N° d'ordre: 92/2024-D/INF

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

Ministry Of Higher Education And Scientific Research

University Of Sciences And Technology Of Houari Boumediene

Faculty of Computer Science

DOCTORAL DISSERTATION IN SCIENCE

Submitted in partial fulfillment of the requirements for **Doctorate degree**

In: Computer Science

Option: Informatics Systems

Presented by: Maamar Hamadouche

Theme:

Securing Biometric Systems by using Perceptual Hashing

Publicly defended on 04/07/2024 before the jury composed of:

Mr.	BELKHIR Abdelkader	Professor	at USTHB	Chairperson
Mr.	ZAFOUNE Youcef	Associate Professor A	at USTHB	Thesis Supervisor
Mr.	ZEBBICHE Khalil	Associate Professor A	at ESGN	Co-supervisor
Mr.	LOUKHAOUKHA Khaled	Senior Researcher A	at CRD-GN	Examiner
Mr.	DJOUMAA Amir	Associate Professor A	at ENSIA	Examiner
Mr.	BENKAOUHA Haroun	Associate Professor A	at USTHB	Examiner
Mr.	GUERROUMI Mohamed	Professor	at USTHB	Guest

Acknowledgements

Praise be to the Almighty God who has given me faith, courage, and patience to carry out this work.

I want to express my deep gratitude to my supervisor Dr. Zafoune Youcef for his orientations, availability, listening, and patience during the realization of this job.

I thank Dr. Zebbiche Khalil for having co-supervised me, for the confidence he has placed in me, through his presence, his directions, modesty, advice, and constructive remarks for the good progress of this work.

I would like to thank Pr. Mohamed Guerroumi who helps me to improve this work and gave me remarks that helped me to perfect this manuscript.

I would like to express my sincere gratitude to the chairperson Pr. BELKHIR Abdelkader and the members of my thesis jury Dr. LOUKHAOUKHA Khaled, Dr. AMIR Djouamaa, and Dr. BENKAOUHA Haroun for their valuable time, expertise, and contribution to the evaluation of my work. Their insightful comments, suggestions, and constructive feedback have greatly enriched the quality of my research.

I express my deep gratitude to my wife and my children for their encouragement and prayers that allowed me to achieve this modest work. I am very grateful for the confidence they have placed in me.

Finally, I express my gratitude to all those who have contributed in one way or another to the achievement of this work.

O Allah, send your blessings on your noble messenger, his family, and companions, and bless us in our life.

Dedication

I would like to dedicate this work in loving memory of my parents, who have passed away :

May they find within it a testament to my deep gratitude and appreciation for everything they have done for me.

To my beloved wife Dr. Tebbi Hanane and my children Raouf, Iyad et Serine, whose unwavering love and support have been my guiding light throughout this journey, who have always provided love and liveliness, I express my heartfelt dedication.

To my siblings, grandparents, and extended family, and to all those who have offered their assistance, directly or indirectly, and who have shared in the emotional moments of completing this work.

To my superiors and all my mates who have always encouraged me, and to whom I wish more success.

Thanks!

Maamar Hamadouche

ملخص

تركز رسالةْ الدكتوراه بعنوانْ أمان أنظمة البيومترية باستخدام تقنيات الهاش المعرفية على تعزيز أمان أنظمة البيومترية من خلال استخدام تقنيات الهاش المعرفية. أصبحت أنظمة البيومترية متواجدة بشكل متزايد في تطبيقات مختلفة، بدءًا من التحكم في الوصول إلى التحقق من الهوية. ومع ذلك، فإن هذه الأنظمة عرضة للهجمات، مثل هجمات العرض وهجمات إعادة البث، والتي يمكن أن تؤثر على سلامة ودقة النظام. لمعالجة هذه التحديات، تقترح الرسالة دمج تقنيات الهاش المعرفية، والتي تمكن من المصادقة القوية والفعالة للبيانات البيومترية. تستكشف البحث تقنيات الهاشٍ المعرفية المختلفة، وتقوم بتقييم أدائها، وتطوير نهج جديد مصمم خصيصاً لأنظمة البيومترية. تهدف التقنيات المقترحة إلى تعزيز أمان وموثوقية أنظمة البيومترية، وتوفير حماية قوية ضد مختلف الهجمات مع الحفاظ على الاستخدام السهل والكفاءة.

كلمات مفتاحية: الأمان، أنظمة البيومترية، الهاش المعرفية، المصادقة، الهجمات، القوة، الاستخدام، الكفاءة.

Abstract

This PhD thesis titled "The Security of Biometric Systems Using Perceptual Hashing" focuses on enhancing the security of biometric systems by using perceptual hashing techniques. Biometric systems have become increasingly prevalent in various applications, ranging from access control to identity verification. However, these systems are susceptible to attacks, such as presentation attacks and replay attacks, which can compromise their integrity and accuracy. To address these challenges, the thesis proposes the integration of perceptual hashing methods, which enable robust and efficient authentication of biometric data. The research investigates different perceptual hashing algorithms, evaluates their performance, and develops novel approaches designed specifically for biometric systems. The proposed techniques aim to enhance the security and reliability of biometric systems, providing robust protection against various attacks while maintaining usability and efficiency.

Key words: *Security, Biometric Systems, Perceptual Hashing, Authentication, Attacks, Robustness, Usability, Efficiency.*

Résumé

Cette thèse de doctorat intitulée "La sécurité des systèmes biométriques avec l'utilisation du hachage perceptuel" se concentre sur le renforcement de la sécurité des systèmes biométriques grâce à l'utilisation de techniques de hachage perceptuel. Les systèmes biométriques sont de plus en plus répandus dans diverses applications, allant du contrôle d'accès à la vérification d'identité. Cependant, ces systèmes sont vulnérables aux attaques telles que les attaques de présentation et les attaques de rejeu, qui compromettent leur intégrité et leur précision. Pour relever ces défis, la thèse propose l'intégration de méthodes de hachage perceptuel, qui permettent une authentification robuste et efficace des données biométriques. La recherche explore différentes algorithmes de hachage perceptuel, évalue leurs performances et développe de nouvelles approches spécifiquement adaptées aux systèmes biométriques. Les techniques proposées visent à renforcer la sécurité et la fiabilité des systèmes biométriques, en fournissant une protection solide contre diverses attaques tout en maintenant l'utilisabilité et l'efficacité.

Mots clés : *Sécurité, Systèmes biométriques, Hachage perceptuel, Authentification, Attaques, Robustesse, Utilisabilité, Efficacité.*

Contents

List of Figures

List of Tables

List of Abbreviations

1 General introduction **1**

General introduction **1**

1.1 Introduction 1
1.2 Motivation and challenges 4
1.3 Main contributions 5
1.4 Related works 6
1.5 Dissertation outline 8

2 Introducing Biometric Systems Attacks **11**

2.1 Introduction 11
2.2 General Concepts 11
2.3 Applications of Biometric Systems 12
2.4 Architecture of a biometric system 13
2.5 General scheme of a biometric system 14
2.5.1 **Enrollment**: 15
2.5.2 **Authentication**: 15
2.5.3 Identification: 16
2.6 Evaluation of a biometric system 16
2.6.1 Usage Evaluation 17
2.6.2 Performance evaluation 17
2.6.3 Security evaluation 17
2.7 Security concerns of Biometric Systems 18
2.7.1 Vulnerabilities and attacks models 18
2.7.2 Threat Models 19
2.7.3 Suggested Solutions 33

