies. *IEEE* Southeastcon. *2014*, Lexington, KY, (pp. 1–8). https://doi.org/10.1109/SECON.2014.6950706

Wallace, D. R. D. R., & Fujii, R. U. R. U. U. (1989). Software verification and validation: an overview. *IEEE Software*, *6*(3), 10–17. https://doi.org/10.1109/52.28119

Wang, N., Li, Q., Abd, A. A., Peng, J., Yan, X., & Niu, X. (2014). A novel template protection scheme for multibiometrics based on fuzzy commitment and chaotic system. *Signal Image and Video Processing*, *9*, 99-109. https://doi.org/10.1007/s11760-014-0663-2

Yager, N., & Amin, A. (2004a). Fingerprint classification: A review. *Pattern Analysis and Applications*, *7*(1), 77–93. https://doi.org/10.1007/s10044-004-0204-7

Yager, N., & Amin, A. (2004b). Fingerprint verification based on minutiae features: A review. *Pattern Analysis and Applications*, *7*(1), 94–113. https://doi.org/10.1007/s10044-003-0201-2

Yalçin, T., & Kavun, E. B. (2013). On the implementation aspects of sponge-based authenticated encryption for pervasive devices. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *7771 LNCS*, 141–157. https://doi.org/10.1007/978-3-642-37288-9_10

Yan, X., Lu, Y., Chen, Y., & Lu, C. (2017). Secret image sharing based on error-correcting codes. *2017 IEEE 3rd International Conference on Big Data Security on Cloud*, 86–89. China. (pp. 86–89). https://doi.org/10.1109/BigDataSecurity.2017.19

Yang, W., Wang, S., Hu, J., Zheng, G., & Valli, C. (2019). Security and Accuracy of

Fingerprint-based biometrics: A review. *Symmetry*, *11*(141), 1–19. https://doi.org/10.3390/sym11020141

Yang, Y., Yu, J., Zhang, P., & Wang, S. (2015). A fingerprint encryption scheme based on irreversible function and secure authentication. *Computational and Mathematical Methods in Medicine*, *2015*(673867), 1-10. https://doi.org/10.1155/2015/673867

Yasuda, M., Shimoyama, T., Abe, N., Yamada, S., Shinzaki, T., & Koshiba, T. (2016). Privacy-preserving fuzzy commitment for biometrics via layered error-correcting codes. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9482*, 117–133. https://doi.org/10.1007/978-3-319-30303-1_8

Zheng, G., Fang, G., Orgun, M. A., & Shankaran, R. (2015). A comparison of key distribution schemes using fuzzy commitment and fuzzy vault within wireless body area networks. *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, *2015-Decem*, 2120–2125. https://doi.org/10.1109/PIMRC.2015.7343648

Zhou, X. (2012). Privacy and security assessment of biometric template protection. *Itit*, *54*(4), 197–200. https://doi.org/10.1524/itit.2012.0676

Zhou, X., Kuijper, A., & Busch, C. (2012). Retrieving secrets from iris fuzzy commitment. *Proceedings - 2012 5th IAPR International Conference on Biometrics, ICB 2012*, 238–244. https://doi.org/10.1109/ICB.2012.6199814

Zhou, X., Kuijper, A., Veldhuis, R., & Busch, C. (2011). Quantifying privacy and security of biometric fuzzy commitment. *2011 International Joint Conference*

on Biometrics (IJCB), Washington, DC, USA, (pp. 1-8). doi: 10.1109/IJCB.2011.6117543.