2.8 Conclusion 34

3 An Overview of Perceptual Hashing for Image Fingerprinting 36
3.1 Introduction 36
3.2 Cryptographic Vs Perceptual hashing Functions 36
3.2.1 Definitions and properties of perceptual hashing 38
3.2.2 Properties of Perceptual Hashing (PH) 39
3.2.3 General framework of a perceptual image hashing system . 40
3.2.4 Classification of perceptual image hashing techniques . . . 43
3.3 Applications of Perceptual Hashing 48
3.3.1 Content Identification: 48
3.3.2 Integrity check 50
3.3.3 Watermarking Support 51
3.3.4 Content-Based Media Retrieval and Processing 54
3.3.5 Similarity Metrics for Perceptual Hashes 55
3.4 Conclusion 58

4 A Comparative Evaluation of Perceptual Image Hashing Approaches 59
4.1 Introduction 59
4.2 The compared PIH Techniques 60
4.2.1 A-Hash 60
4.2.2 D-Hash 61
4.2.3 P-Hash 62
4.2.4 W-Hash 63
4.2.5 SVD-Hash 64
4.3 Experimental Results 65
4.3.1 Perceptual Robustness 65
4.3.2 Discrimination capability 66
4.3.3 Authentication results 69
4.4 Conclusion 71

5 A Replay Attack Detection Scheme based on Perceptual Image Hashing 73
5.1 Introduction 73
5.2 The Proposed solution 74
5.2.1 The proposed RAD scheme 75
5.2.2 Proposed perceptual hashing system 78
5.2.3 Hash generation 78
5.2.4 Content authentication 83
5.2.5 Content identification 84
5.3 Experimental results and discussions 86
5.3.1 Generated Hash analysis 87

5.3.2 Authentication Performance 89
5.3.3 Identification Performance 95
5.4 Conclusion 101

6 General Conclusion and Perspectives **102**

Bibliography **105**

List of Figures

2.1 Generic Architecture of a biometric system [1] 14
2.2 Joshi's et al. Threat Model : A number corresponding to an arrow or a dashed line represents an attack point (AP) [2]. 20
2.3 Ratha et al. [3] Threat Model : The arrows and numbers indicate the access point of information for mounting a specific type of attacks . 26
2.4 Adler's ten Vulnerabilities Model [4]. 27
2.5 The Fiche-bone threat model proposed by Jain et al. [5] 30
2.6 The Bartlow and Cukic threat Model [6] 32

3.1 Keyed Hashing Function 38
3.2 Unkeyed Hashing Function 39
3.3 General Framework of a PIH system [7, 8] 41
3.4 The usage of PH in Content Identification. 49
3.5 The General Principle of Integrity verification using PH 50
3.6 Creation of Digital Signature process 51
3.7 The Digital Signature checking process 52
3.8 Watermark support: Perceptual hash watermark Embedding 53
3.9 Watermark support: Perceptual hash watermark extraction 54

4.1 Block diagram of the A-Hash algorithm 61
4.2 Block diagram of the D-Hash algorithm 62
4.3 Block diagram of the P-Hash algorithm 63
4.4 Block diagram of the W-Hash algorithm 64
4.5 Block diagram of the SVD-Hash algorithm 65
4.6 Distribution of Normalized Hamming distances corresponding to the five tested techniques. 68

5.1 Illustrative diagram of a basic DRA. 74
5.2 Block diagram of the proposed scheme. 75
5.3 Proposed perceptual hashing system. 79
5.4 Examples of images used in our tests. 87
5.5 Histogram of the generated hash values. Block size: 32×32. 88
5.6 Authentication distances histograms for different block sizes. 90

5.7 Quantile-Quantile plots of the authentication distances for different block sizes. 91
5.8 Identification distances histograms for different block sizes. 95
5.9 Quantile-Quantile plots of the identification distances for different block sizes. 96
5.10 Identification distance histograms of the compared schemes. 98
5.11 Robustness under different kinds of attacks. 99
5.12 Performance comparison using ROC curves. 100

List of Tables

3.1 Examples of calculating the Hamming distance (HB) of two given strings X and Y having different systems (binary, Latin alphabet and decimal system). . . . 56
3.2 Examples of calculating the Normalized Hamming distance (NHD) of two given binary hashes H_1 and H_2. . . . 57
3.3 Examples of calculating the Bit Error Rate (BER) of two given binary hashes H_1 and H_2. . . . 57

4.1 Content-preserving manipulations and parameters setting 66
4.2 Hamming distances under different content-preserving manipulations. Best values are in bold. . . . 66
4.3 Collision Probability of the evaluated algorithms with different thresholds T 69
4.4 Quantitative comparisons using the Recall, Precision and F1-measure metrics. Best results are in bold. . . . 70

5.1 Hash length comparison 89
5.2 False Positive Rate (FPR) of the authentication process for different block sizes. . . . 92
5.3 True Positive Rate (TPR) of the authentication process under rotation distortion with different angle values. . . . 93
5.4 True Positive Rate (TPR) of the authentication process under image shifting distortion with different shift values. . . . 93
5.5 True Positive Rate (TPR) of the authentication process under image manipulations. . . . 94
5.6 False Positive Rate (FPR) in the identification process in the absence of distortions. . . . 98
5.7 AUC performance comparison 101

List of Abbreviations

ASV: Automatic Speaker Verification
AWGN: Additive White Gaussian Noise
ALCH: Angular Location Context Hashing
CBIR: Content-Based Image Retrieval
CCA: Canonical Correlation Analysis
CC: Common Criteria
CNN: Convolutional Neural Network
CQCC: Constant-Q Cepstral Coefficients
CPOs: Content-Preserving Operations
CCOs: Content-Changing Operations
CS: Compressed Sensing
DCT: Discrete Cosine Transform
DFT: Discrete Fourier Transform
DLFS: Decision-Level Feature Switching
DWT: Discrete Wavelet Transform
DST: Discrete Sine Transform
DBLM: Differential Block Luminance Mean
DRA: Digital Replay Attacks
ECC: Error-Correcting Coding
FJLT: Fast Johnson-Lindenstrauss Transform
GLCM: Gray Level Co-occurrence Matrix
HD: Hamming distance
HVS: Human Visual System
IA-IF: Iterative Adaptive Inverse Filtering
GMM: Gaussian Mixture Model
LFS: Linear Filterbank Slope
LLE: Locally Linear Embedding
MFCC: Mel Frequency Cepstral Coefficients
MFS: Mel Filterbank Slope
MiTM: Man-in-The-Midle
NMF: Nonnegative Matrix Factorization
NHD: Normalized Hamming distance
MV-SHPS: Multi-View Semi-supervised Hashing algorithm with Perceptual Saliency

PIH: Perceptual Image Hashing
PH: Perceptual Hashing
PCA: Principal Component Analysis
PR: Pseudo-Random
RAD: Replay Attack Detection
RLCH: Round Location-Context Hashing
SCA: Side Channel Attack
SIFT: Scale-Invariant Feature Transform
STACOG: Space-Time Auto-Correlation Of Gradients
SVD: Singular Value Decomposition
UI: User Interface
VPAHS: Virtual Prior Attacked Hash Space

Chapter 1

General introduction

1.1 Introduction

The rise of intelligent sensing technologies is transforming the way we interact with machines. In many cases, these technologies are mediating our interactions, as in the case of smartphone unlocking using face or fingerprint recognition.

In the face of growing crime, such as illegal immigration, document fraud, and identity theft, and emerging threats such as cybercrime, new technological solutions are being gradually implemented. One of these technologies is biometrics, which has quickly distinguished itself as the most relevant way to identify and authenticate individuals reliably and quickly by exploiting their unique physical or behavioral characteristics.

In today's world, fraud and insecurity pose increasing concerns, particularly in sectors like e-commerce and banking. Traditional authentication methods, such as knowledge-based approaches (e.g., passwords and PINs) and token-based approaches (e.g., badges, identification papers, keys), have proven to be insecure. Passwords can be forgotten or easily guessed, while badges and other forms of identification can be lost or stolen. Biometric characteristics offer an alternative to these conventional methods. Leveraging the universality, uniqueness, measurability, and permanence of biometric features brings various benefits for authentication purposes. Highly populated countries like India and China are at the forefront of this trend, building massive national identity networks that rely entirely on biometric characteristics. These networks raise a number of important social and ethical questions, such as how to protect individual privacy and prevent discrimination [5, 9, 10]. Biometric systems can fulfill two distinct functions, identification and authentication. The identification answers the question "who are you?". In this case, the person is identified among others (1 to N check). Their personal data is compared with the data of other people contained in the same database or in any linked databases. Authentication answers the question, "Are you who you claim you are?" In this case, biometrics make it possible to certify the identity of a person by comparing the data that he/she

is presenting with the pre-recorded data of the person he/she claims to be (1 to 1 verification). Biometric-based authentication systems are more and more present in our daily life as an alternative solution to traditional authentication systems, and are getting high accuracy rates since they are based on something that we are (e.g., fingerprint, face, iris, etc.). Nevertheless, with advances in technology, biometric data (e.g., fingerprint, face, iris) risk to be captured/sniffed, duplicated, modified, and then resubmitted in the same channel or in other applications that utilize the same biometric features, to gain unauthorized access to the system.

This thesis provides a comprehensive overview of the latest advancements in biometric security and privacy, highlighting current trends and future challenges. It addresses the hurdles associated with designing secure and privacy-preserving biometric authentication protocols. Additionally, it explores the primary threats to biometric system security and provides guidance on developing secure and privacy-preserving biometric systems. To enhance the usability of biometric authentication systems, the thesis categorizes and evaluates the existing state of biometric systems with a specific emphasis on privacy and security concerns. It presents an overview of biometric systems and their privacy protection mechanisms. Furthermore, it delves into personal privacy considerations within the context of adopting biometric identification systems.

Saying that biometric data are unique personal identifiers that cannot be forged, is fundamentally wrong and can raise a false sense of security. Recently, a research reveals that the human biometric characteristics can be forged (falsified) by malicious persons, and copying digitized biometric data may be even easier than copying physical biometrics [11]. The worst thing of all of this is that, on the one hand, once compromised, biometric data is compromised forever; peoples cannot change their stolen own characteristics the way they do in case of hacked passwords, and on the other hand, biometric data may be used for numerous applications at the same time. The victim will therefore potentially be affected for the rest of his or her life. In the same context, and as any other authentication system, biometric systems are subject to various attacks that can be launched by attackers for fraudulent and illegal purposes [12, 13]. The success of these attacks is greater in the case of open, remote and uncontrolled systems.

To address these security issues, several hardware-based and software-based solutions have been proposed and implemented. Among the software solutions, we note the techniques for protecting and authenticating digital biometric data. These techniques are inspired by those used in the fields of cryptography and digital watermarking. Recently, some techniques based on perceptual hashing, also known as robust hashing, have been proposed to authenticate multimedia biometric data (image, sound, video). This is the perspective in which our work fits. It aims to propose a solution using perceptual hashing techniques to address one of the most devas-

tating attacks, known as digital replay attacks, which target fingerprint identification systems. This attack represent type 2 of the attacks model described by Ratha *et al.* [14]. These attacks occur when the attacker replaces the digital biometric signal, usually captured by the remote sensor, with a fraudulent digital signal previously eavesdropped from the target victim. Indeed, with the widespread proliferation of network-based biometric authentication systems, transmitted biometric data can now be intercepted/sniffed, duplicated, modified, and then later resubmitted in the same channel or in other systems that use the same biometric data, in order to gain unauthorized access in most cases. This kind of attack is commonly used, easy to implement, and does not require the attacker to have any skills or knowledge in signal processing to perform it [15, 16]. Furthermore, these attacks may be automated, especially by using malicious software such as Trojan Horses or viruses to steal a copy of digital biometric data directly from the victim's own computer. This automation could affect many target victims at the same time [17].

Although digital replay attacks are considered by many researchers a serious security concern, most existing solutions focused on spoofing or presentation attacks and proposed solutions to discriminate between fake and live traits, rather than tackling the real problem of digital replay attacks. Indeed, in the literature, only few published works have addressed and proposed practical countermeasures [17]. These countermeasures are mainly based on the Replay Attack Detection (RAD) principle, which aims to detect the difference between the original and the replayed signals [18]. The major part of these works confirm that the detection of replay attacks is a very challenging task to achieve and is still a problem far from being solved, to obtain effective countermeasures [19]. With respect to the practical implementation, some countermeasures operate in controlled environments [20] or need the cooperation of end users [21–24]. Other solutions suggest integrating intelligent hardware modules into sensors for complex encryption or Challenge/Response operations. Also, software-based modules are proposed to be introduced into the sensors that implement data encryption [25, 26] or digital watermarking algorithms [27–29].

Over the last decade, the field of perceptual image hashing (also known as robust image hashing) has witnessed a skyrocketing growth. It consists of converting the human visual perception of an image into a short character representation, which does not change with the specific data representation of the image, and requires a small storage space [30]. Its main objective is to produce a robust hash that resists against intentional manipulations or distortions caused by content preservation operations or geometric distortions, on the one hand, and to be sensitive to malicious tampering to image content and have a certain degree of security on the other hand. Due to its advantageous features, perceptual image hashing has become an effective solution for image retrieval and content authentication [31, 32].

In the literature, there are few published works that present adapted perceptual hashing techniques to protect biometric data. To the best of our knowledge, for the par-

ticular case of fingerprints, only the works of [33–36] were conducted in this context. Muthu and Rani [33] proposed a method for fingerprint identification based on perceptual minutiae image hashing. The proposed scheme combines SIFT-Harris feature points with the minutiae of the fingerprint images to incorporate the orientation and descriptor found in the minutiae of the fingerprint image. Muthu *et al.* [34] proposed a solution to improve the minutiae extraction of fingerprint images using perceptual hashing. The proposed hashing technique utilizes the Singular Values Decomposition of the approximation Wavelet Transform coefficients (DWT-SVD) to extract the hash values using a quantization step, and stores them into a database. This database is used, when a given image is distorted by noise, to recover the stored hash to be used instead of the distorted image. Abdullahi *et al.* [35] proposed a fingerprint image hashing scheme based on the shape context and minutiae. Their proposed method is based on texture pattern of the distributed minutiae points used in the shape context to generate a compact hash signature. The shape context is used to provide the description of the geometric structure of the fingerprints shape. Birouk *et al.* [36] developed a technique based on four functions namely SIFT (Scale-Invariant Feature Transform), Harris, DWT (Discrete Wavelet Transform) and SVD to produce a robust perceptual hashing scheme to be used in the context of fingerprint-based biometric template protection. First, the SIFT algorithm is applied on the extracted minutiae to produce robust features that resist against geometric attacks. Then the Harris technique is used to preserve only the stable key-points of the resulting hash vector. Next, the fingerprint template is formed through image binarisation and decomposed into blocks. Finally, SVD is applied on the resulting blocks to compute the approximation coefficients of each image block and therefore obtain the final hash signature.

In our thesis, we investigate the use of perceptual image hashing to protect biometric systems against digital replay attacks by authenticating biometric images and detecting possible fraudulent resubmission of these images. Indeed, our research represents a proposal for a suitable and viable RAD scheme that relies on the Challenge/Response principle and combines perceptual hashing and data hiding techniques. This scheme uses a two-step verification process at the receiver side to detect possible digital attacks by performing image authentication and identification operations. Furthermore, a new adapted perceptual image hashing technique is designed to perform both image authentication and identification operations instead of using two different techniques, which is more practical in real scenarios.