APPENDIX A

PROTOTYPE INTERFACES

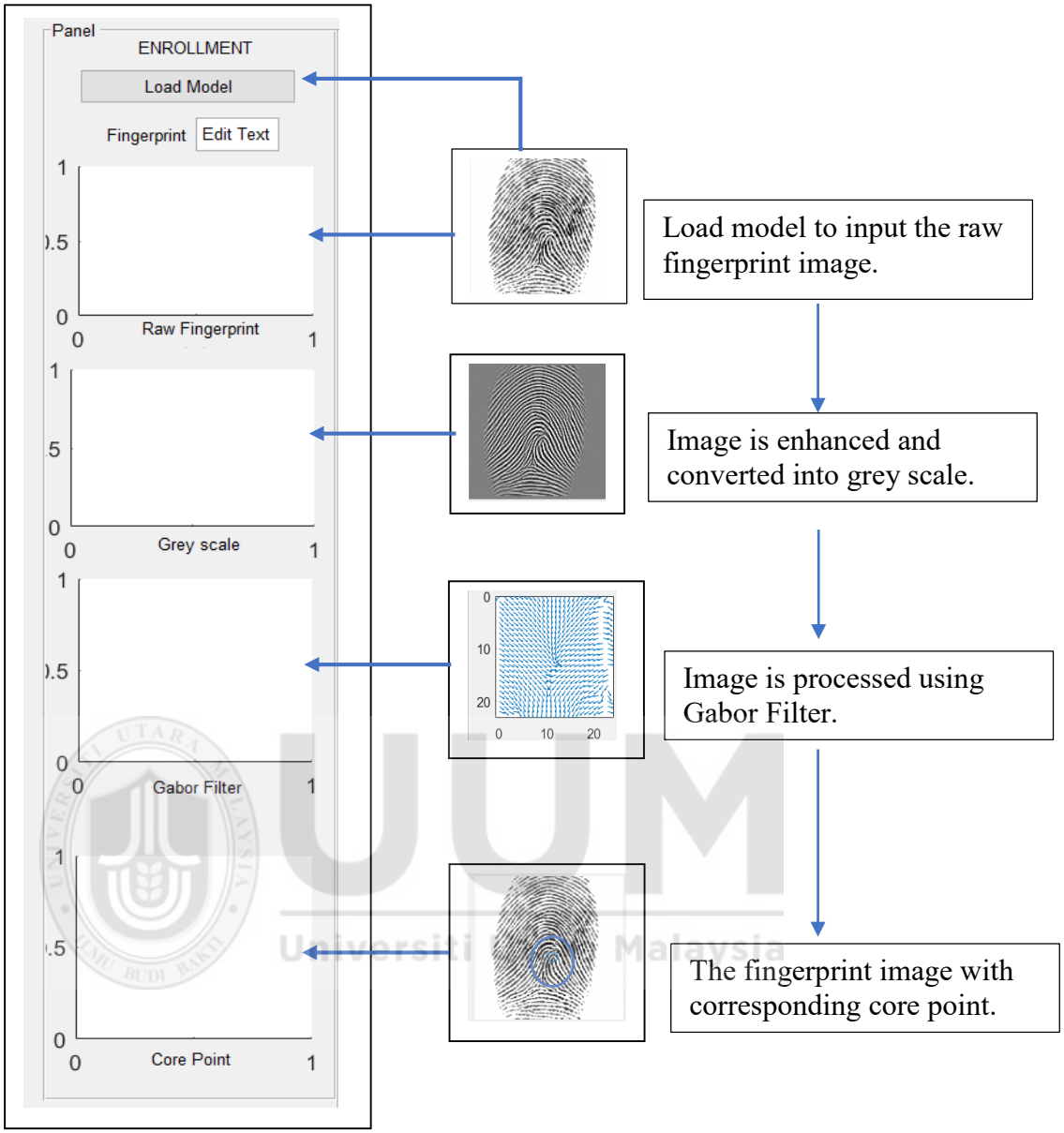


Figure 1: Example of prototype interface in Enrollment module

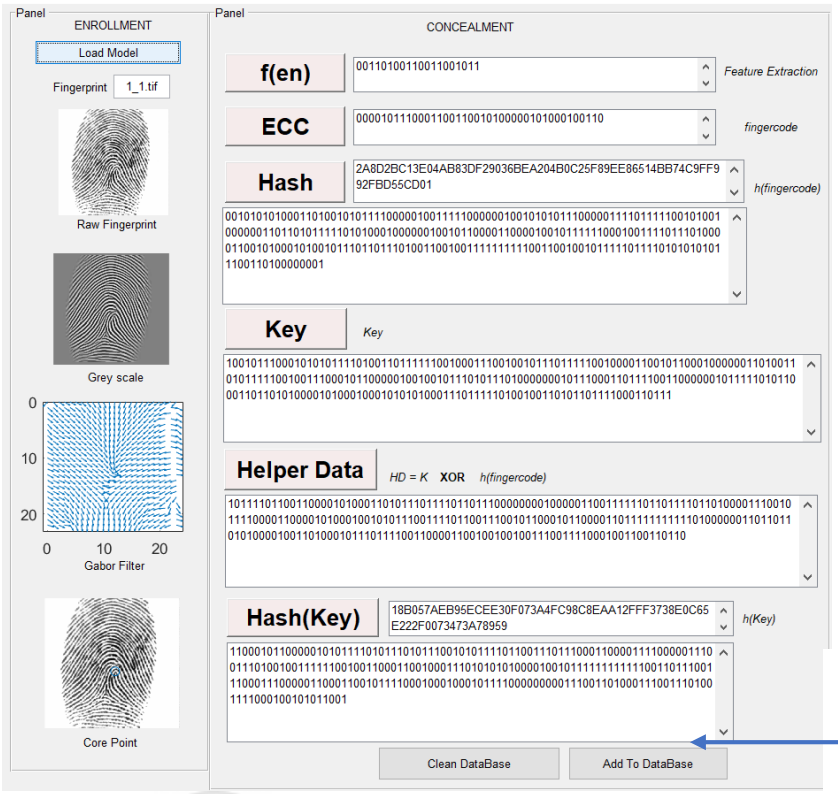


Figure 2: Example of prototype interface for Enrollment module and Concealment module

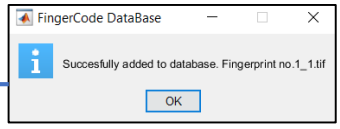


Figure 3: A message box indicates that the *HelperData* and *Key* of Fingerprint been successfully added to the database.