1.2 Motivation and challenges

Our motivation for carrying out this research is, on the one hand, the devastating effects of digital replay attack on biometric systems, and on the other hand, the lack of published works that address this issue and propose practical solutions to mitigate

this kind of attack, especially for biometric images such as fingerprints. Furthermore, through this work, it is intended to exploit the advantageous features of perceptual image hashing techniques related to image authentication and identification, particularly its ability to detect slightly altered images (i.e., detect similarity between an image and its altered copy) in order to design a practical and yet efficient countermeasure for biometric systems against Digital Replay Attacks (DRA).
The first challenge faced in this work is to design a solution that satisfies the conventional requirements of any DRA countermeasure. In fact, an efficient countermeasure should be able to: first, validate the integrity of the received biometric image (i.e. confirm that the received image is the one captured and sent by the sensor and has not been forged during transmission), and second, be able to detect if a biometric image has been received before, even if the image has encountered slight alterations by applying some commonly known image manipulation operations (i.e. to guarantee that the image has not been replayed in the biometric system). Adding to this challenge, the proposed solution should operate in an uncontrolled environment and does not need the cooperation of end users.
To handle this challenge, a novel two-step verification scheme that relies on the Challenge/Response principle is proposed. This scheme comprises two processes: The first, which is implemented on the sensor side, uses a perceptual image technique to compute a perceptual hash and send it along with the acquired image. The second process, which is performed on the receiver side, is a two-step verification process that uses the same hashing technique to perform the operations of verifying the authenticity of the received image and identifying if it has been submitted before to prevent replay attacks.
The second challenge was to develop or adapt a perceptual hashing technique that is capable of distinguishing between biometric images that visually look the same, especially for a non-expert person, and to detect a tampered version of the image. Indeed, in the case of fingerprint images, they are perceived by the majority of people as an alternation of curved dark lines, representing the ridges, along with white lines, which represent the valleys, and they almost have the same shape. Therefore, the point is to develop a technique that can detect dissimilarities between different biometric images, and in the meantime detect similarities between images and their altered versions. In addition, it is important that, for optimization purposes, the hashing technique must be capable of performing both image authentication and identification operations. To address this point, we developed a new perceptual hashing technique for fingerprint images that, can achieve both authentication and identification operations using the same hash.

1.3 Main contributions

The main contributions of this work are highlighted as follows:

- Design of a novel RAD scheme for biometric systems that takes advantage of the image perceptual hashing approach. To the best of our knowledge, there have been no published countermeasures in the literature that are based on perceptual hashing. The proposed scheme, which relies on the Challenge/Response principle, is a two-stage verification mechanism : the first one verifies that the received biometric image is actually the one sent by the sensor; this is accomplished by calculating the perceptual hash value of the received image using the same technique and the same parameters used by the sensor to calculate the hash value, and send it together with the biometric image. The second one checks that the received image has not been submitted before by seeking the received perceptual hash value in a separate database that contains all the perceptual hash values of the previously logged authentic users.
- Development of a new perceptual image hashing system for fingerprint images, that performs both image identification and authentication at the same time and using the same perceptual hash sequence. The proposed technique builds upon the works of [37] and [38], which are both robust video hashing systems for content identification and authentication, with the introduction of some adaptations to make it suitable for fingerprint images. The experimental results show that the proposed system provides higher performance in terms of authentication and identification.

1.4 Related works

In the literature, a number of works have been proposed as countermeasures to thwart against DRA. Ratha *et al.* [14] proposed a solution based on the Challenge/Response principle, where a transaction server generates a pseudo-random challenge and sends it to the biometric sensor. Upon receiving the challenge, the acquired sample and the corresponding response are sent to the transaction server, where the response is verified against the received sample for consistency. An inconsistency reveals the possibility of a replay attack. Czajka *et al.* [25] presented a replay attack prevention method for iris-based biometric system, that relies on randomized iris sectors. It takes advantage of appropriate data alteration within the image space, resulting in significant changes of the resulting code in the iris feature space. In the reference implementation, the authors have used the original Zak-Gabor-based iris coding, and the minimal dependency between the iris image alteration procedure and the resulting iris template in Zak-Gabor coding enables transmitting the iris image alteration code and the resulting iris template. In [39], Boutellaa *et al.* proposed an audiovisual synchrony assessment for the detection of replay attacks in a biometric system based on a talking face. The proposed assessment is based on the use of space-time auto-correlation of gradients (STACOG) for measuring the audiovisual synchrony. The STACOG description was then mixed with The Mel-Frequency Cepstral Coef-

ficients (MFCC) features using Canonical Correlation Analysis (CCA). To evaluate the effectiveness of their proposed approach, authors used realistic and challenging synthetic replay attacks that have been generated by publicly increasing the BANCA and XM2VTS databases. Gui *et al.* [40] addressed the security of a biometric-based brain print system against replay attack, and proposed a replay detection approach that takes advantage of noise residual features to detect if the input template is adversely modified and generated by adding noises to the legitimate brain print template. Mainly, the proposed approach consists of two separate steps: the identity recognition stage, which uses the Convolutional Neural Network (CNN) to classify the input brainwaves and verify the user's identity, and the replay detection stage, which uses the ensemble classifier to detect if the brainwave signals have been compromised and manipulated by using noise residual features. In [26], Loukhaoukha *et al.* proposed an image encryption algorithm to protect fingerprint-based systems. The proposed algorithm, which operates in the one-level Integer-to-Integer Lifting Wavelet Transform, consists of two operations: permutation and diffusion. Permutation operations are applied after partitioning the approximation and detail subbands into blocks and then permuting them using a permutation key. The diffusion operation is based on the Rubik's cube principle and is applied to each sub-band. In [41], Mel Filterbank Slope (MFS) and Linear Filterbank Slope (LFS) along with MFCC and Constant-Q Cepstral Coefficients (CQCC) were used in a Decision-Level Feature Switching (DLFS) paradigm as the features that best identify a replayed instance in a given environment. The MFS features use the Mel scale which emphasizes low frequency regions, while LFS uses the linear frequency scale where the resolution at high frequencies is better than that of MFS. Also, other features, namely: MFCC and CQCC, were used in this work for the detection of other spectral artifacts. The effectiveness of MFS and LFS features was evaluated and used in this work, according to the authors, for the first time for the detection of replay attacks in a voice-based biometric system. Avila *et al.* [42] presented a solution based on a blind estimate of the magnitude of channel responses to countermeasure replay attack in an Automatic Speaker Verification (ASV) system. To that end, they predict the log-spectrum average of the clean speech signal from a Gaussian mixture model (GMM) of RASTA filtered MFCCs trained using clean speech. They obtained the magnitude response of the channel by subtracting the log-spectrum of the observed signal from the predicted log-spectrum average of the proper signal. To evaluate their proposed technique, they used two datasets, namely the TIMIT dataset for the training task, and a second dataset containing replay attacks used during the second Automatic Speaker Verification Spoofing and Countermeasures Challenge (ASVspoof 2017). Wang *et al.* [43] used the replay clues of a whole utterance, by separately extracting different features from voiced and non-voiced segments and training separate Gaussian Mixed Models. To this end, they adopted a voice activity detector to get accurate boundaries of the different segments, then they extracted Constant-Q Cep-

stral Coefficients and Inverse Mel Frequency Cepstral Coefficients from voiced and non-voiced segments respectively. At the end, a Score Calibrator Toolkit is used to fuse the scores of voiced and non-voiced segments. Karmakar *et al.* [44] proposed an algorithm based on a High Frequency energy indicator to obtain the discriminating criteria for liveness detection. Their approach mainly consists of three stages. In the first stage, they select frames with the existence of maximal micro-expressions. After that, a 3D DFT is constructed and finally on the basis of high frequency content the liveness is detected. In addition, they used a modified high frequency descriptor for proper discrimination. Bharath and Kumar [45] studied the effect of glottal excitation and the high frequency band on the detection of Replay Attacks in an Automatic Speaker Verification system. First, the Iterative Adaptive Inverse Filtering (IA-IF) technique is used to eliminate the lip radiation and vocal tract effect, by applying integration and inverse filtering in order to extract the glottal excitation information of the speech spectrum, which allows to differentiate between the genuine and the replayed speech samples. Then, the constant-Q Cepstral coefficients method is applied to the high-frequency band of the glottal excitation speech spectrum, to finally obtain the desired features. In [46], Dua *et al.* published a study in which they analyzed the performance of three back-end models, namely: CNN, Long Short-Term Memory and a hybrid one, with different input feature formats in a multi-replay attack detection scenario on ASV systems. The Mel spectrograms are used as the feature extraction technique for the CNN-based model, whereas the Constant-Q Cepstral Coefficient approach is used for the extraction of features for the long-term memory-based model.

1.5 Dissertation outline

The rest of the dissertation layout is detailed as follows :

Chapter Two : Introducing Biometric Systems Attacks

This chapter serves as an introduction to biometric systems, offering a comprehensive understanding of their fundamental concepts. The chapter begins by defining biometric systems and exploring their diverse fields of application. Next, it delves into the operational mechanisms and general architecture of these systems. Then, it discusses the crucial aspects of evaluating biometric systems, including performance metrics and assessment methodologies. After that, it explains the vulnerabilities related to these systems, while also presenting proposed solutions aimed at mitigating potential risks and ensuring robust security.

Chapter Three: Perceptual Hashing

This chapter provides an extensive and in-depth description of perceptual hashing, providing a thorough understanding of its fundamental principles, key properties, and wide-ranging applications. In addition to covering the theoretical aspects, this chapter presents a comprehensive overview of the current state-of-the-art approaches

and techniques used in the field of perceptual hashing. It highlights the latest advancements and innovations made in perceptual hashing techniques, showcasing the progress and ongoing research in this rapidly evolving field. The main objective of this chapter is to provide the readers with the knowledge and insights necessary to comprehend its significance and potential impact across various domains.

Chapter Four: Perceptual Hashing : A comparative study

In this chapter, we conduct a comprehensive analysis and performance evaluation of five widely used perceptual hashing techniques applied specifically to fingerprint images, with a specific focus on their ability to authenticate image integrity. The evaluated techniques encompass both spatial and frequency domain approaches. In the spatial domain, we thoroughly assess the effectiveness of two techniques: Average hashing (A-Hash) and Difference hashing (D-Hash). These methods operate by analyzing the pixel values of fingerprint images and computing average and difference values, respectively, to generate corresponding perceptual hashes. In the frequency domain, we tackled three techniques: DCT-based hashing (P-Hash), Wavelet-based hashing (W-Hash), and SVD-based hashing (SVD-Hash). These methods leverage transforms such as Discrete Cosine Transform (DCT), Wavelet Transform, and SVD to extract frequency-based features from fingerprint images. These features are then used to generate perceptual hashes, enabling further analysis and verification of image authenticity. By thoroughly examining and evaluating these perceptual hashing techniques, we aim to provide a comprehensive understanding of their performance and suitability for fingerprint image verification. The inclusion of both spatial and frequency domain approaches allows for a comparative analysis of their respective strengths and weaknesses, providing valuable insights into their practical applicability in real-world scenarios.

Chapter Five: A Replay Attack Detection Scheme based on Perceptual Image Hashing

In this chapter, we present our proposed solution that rely on the application of perceptual image hashing to enhance the security of biometric systems against digital replay attacks. The proposed solution is a novel Replay Attack Detection (RAD) scheme that is capable of verifying the authenticity of the received biometric images and detects any attempts at resubmission. The proposed RAD operates effectively in uncontrolled environments, without the need of the cooperation of end users. The RAD scheme combines the Challenge/Response principle with perceptual image hashing techniques and incorporates data hiding methods to achieve robust protection. Additionally, we introduce in this chapter a new perceptual image hashing system that is specifically designed for fingerprint images. This system offers comprehensive content authentication and identification capabilities. It harnesses a shift-invariant calibration signal technique, leveraging the power of both the DCT and the Discrete Sine Transform (DST) applied to the Differential Block Luminance

Mean (DBLM) features. Through this process, a hash sequence is generated, enabling reliable and efficient verification of the integrity and uniqueness of fingerprint images. By presenting our innovative RAD scheme and the dedicated perceptual image hashing system for fingerprints, we aim to provide an enhanced level of security for biometric systems, safeguarding them against the potential threats posed by digital replay attacks.

Chapter 2

Introducing Biometric Systems Attacks

2.1 Introduction

Nowadays, biometric systems are introduced in multiple fields and at different scales. These systems rely on biometric characteristics to determine or to confirm a person's identity. These systems, therefore have several advantages over traditional authentication systems, which are based on things we know (passwords or PIN codes) or means we have (cards or badges). Despite their advantages, biometric systems have vulnerabilities that can be exploited to impact their functionality and significantly degrade the value of deploying them. Thus, the evaluation of the security of biometric systems has become essential to guarantee the operationality of these systems. This chapter presents biometric systems in general. Firstly, we present the definition and fields of use of biometric systems. Thereafter, we approach the operation as well as the general architecture of a biometric system. Then, we expose the evaluation aspects of these systems. Finally, we present the vulnerabilities of these systems as well as the solutions proposed to counter them.

2.2 General Concepts

Biometrics can be defined as the science that uses unique physical, chemical, or behavioral characteristics to identify individuals [47]. However, it is important to note that there is no single, universally accepted definition of biometrics. Different authors and organizations may define biometrics in slightly different ways.
Biometrics is also the mathematical study of biological variations within a given group. The term 'biometrics' is often used interchangeably with the term 'anthropometry', but there is a subtle difference between the two terms. Anthropometry is the measurement of the physical dimensions of the human body, while biometrics is the measurement of physical characteristics that are unique to an individual, such as fingerprints, facial features, and iris patterns [48].
A biometric system is an automatic system used for the identification of a person, based on his/her own characteristics. These characteristics can be [49]:

- Morphological (physiological) characteristics: these are particular physical traits that are unique and permanent for any person; for example: fingerprints, hand shape, facial features, retina and iris, etc.
- Behavioral characteristics: which are linked to certain behaviors of an individual, for example: the dynamics of the signature, the voice, the gait, the way of typing on a keyboard, etc. However, it's important to note that all these characteristics change with age. They can be subject to change also due to a variety of constraints, such as injury, illness, emotional state, or even conscious effort to modify one's habits and behaviors. For example, an individual's gait may be altered by the onset of a physical condition, or their typing patterns could evolve as they become more experienced with using a keyboard.
- Biological characteristics: which are characteristics linked to the biology of the human being, and which are considered as individual signatures for each person; for example: DNA, smell, saliva, blood, urine, etc.

To be exploitable in the field of biometrics, a characteristic must satisfy the following criteria [50]:

- Universality: it must exist for each individual;
- Uniqueness: it must be different enough between individuals to be able to differentiate them;
- Permanence: it must be sufficiently invariant for a fairly long period of time;
- Measurability: it must be able to be measured quantitatively;
- Acceptability: it must have certain properties such as ease and speed of acquisition;
- Performance: refers to the accuracy of the authentication / identification;
- Circumvention: refers to the ease with which the biometric characteristic can be imitated or regenerated using physical means.

2.3 Applications of Biometric Systems

Biometric systems are used nowadays in different domains, indeed, all areas that require verifying or determining the identity of people are concerned. The applications of biometric systems can be roughly grouped into three categories as follows [51, 52]:

- **Legal Applications:** For justice and law enforcement, the biometric technology, especially fingerprint, has been used for a very long time in identity management purposes. These days, law enforcement are using multimodal based biometric system, i.e. fingerprint, face, and voice recognition to improve public safety and keep tracking criminals.