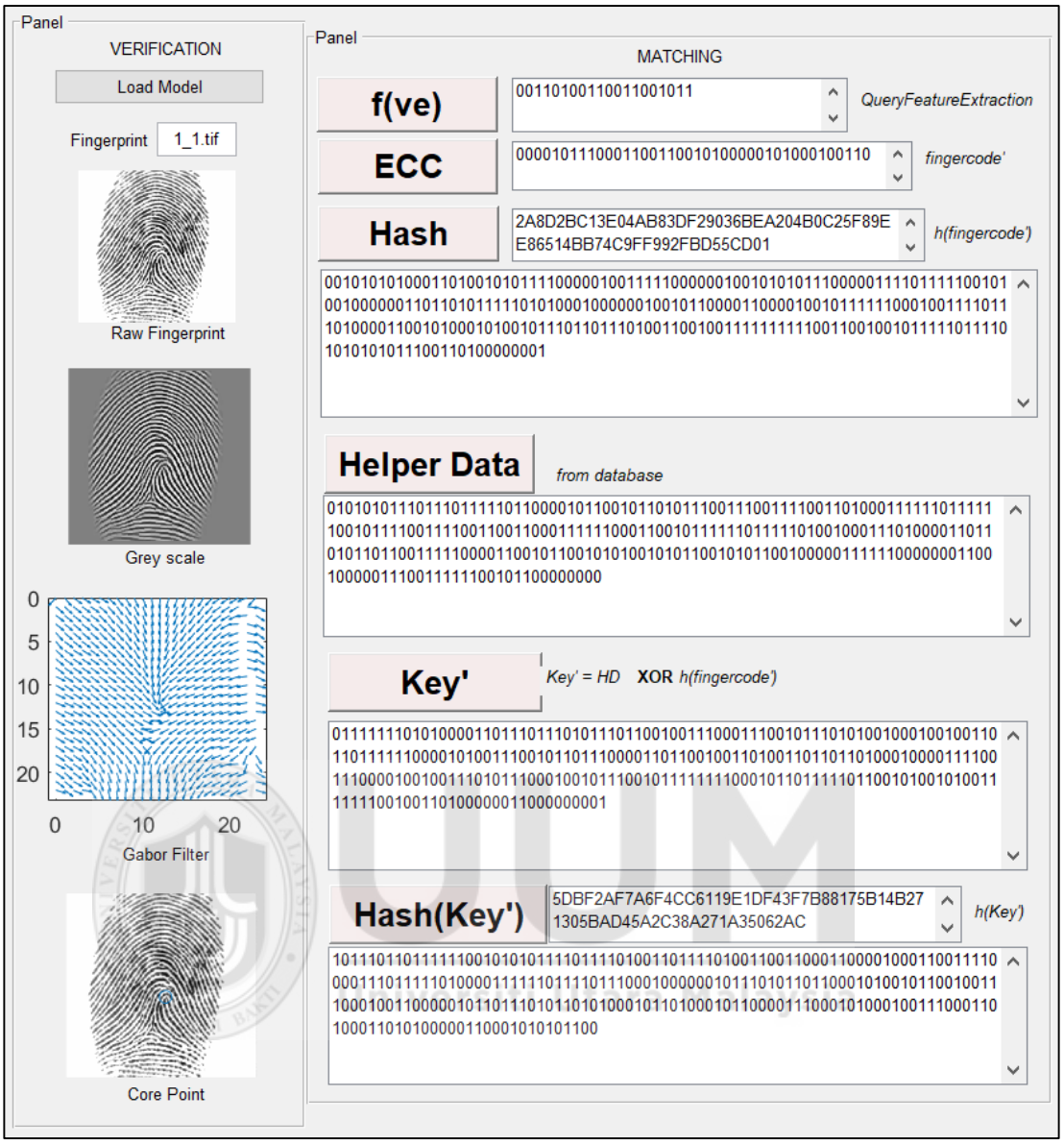


Figure 4: Prototype Interface: Verification panels

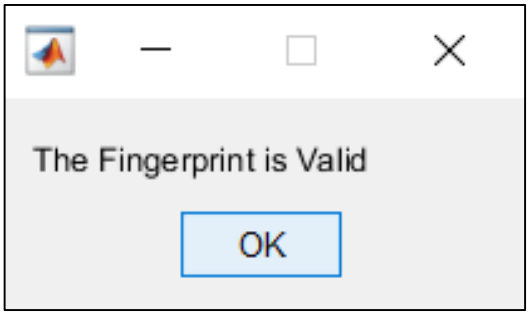


Figure 5: A message box to indicate that process matching is successful

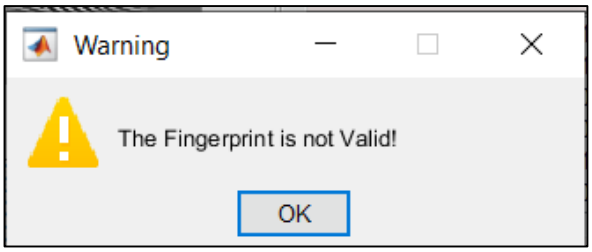


Figure 6: A message box to indicate that process matching is unsuccessful (rejected) when *h(Key)* is not equal to h*(Key')*.

APPENDIX B

SOURCE CODE OF THE PROTOTYPE

```
function varargout = FP3(varargin)
% FP3 MATLAB code for FP3.fig
%      FP3, by itself, creates a new FP3 or raises the
existing
%      singleton*.
%
%      H = FP3 returns the handle to a new FP3 or the handle
to
%      the existing singleton*.
%
%      FP3('CALLBACK',hObject,eventData,handles,...) calls the
local
%      function named CALLBACK in FP3.M with the given input
arguments.
%
%      FP3('Property','Value',...) creates a new FP3 or raises
the
%      existing singleton*.  Starting from the left, property
value pairs are
%      applied to the GUI before FP3_OpeningFcn gets called.
An
%      unrecognized property name or invalid value makes
property application
%      stop.  All inputs are passed to FP3_OpeningFcn via
varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI
allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help FP3



gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @FP3_OpeningFcn, ...
                   'gui_OutputFcn',  @FP3_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
```

```
    [varargout{1:nargout}] = gui_mainfcn(gui_State,
varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before FP3 is made visible.
function FP3_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)
% varargin   command line arguments to FP3 (see VARARGIN)
if (exist('FingerPrint_database.dat')==2)
            load('FingerPrint_database.dat','-mat');

 %disp('*********   Database Info   *********');


            %set(handles.edit2,'String',w);
            %disp('Number of fingerprints present in
database:');
            %w='Number of fingerprints present in database:';
           % set(handles.edit2,'String',w);
            %disp(fp_number);
            w='                *********   Database Info
*********          ';
            %set(handles.edit2,'String',w);
            %disp(fp_number);
           % disp('List of fingerprints added and
corresponding directories');
            for ii=1:fp_number
                lengthname = length (namefile_vector {ii});
                 addedspaces = 30 - lengthname;
                 if addedspaces <0
                     additionalspaces = 3;
                 end
                 emptystring = '-';
                 stringadd = '';
                 for jj = 1: addedspaces
                     stringadd = strcat (stringempty,
stringadd);
                 end
                 % disp ([namefile_vector {ii}, stringadd,
path_vector {ii}]);
%set(handles.edit3,'String',fp_number);

            end
        else
            %disp('Database is empty');
            w='Database is Clenar';
           % set(handles.edit2,'String',0);
```

```
        end
% Choose default command line output for FP3
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes FP3 wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command
line.
function varargout = FP3_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see
VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)
global pathname;
global namefile;
global img;
global selected;
global xor25;
global str26;
global arr26;
global arr29;
global hashStr;
global hashStr2;
global str24;
global str42;

clc;
        selected=0;


[namefile,pathname]=uigetfile({'*.bmp;*.tif;*.tiff;*.jpg;*.jpe
g;*.gif;*.pgm','IMAGE Files
(*.bmp,*.tif,*.tiff,*.jpg,*.jpeg,*.gif,*.pgm)'},'Chose
GrayScale Image');
        if namefile~=0
            [img,map]=imread(strcat(pathname,namefile));
```

```
            axes(handles.axes2);
            imshow(img);
            [oimg,fimg,bwimg,eimg,enhimg] =
fft_enhance_cubs(img);
             axes(handles.axes3);
            imshow(enhimg);
            [gimg,oimg] = orientation_image_rao(img);
             axes(handles.axes4);
             view_orientation_image(oimg);
             %display(oimg)
              [oimg,fimg,bwimg,eimg,enhimg] =
fft_enhance_cubs(img);
                [xc,yc]=supercore7(enhimg);
                %figure('Name','Input fingerprint and core
point');
                 axes(handles.axes5);
                 binary = dec2bin([xc,yc],10);
                binary_t=transpose(binary);
                bin=binary_t-'0';
                str = sprintf('%x', bin);
                arr=str-'0';
                 display([xc,yc])
                  imshow(img);
    set(handles.edit37,'String',str);
                hold on;
                plot(yc,xc,'o');
                hold off;
            selected=1;
        else
            w='Select a grayscale image';
            set(handles.edit2,'String',w);
        end