- **Government applications:** For these kind of applications, biometric systems are used mainly for the identity verification of passengers crossing the borders or in the airports. Also this technology is progressively introduced in the healthcare field to be used as a personnel document that can represent the patient medical records quickly and in an accurate manner.
- **Commercial applications:**
  - Security: Today, biometric technology is more and more used to secure the access to every thing that need to be protected from a car door to a phone's PIN. This technology has proven its reliability and efficiency over the old security methods.
  - Finance: In the financial sector, biometric technology finds extensive use in identification, verification, and authentication processes. By employing biometric traits like fingerprints, iris scans, voice patterns, and facial features, banking, purchasing, and account management become more secure and user-friendly, safeguarding sensitive financial data from unauthorized access and fraud.
  - Mobile: The proliferation of mobile devices has propelled the importance of mobile biometric solutions, which combine biometric measures with modern technology for versatile authentication and identification capabilities.
  - Eye movements tracking applications: Eye movement tracking applications have gained prominence in various domains as automotive industry to monitoring driver eye movements aids in assessing levels of sleepiness, tiredness, or drowsiness, contributing to road safety, for screen Navigation in the case of people with disabilities enabling to execte tasks such as scrolling web pages and interacting with computer or mobile interfaces, in aviation and flight simulators for tracking and analyzing pilot behavior and performance, facilitating pilot training and enhancing monitoring of aircraft indicators.

These diverse applications of biometric systems demonstrate their wide-ranging potential to improve security, efficiency, and convenience in a variety of sectors.

2.4 Architecture of a biometric system

According to the ISO/IEC 19795-1 standard [1], a biometric system consists of five modules as shown in figure 1.1:

- Capture module: consists of acquiring biometric data and extracting a digital representation called a sample, which will be processed later during the enrollment, verification or identification processes. The biometric sensor, like the fingerprint scanner, is the main component in the capture module, which can also contain other components, for example, a keyboard and a screen to enter other

information;

- Signal processing module : makes it possible to extract a set of compact and representative characteristics of the biometric data, called template or model , from the acquired sample; optimizing the amount of data to be processed and stored. In addition, this module may include a module for quality control of the acquired sample;
- Storage module : contains the templates previously inserted in the biometric system, as well as the demographic information of the registered persons. Depending on the application and the desired level of security, this module can be a central database or integrated into a personal element specific to each person (as in the case of the biometric passport);
- Similarity module: allows you to compare the template of the biometric modality presented to the system with one or more templates already registered. It thus indicates the degree of similarity or divergence between two biometric templates;
- Decision module: allows you to indicate whether the degree or the index of similarity returned is sufficient to confirm/disprove the identity of a person based on a previously established threshold means.

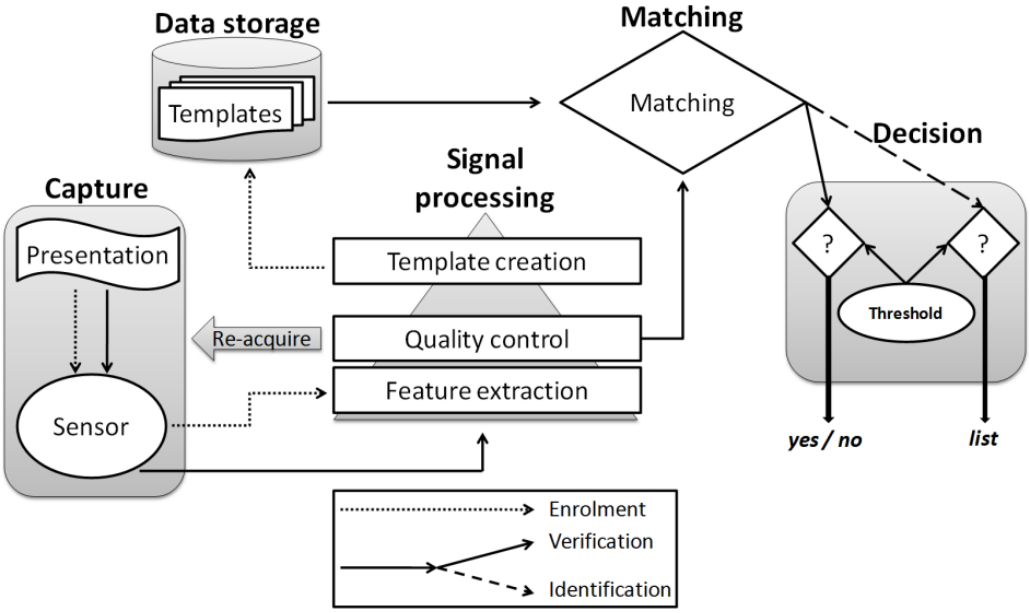


Figure 2.1: Generic Architecture of a biometric system [1]

2.5 General scheme of a biometric system

Generally, a biometric system offers a three main functionalities [53–55], namely enrollment, authentication, and identification as described bellow :

2.5.1 Enrollment:

The enrollment phase is the process of creating a reference biometric template for an individual. This template is a digital representation of the individual's biometric trait, such as his/her fingerprint, face, or iris. The template is created by capturing the biometric trait and then processing it using a biometric algorithm. The enrollment process typically involves the following steps:

1. The user presents his/her biometric trait to the biometric system.
2. The biometric system captures the user's biometric trait.
3. The biometric system processes the user's biometric trait to create a template.
4. The biometric system stores the template in a database.

The enrollment phase is an important part of any biometric system. The quality of the template created during the enrollment phase will affect the accuracy of the system. Therefore, it is important to ensure that the enrollment process is properly conducted.

2.5.2 Authentication:

Consists of verifying whether the individual using the system is the person he/she claims to be. The system compares the template extracted from the acquired biometric sample, called authentication template T_a, with the corresponding enrollment template stored in the storage module; so the system performs a one-to-one comparison. In this case, the biometric system only sends a binary decision (yes or no). The authentication process, can be formalized as follows:

- Extract the authentication template vector T_a from the biometric characteristic of an individual I acquired by the system
- Prepare the corresponding enrollment template stored in the storage module T_e .
- Calculate the function f defined by:

$$f(T_a, T_e) = \begin{cases} 1 & \text{if } S(T_a, T_e) >= T \\ 0 & \text{Otherwise} \end{cases} \tag{2.1}$$

  Where: S stands for the similarity function defining the correspondence between the two templates, and T is the decision threshold from which the two templates are considered identical.
- The system returns a Boolean value depending on the result obtained by the above equation.

2.5.3 Identification:

In this process, the biometric system determines, from the storage module, the identity of an unknown individual using one or more biometric characteristic(s). The system compares the template extracted from the biometric sample of this individual, called the identification template T_i, with all the enrolled templates stored in the database; it is therefore a one-to-many comparison. In this case, the system can either attribute to the unknown individual the identity corresponding to the closest profile found in the database (or a list of similar profiles), or attribute nothing and the individual remains unknown. The identification process, can be formalized as follows:

- Extract the the identification template vector T_i from a biometric characteristic of an unknown individual I .
- Calculate the identification function f as follow :

$$f(T_i) = \begin{cases} I_k & \text{if } max_{1<=k<=N} S(T_i, T_e) >= T \\ I_0 & \text{Otherwise} \end{cases} \tag{2.2}$$

  where: T_i is the biometric template corresponding to the identity of I_k, S is the similarity function, and T is the decision threshold.
- The identification purpose is then to find the identity of $I_k, k \in \{0,1,\ldots,N\}$ where $I_1,\ldots,I_N$ are the identities of individuals previously enrolled in the system, and I_0 indicates an unknown identity.