        if (any(namefile~=0) &&  length(size(img))>2)
            w='Select a grayscale image';
            set(handles.edit2,'String',w);

            selected=0;
        end
        set(handles.edit53,'String',namefile);

trellis = poly2trellis(5,[25 17])
data = arr
codedData = convenc(data,trellis)
vitdec(codedData,trellis,10,'trunc','hard')
str16 = sprintf('%x', codedData);
set(handles.edit38,'String',str16);
binary16 = dec2bin(codedData);
 binary_t19 = transpose(binary16);
 bin16 = binary_t19-'0';
 str16 = sprintf('%x', bin16);
algorithm = 'SHA256';
hasher =
System.Security.Cryptography.HashAlgorithm.Create('SHA256')
% GENERATING THE HASH:
str = str16
hash_byte = hasher.ComputeHash( uint8(str) );
```

```
% System.Byte class
hash_uint8 = uint8( hash_byte );
% Array of uint8
hash_hex = dec2hex(hash_uint8);
% Array of 2-char hex codes

% Generate the hex codes as 1 long series of characters
hashStr = str([]);
nBytes = length(hash_hex);
for k=1:nBytes
    hashStr(end+1:end+2) = hash_hex(k,:);
end
fprintf(1, '\n\tThe %s hash is: "%s" [%d bytes]\n\n',
algorithm, hashStr, nBytes);

binary29 = hexToBinaryVector(hash_hex)
 binary_t29=transpose(binary29);
str29 = sprintf('%x', binary_t29)
 arr29 = str29-'0';
 set(handles.edit39,'String',hashStr);
 set(handles.edit43,'String',str29);

x = randi([0 1],1,256);%key
xor25 = x
binary25 = dec2bin(xor25);
 binary_t25=transpose(binary25);
 bin25=binary_t25-'0';
 str25 = sprintf('%x', bin25)
 arr25 = str25-'0';
 set(handles.edit40,'String',str25)

xor26= bitxor(arr29,arr25);
binary26 = dec2bin(xor26);
 binary_t26=transpose(binary26);
 bin26=binary_t26-'0';
 str26 = sprintf('%x', bin26)
 arr26 = str26-'0'
 set(handles.edit41,'String',str26)

algorithm2 = 'SHA256';
hasher2 =
System.Security.Cryptography.HashAlgorithm.Create('SHA256')
% GENERATING THE HASH:
str2 = str25
hash_byte2 = hasher2.ComputeHash( uint8(str2) );
% System.Byte class
hash_uint82 = uint8( hash_byte2 );
% Array of uint8
hash_hex2 = dec2hex(hash_uint82);
% Array of 2-char hex codes

% Generate the hex codes as 1 long series of characters
hashStr2 = str2([]);
nBytes2 = length(hash_hex2);
for k=1:nBytes2
    hashStr2(end+1:end+2) = hash_hex2(k,:);
```

```
end
fprintf(1, '\n\tThe %s hash is: "%s" [%d bytes]\n\n',
algorithm2, hashStr2, nBytes2);

binary24 = hexToBinaryVector(hashStr2);
 binary_t24=transpose(binary24);
str42 = sprintf('%x', binary_t24);
set(handles.edit52,'String',hashStr2);
set(handles.edit42,'String',str42);


% --- Executes on button press in pushbutton19.
function pushbutton19_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton16 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)
 selected=0;


[namefile,pathname]=uigetfile({'*.bmp;*.tif;*.tiff;*.jpg;*.jpe
g;*.gif;*.pgm','IMAGE Files
(*.bmp,*.tif,*.tiff,*.jpg,*.jpeg,*.gif,*.pgm)'},'Chose
GrayScale Image');
        if namefile~=0
            [img,map]=imread(strcat(pathname,namefile));
            axes(handles.axes7);
            imshow(img);
            [oimg,fimg,bwimg,eimg,enhimg] =
fft_enhance_cubs(img);
                [xc,yc]=supercore7(enhimg);
                 binary = dec2bin([xc,yc],10);
                binary_t=transpose(binary);
                bin=binary_t-'0';
                str = sprintf('%x', bin);
  selected=1;
        else
            w='Select a grayscale image';
            set(handles.edit2,'String',w);
        end
        if (any(namefile~=0) &&  length(size(img))>2)
            w='Select a grayscale image';
            set(handles.edit2,'String',w);
            selected=0;
        end



% --- Executes on button press in pushbutton2.
function pushbutton28_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
```

```
% handles    structure with handles and user data (see
GUIDATA)
global pathname;
global namefile;
global img;
global selected;
global immagine n_bands h_bands n_arcs h_radius h_lato
n_sectors matrice matricer num_disk


n_bands=4;
h_bands=20;
n_arcs=16;
h_radius=12;
h_lato=h_radius+(n_bands*h_bands*2)+16;
if mod(h_lato,2)==0
    h_lato=h_lato-1;
end
n_sectors=n_bands*n_arcs;
num_disk=8;

matrice  = zeros(h_lato);
% sectorization matrix for the input image
matricer = zeros(h_lato);
% sectorization matrix for the rotated input image
for ii=1:(h_lato*h_lato)
    matrice(ii) = whichsector(ii,0);
    matricer(ii)= whichsector(ii,1);
end
 if (any(namefile~=0) &&  length(size(img))>2)
            w='Select a grayscale image';
            set(handles.edit2,'String',w);
            selected =0;
        end

        if selected ==1


            immagine=double(img);

            if isa(img,'uint8')
                graylevmax=2^8-1;
            end
            if isa(img,'uint16')
                graylevmax=2^16-1;
            end
            if isa(img,'uint32')
                graylevmax=2^32-1;
            end
            fingerprint = immagine;


            N=h_lato;

            [oimg,fimg,bwimg,eimg,enhimg] =
fft_enhance_cubs(fingerprint);
            fingerprint = enhimg;
```

```
            [YofCenter,XofCenter]=supercore7(fingerprint);

[CroppedPrint]=cropping(XofCenter,YofCenter,fingerprint);

[NormalizedPrint,vector]=sector_norm(CroppedPrint,0,0);

            for (angle=0:1:num_disk-1)
                gabor=gabor2d_sub(angle,num_disk,0);

ComponentPrint=conv2fft(NormalizedPrint,gabor,'same');
                [disk,vector]=sector_norm(ComponentPrint,1,0);
                finger_code1{angle+1}=vector(1:n_sectors);
            end



[NormalizedPrint,vector]=sector_norm(CroppedPrint,0,1);

            for (angle=0:1:num_disk-1)
                gabor=gabor2d_sub(angle,num_disk,1);

ComponentPrint=conv2fft(NormalizedPrint,gabor,'same');
                [disk,vector]=sector_norm(ComponentPrint,1,1);
                finger_code2{angle+1}=vector(1:n_sectors);
            end
            % FingerCode added to database
            if (exist('FingerPrint_database.dat')==2)
                load('FingerPrint_database.dat','-mat');
                fp_number=fp_number+1;
                data{fp_number,1}=finger_code1;
                data{fp_number,2}=finger_code2;
                namefile_vector{fp_number} = namefile;
                path_vector{fp_number}     = pathname;

save('FingerPrint_database.dat','data','fp_number','namefile_v
ector','path_vector','-append');
            else
                fp_number=1;
                data{fp_number,1}=finger_code1;
                data{fp_number,2}=finger_code2;
                namefile_vector{fp_number} = namefile;
                path_vector{fp_number}     = pathname;

save('FingerPrint_database.dat','data','fp_number','namefile_v
ector','path_vector');
            end

            message=strcat('Succesfully added to database.
Fingerprint no. ',namefile);
            msgbox(message,'FingerCode DataBase','help');
        end
load('FingerPrint_database.dat','-mat');

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)
global pathname;
global namefile;
global img;
global selected;
n_bands=4;
h_bands=20;
n_arcs=16;
h_radius=12;
h_lato=h_radius+(n_bands*h_bands*2)+16;
if mod(h_lato,2)==0
    h_lato=h_lato-1;
end
n_sectors=n_bands*n_arcs;
num_disk=8;

matrice  = zeros(h_lato);
% sectorization matrix for the input image
matricer = zeros(h_lato);
% sectorization matrix for the rotated input image
for ii=1:(h_lato*h_lato)
    matrice(ii) = whichsector(ii,0);
    matricer(ii)= whichsector(ii,1);
end
 if (exist('FingerPrint_database.dat')==2)
            load('FingerPrint_database.dat','-mat');

            if (any(namefile~=0) &&  length(size(img))>2
                w='Select a grayscale image';
            set(handles.edit2,'String',w);
                selected =0;
            end

            if selected==1

                message = strcat('Image selected for
fingerprint matching: ',namefile);

            set(handles.edit2,'String',message);

                message = strcat('Location: ',pathname);
                disp(message);

                immagine=double(img);

                if isa(img,'uint8')
                    graylevmax=2^8-1;
                end
                if isa(img,'uint16')
                    graylevmax=2^16-1;
                end
                if isa(img,'uint32')
                    graylevmax=2^32-1;
```

```
                end
                fingerprint = immagine;

                N=h_lato;

                [oimg,fimg,bwimg,eimg,enhimg] =
fft_enhance_cubs(fingerprint);
                fingerprint = enhimg;
                [list_of_core]= supercore7_list(fingerprint);
                results_img       =
zeros(size(list_of_core,1),1);
                results_dis       =
zeros(size(list_of_core,1),1);
                message = strcat('Candidates for core points:
',num2str(size(list_of_core,1)));

                for scan_core = 1:size(list_of_core,1)
                    message = strcat('Scanning candidate #
',num2str(scan_core));

                    YofCenter = list_of_core(scan_core,1);
                    XofCenter = list_of_core(scan_core,2);
%-------------------------------------------------------------
%-------------------------------------------------------------