2.6 Evaluation of a biometric system

The evaluation of a biometric system is a crucial process that consists of determining its accuracy, reliability, and usability. The accuracy of a biometric system is its ability to correctly identify or verify a person. The reliability of a biometric system is its ability to consistently perform at a high level and the usability of a biometric system is its ease of use for the user.
Therefore, the three aspects to be taken into account during the evaluation a biometric system are mainly [53, 56, 57] performance, usage, and security.
There may be some trade-off between these aspects. Using more advanced methods (such as deep learning) can significantly enhance the system accuracy, but it can also increase computational complexity. Using passwords or credentials as a backup in case of an issue can weaken system security [58]. Tighter security measures can result in more legitimate users being denied access, which can decrease the system's accuracy.
Ultimately, attempting to fully address all aspects will raise costs. As a result, all of these aspects should be carefully examined and weighted, taking into account the desired application, in order to obtain an affordable, efficient, accurate, secure, and

usable system.

2.6.1 Usage Evaluation

The usage evaluation of a biometric system consists in measuring its ease, acceptability and satisfaction degrees seen by the user. This evaluation is crucial to ensure that the biometric system can perform its tasks effectively and meet the desired objectives. In general, the usage evaluation is built upon statistical analysis studies carried out based on user's answers to questionnaires, which elate essentially to two types of information: (i) socio-demographic questions consisting of collecting information about the individual such as age, sex, occupation, etc. ; and (ii) perception questions that aim to measure users' experience of biometrics, their perception and satisfaction when using a biometric system.

2.6.2 Performance evaluation

Performance evaluation is an important step for any biometric system [59]. This phase aims to evaluate the strength and weakness of the system using standard databases and well-defined performance statistics measures.
The performance of a biometric system can be measured primarily using three criteria: accuracy, efficiency (execution speed), and the volume of data that must be stored for each person [60].
Performance evaluation measures the effectiveness and reliability of a biometric system in a specific use case by using metrics to quantify the results of the various processes of a biometric system, which are the rate measures errors, measurements related to processing time and memory occupation, performance curves, etc.
In practice, biometric systems are less precise in terms of similarity (i.e. unlike the similarity index of traditional systems which is a Boolean value, it is given as a percentage, which varies between 0 % and 100%, where 100% is almost never reached for all biometric systems). This is due to the following factors [53]:

- variability during data capture (ie, acquisition noise, use of different sensors, acquisition environment, etc.)
- intra-class variability (ie variability of biometric data of the same person)
- interclass variability (ie similarity of biometric data from different people).

2.6.3 Security evaluation

The security of any Information System is assessed generally based on the Common Criteria (CC) methodology. These Common Criteria are standardized by the ISO 15408 [61]. Although biometric systems are basically information systems,these criteria are not sufficient to fully evaluate them, due to their unique characteristics [56]. In addition to the methods and models proposed in the literature, the ISO/IEC FCD 19792 standard [62] defines the main requirements for evaluating biometric

systems security. They almost conclude that vulnerability analysis and performance testing are the two essential elements when evaluating the security of biometric systems. The security evaluation of a biometric system is discussed in more detail in the next section.

2.7 Security concerns of Biometric Systems

Biometric systems are often considered as more reliable and secure than password or token-based systems, However research has shown that they are not entirely secure and have their own vulnerabilities [2, 63]. One of the most privacy concerns is biometric data theft. Unlike handheld tokens or knowledge-based verification methods, A key drawback of biometric identifiers is that they cannot be easily revoked and replaced if compromised [64] .
Other security concerns related to biometric systems include:

- **Compromised biometrics:** If the biometric data of the individual is compromised, it is then extremely difficult, if not impossible to change it, so it can be used by the imposter for future authentication.
- **Potential fraud by administrators:** Administrators of biometric systems could potentially abuse their power to commit fraud or identity theft.
- **Denial of service attacks:** An attacker could launch a denial of service attack against a biometric system, making it unavailable to legitimate users.
- **Intrusion attempts:** An attacker could try to hack into a biometric system to gain access to sensitive data or to launch other attacks.
- **False acceptance of impostors:** Biometric systems can sometimes falsely accept impostors, which could allow unauthorized individuals to gain access to sensitive systems or accounts.
- **False rejection of genuine users:** Biometric systems can also sometimes falsely reject genuine users, which could prevent them from accessing authorized systems or accounts.
- **Interception of biometric data:** If biometric data is transmitted over an insecure channel, it could be intercepted by an attacker.

In addition to these security concerns, biometric systems can also be subject to technical limitations. For example, fingerprint scanners can be forged by fake fingerprints, and facial recognition systems can be fooled by masks or photos.

2.7.1 Vulnerabilities and attacks models

Roberts [65] categorized the source of vulnerabilities related to the biometric system into three types, namely, **threat agent**, **threat vectors** and **system vulnerabilities**. A **threat agent** is anyone who gains unauthorized access to a biometric system. This

could be an attacker, an impostor, or even a legitimate user who has compromised credentials. All possible attacks on a biometric system form the **threat vectors**.
In the literature, several research works have demonstrated that biometric systems are vulnerable to specific attacks [2–5, 66], such as spoofing, replay, and tampering. These attacks can be lunched at any point in the biometric system life cycle, from enrollment to authentication. Researchers have also proposed models to identify and classify these vulnerabilities, and to assess their impact on the security and functionality of biometric systems. These threat models are described in the next sub-section.

2.7.2 Threat Models

In this section, we present five threat models for fingerprint biometric systems, that identify and assess the potential threats to a biometric system. Each model considers the attacker's capabilities, motivations, and objectives, as well as the system's vulnerabilities and assets. Threat models can be used to develop security controls and mitigation strategies to protect biometric systems from potential attacks.

2.7.2.1. Joshi et al. model

Joshi et al. [2] identify 16 attack points (APs) on fingerprint biometric systems, as shown in Figure 2.2. These APs target the system components and the insecure communication channels between them. The possibility of attacks is mainly attributed to design flaws in the system. One unique aspect of Joshi et al.'s threat model is that it includes the template protection techniques module and identifies the possibility of attacks on this module that aim to reveal the biometric key for the cryptosystem. The model also includes successful side-channel attacks targeting the circuit technology implementation of the biometric system. The 16 APs of Joshi et al. model are categorized as follows [2]:

- **A. Direct and Indirect Attack :**

Joshi et al. [2] classified Biometric attacks into two primary categories: direct attacks, also referred to as presentation attacks, and indirect attacks. In the case of direct attacks, the imposter's objective is to disrupt the normal operation of the biometric system by targeting either the input device or the application that controls its functioning. This disruption often involves physical damage or tampering, resulting in the inaccessibility of the system even to authorized users. In Figure 2.2, these kind of attacks are denoted as attack points, AP 1, AP 2, and AP 16.
On the other hand, indirect attacks encompass a different modus operandi, focusing on intercepting information transmitted over communication channels or manipulating the internal components of the biometric system. These attacks aim to override the system's output or tamper with the stored biometric templates. Attack points