[CroppedPrint]=cropping(XofCenter,YofCenter,fingerprint);

[NormalizedPrint,vector]=sector_norm(CroppedPrint,0,0);

% memory for input vector features
vettore_in=zeros(num_disk*n_sectors,1);
                    for (angle=0:1:num_disk-1)
                        gabor=gabor2d_sub(angle,num_disk,0);

ComponentPrint=conv2fft(NormalizedPrint,gabor,'same');

[disk,vector]=sector_norm(ComponentPrint,1,0);

finger_code{angle+1}=vector(1:n_sectors);

vettore_in(angle*n_sectors+1:(angle+1)*n_sectors)=finger_code{
angle+1};
                    end

                    vettore_a=zeros(num_disk*n_sectors,1);
                    vettore_b=zeros(num_disk*n_sectors,1);
                    best_matching=zeros(fp_number,1);
                    valori_rotazione=zeros(n_arcs,1);
% start checking ------------------------------------
                    for scanning=1:fp_number
                        fcode1=data{scanning,1};
                        fcode2=data{scanning,2};
                        for rotazione=0:(n_arcs-1)
                            p1=fcode1;
                            p2=fcode2;
```

```
% rotate the values inside disk

for disk_account=1:num_disk
                                  disk1=p1{disk_account};
                                  disk2=p2{disk_account};
                                  for old_pos=1:n_arcs

new_pos=mod(old_pos+rotazione,n_arcs);
                                       if new_pos==0
                                           new_pos=n_arcs;
                                       end
                                       for
account_bande=0:1:(n_bands-1)

disk1r(new_pos+band_account*n_arcs)=disk1(old_pos+band_account
*n_arcs);

disk2r(new_pos+bandeaccount*n_arcs)=disco2(old_pos+bandeaccoun
t*n_arcs);                                             end
                                  end
                                  p1{count_disk}=disk1r;
                                  p2{count_disk}=disk2r;
                              end
% ruoto i dischi circularly
                              for old_disk=1:num_disk

new_disk=mod(old_disk+rotation,num_disk);
                                  if new_disk==0
                                     new_disk=num_disk;
                                  end
                                  pos=old_disk-1;

vettore_a(pos*n_sectors+1:(pos+1)*n_sectors)=p1{new_disk};

vettore_b(pos*n_sectors+1:(pos+1)*n_sectors)=p2{new_disk};
                               end
                               d1=norm(vettore_a-vettore_in);
                               d2=norm(vettore_b-vettore_in);
                               if d1<d2
                                   minimum_val=d1;
                               else
                                   minimum_val=d2;
                               end
rotation_value(rotation+1)=minimum_val;
                           end

[minimum,minimum_position]=min(rotation_value);
                           best_matching(scanning)=minimum;
                      end

[minimum_distance,minimum_position]=min(best_matching);
                      results_img(scan_core) =
position_minimum;
                      results_dis(scan_core) =
minimum_distance;
                 end
```

```
%-----------------------------------------------------
                [minimum_distance,minimum_position] =
min(results_dis);
                dito_minimo = results_img(posizione_minimo);

                message=strcat('Recognized
fingerprint:',namefile_vector{minimum_dito});

message=strcat('Location:',path_vector{minimum_dito});

                message=strcat('The Nearest Fingerprint Is :
',num2str(minimum_dito),...
                    ' With a distance Of :
',num2str(minimum_distance));
                set(handles.edit5,'String',minimum_distance);
                 set(handles.edit4,'String',minimum_dito);
                 name=namefile_vector{minimum_dito};
                 pathx=path_vector{minimum_dito};
                 resultx=strcat(pathx,name);
                 imgx=imread(resultx);

                  axes(handles.axes6);
            imshow(imgx);
                msgbox(message,'DataBaseInfo','help');
            end
        else
            message='DataBase is empty. No check is
possible.';
            msgbox(message,'FingerCode DataBase
Error','warn');
        end

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)
if (exist('FingerPrint_database.dat')==2)
            button = questdlg('Do you want to Delete
Database?');
            if strcmp(button,'Yes')
                delete('FingerPrint_database.dat');
                msgbox('Database was succesfully
Removed.','Database removed','help');
            end
        else
            warndlg('Database is empty.',' Warning ')
        end
```

```
% --- Executes on button press in pushbutton27.
function pushbutton27_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton27 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see
GUIDATA)

global pathname;
global namefile;
global img;
global selected;
global xor25;
global str26;
global arr26;
global arr29;
global hashStr;
global hashStr2;
global str24;
global str42;

clc;
        selected=0;


[namefile,pathname]=uigetfile({'*.bmp;*.tif;*.tiff;*.jpg;*.jpe
g;*.gif;*.pgm','IMAGE Files
(*.bmp,*.tif,*.tiff,*.jpg,*.jpeg,*.gif,*.pgm)'},'Chose
GrayScale Image');
        if namefile~=0
            [img,map]=imread(strcat(pathname,namefile));
           axes(handles.axes8);
           imshow(img);
           [oimg,fimg,bwimg,eimg,enhimg] =
fft_enhance_cubs(img);
            axes(handles.axes9);
           imshow(enhimg);
           [gimg,oimg] = orientation_image_rao(img);
            axes(handles.axes10);
            view_orientation_image(oimg);
             [oimg,fimg,bwimg,eimg,enhimg] =
fft_enhance_cubs(img);
                [xc,yc]=supercore7(enhimg);

                 axes(handles.axes11);
                 binary31 = dec2bin([xc,yc],10);
                binary_t31=transpose(binary31);
                bin31=binary_t31-'0';
                str31 = sprintf('%x', bin31);
                arr31=str31-'0';
                 display([xc,yc])
                  imshow(img);
    set(handles.edit44,'String',str31);
                hold on;
```

```
                plot(yc,xc,'o');
                hold off;
            selected=1;
        else
            w='Select a grayscale image';

        end
        if (any(namefile~=0) &&  length(size(img))>2)
            w='Select a grayscale image';
            set(handles.edit8,'String',w);

            selected=0;
        end
        set(handles.edit54,'String',namefile);

        %decode & encode
 trellis = poly2trellis(5,[25 17])
data = arr31
codedData = convenc(data,trellis)
vitdec(codedData,trellis,10,'trunc','hard')
str16 = sprintf('%x', codedData);

set(handles.edit45,'String',str16);
binary16 = dec2bin(codedData);
 binary_t19 = transpose(binary16);
 bin16 = binary_t19-'0';
 str16 = sprintf('%x', bin16);

 algorithm = 'SHA256';
hasher =
System.Security.Cryptography.HashAlgorithm.Create('SHA256')
% GENERATING THE HASH:
str = str16
hash_byte = hasher.ComputeHash( uint8(str) );  % System.Byte
class
hash_uint8 = uint8( hash_byte );               % Array of
uint8
hash_hex = dec2hex(hash_uint8);                % Array of 2-
char hex codes

% Generate the hex codes as 1 long series of characters
hashStr = str([]);
nBytes = length(hash_hex);
for k=1:nBytes
    hashStr(end+1:end+2) = hash_hex(k,:);
end
fprintf(1, '\n\tThe %s hash is: "%s" [%d bytes]\n\n',
algorithm, hashStr, nBytes);

binary29 = hexToBinaryVector(hash_hex)
 binary_t29=transpose(binary29);
 %bin29 = binary_t29-'0'
str29 = sprintf('%x', binary_t29)
 arr29 = str29-'0';
 set(handles.edit46,'String',hashStr);
 set(handles.edit47,'String',str29);
```

```
 set(handles.edit48,'String',str26)

xor27= bitxor(arr29,arr26);
binary27 = dec2bin(xor27);
 binary_t27=transpose(binary27);
 bin27=binary_t27-'0';
 str27 = sprintf('%x', bin27);
 arr27 = str27-'0'
set(handles.edit49,'String',str27)

 x = randi([0 1],1,448);%key
xor25 = x
binary25 = dec2bin(xor25);
 binary_t25=transpose(binary25);
 bin25=binary_t25-'0';
 str25 = sprintf('%x', bin25)
 arr25 = str25-'0';

algorithm2 = 'SHA256';
hasher2 =
System.Security.Cryptography.HashAlgorithm.Create('SHA256')
% GENERATING THE HASH:
str2 = str27;
hash_byte2 = hasher2.ComputeHash( uint8(str2) ); % System.Byte
class
hash_uint82 = uint8( hash_byte2 );              % Array of
uint8
hash_hex2 = dec2hex(hash_uint82);               % Array of 2-
char hex codes

% Generate the hex codes as 1 long series of characters
hashStr2 = str2([]);
nBytes2 = length(hash_hex2);
for k=1:nBytes2
    hashStr2(end+1:end+2) = hash_hex2(k,:);
end
fprintf(1, '\n\tThe %s hash is: "%s" [%d bytes]\n\n',
algorithm2, hashStr2, nBytes2);

binary24 = hexToBinaryVector(hashStr2);
 binary_t24=transpose(binary24);
 %bin24=binary_t24-'0'
str24 = sprintf('%x', binary_t24);
% arr24=str24-'0'
 set(handles.edit51,'String',hashStr2);
set(handles.edit50,'String',str24);

if str42==str24
               msgbox('The Fingerprint is Valid','help');

else
           warndlg('The Fingerprint is not Valid!',' Warning
');
end
```