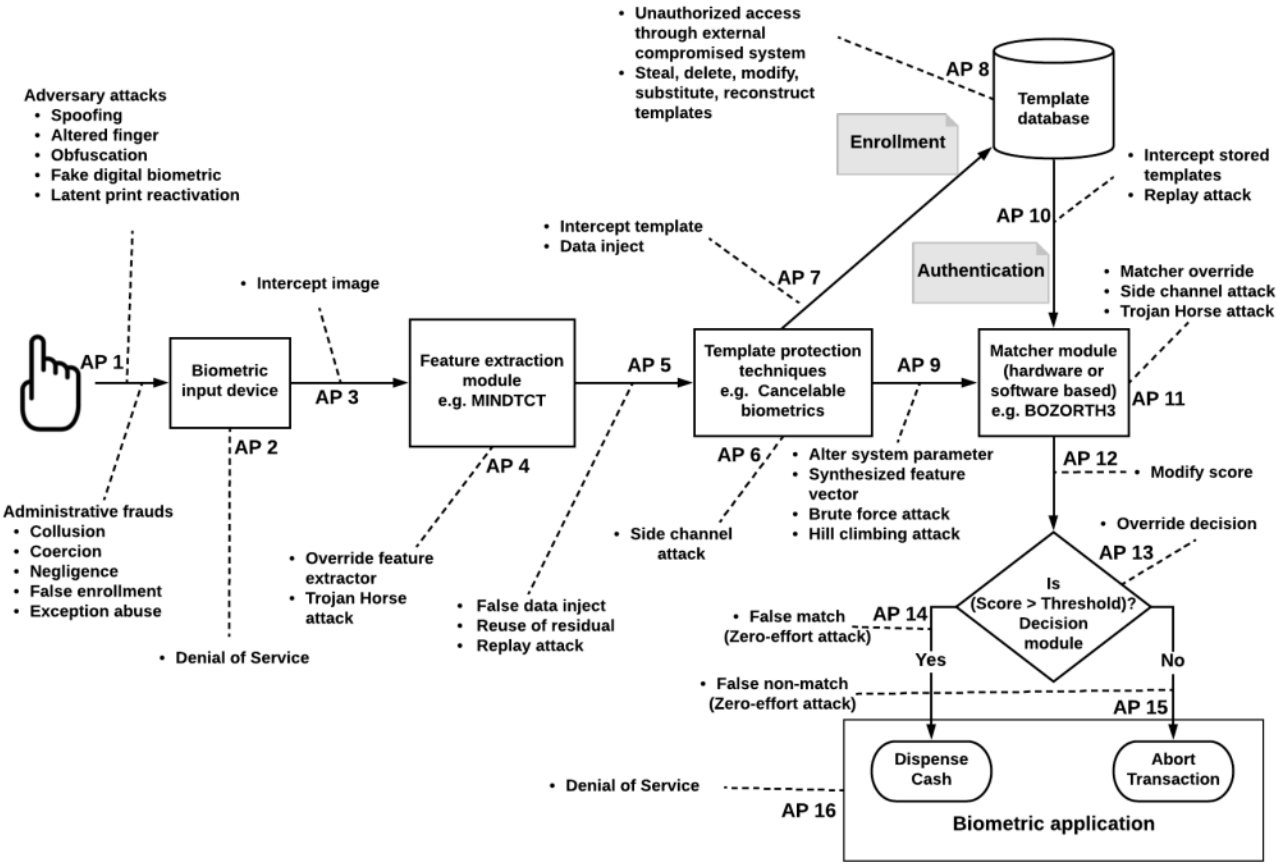


Figure 2.2: Joshi's et al. Threat Model : A number corresponding to an arrow or a dashed line represents an attack point (AP) [2].

other than those mentioned earlier (AP 1, AP 2, and AP 16 in Figure 2.2) are classified as indirect attacks.

- **B. Repudiation:**

Biometric systems are inherently vulnerable to attack. Adversaries can exploit this vulnerability by denying access to the system by claiming that a false acceptance has occurred. For example, a bank technician who withdrew cash using the biometric application shown in Figure 2.2 could argue that the system falsely accepted their fingerprint.

- **C. Contamination or covert acquisition :**

The imposter can use the fingerprint traces, that the authorized user could accidentally leave behind him, to generate a fake fingerprint using spoofing techniques or constructing a mold by lifting latent fingerprint to mount the attack. This can be the case of a user dispensing cash from a biometric application similar to the one described in Figure 2.2.

- **D. Coercion:**

An attacker can force a legitimate user to provide his/her biometric data to the sensor, such as at gunpoint. The attacker, then uses the acquired biometric credentials to accomplish financial transactions. This type of attack is especially likely to occur

on ATMs with poor security, such as those located in dangerous areas and those that lack security personnel or CCTV cameras.

- **E. Negligence:**

The legitimate user can neglect to log out from the biometric application, while an imposter is noticing him, and can therefore capitalize on such a moment of user negligence by continuing the active session, masquerading as an authorized user. In this guise, the attacker gains unauthorized access to the system and can potentially carry out additional transactions or access sensitive user information.

- **F. Insider Attacks:**

Insider attacks are attacks carried out by authorized users or system administrators. These attacks can be particularly difficult to detect and prevent. Examples of insider attacks against biometric systems include:

1. **Collusion:** A system administrator with full access privileges may collude with an attacker to compromise the biometric system. This could involve illegally accessing and modifying system parameters, such as the predefined threshold, or enrolling an attacker as a legitimate user.
2. **Exception abuse:** The exception handling procedure designed to facilitate authorized users in case of emergency can be exploited by an insider to help an attacker access the system. For example, a system administrator could reset the predefined threshold to a lower value, allowing the attacker to succeed despite a false match error.
3. **Function creep:** A system administrator may collect biometric samples of an individual in excess of what is needed for authentication purposes. This data may then be sold to private organizations for financial or other malicious reasons.

- **G. Sensor Attacks:**

Sensor attacks are attacks that target the input device or the application controlled by the biometric system. These attacks are particularly difficult to mitigate because they occur at the very entry point of the system, where digital security mechanisms such as cryptosystems are ineffective. This kind of attacks include:

1. **Spoofing attacks:** In a spoofing attack, the attacker presents a fake biometric to the sensor. This can be done using a variety of methods, such as using a fake fingerprint, facial mask, or iris lens.
2. **Masquerade attacks:** In a masquerade attack, the attacker uses a legitimate user's biometric data to impersonate that user. This can be done by stealing the user's biometric data or by using a synthetic biometric.
3. **Altered fingerprint submission:** In an altered fingerprint submission attack, the attacker alters their own fingerprint to make it more difficult for the biometric

system to identify them. This can be done using a variety of methods, such as using an abrasive material to remove the ridges of the fingerprint.

4. **Obfuscation attacks:** In an obfuscation attack, the attacker attempts to hide their true biometric identity from the biometric system. This can be done by altering their biometric features prior to their acquisition by the sensor. For example, an attacker might mutilate their fingerprints by using an abrasive material.
5. **Fake digital biometric attacks:** In a fake digital biometric attack, the attacker presents a fake digital biometric template to the biometric system. This can be done by replaying a previously captured biometric template or by creating a synthetic biometric template.
6. **Fake physical biometric attacks:** In a fake physical biometric attack, the attacker presents a fake physical biometric to the biometric sensor. This can be done using a variety of methods, such as using a fake fingerprint or facial mask.
7. **Latent print reactivation attacks:** In a latent print reactivation attack, the attacker uses a latent fingerprint that has been left on the biometric sensor to create a readable print. This can be done using a variety of methods, such as using a powder or fuming agent.

- **H. Feature extractor attacks:**

Feature extractor attacks target the feature extraction module of a biometric system. The feature extraction module is responsible for extracting the unique features from a biometric sample. By attacking this module, an attacker can cause the system to generate incorrect feature vectors, which can then be used to spoof the system or to gain unauthorized access. Examples of feature extractor attacks against biometric systems include:

1. **Override feature extractor:** In an override feature extractor attack, the attacker replaces the feature extraction module with a malicious module that generates pre-determined feature vectors.
2. **Trojan horse attack:** In a Trojan horse attack, the attacker injects malicious code into the feature extraction module. This code can then be used to generate incorrect feature vectors or to steal biometric data. The Trojan horse could generate a pre-programmed feature set to be fed as input to the template protection techniques module. This would allow the attacker to bypass the template protection techniques and gain unauthorized access to the system.

- **I. Attacks on template protection techniques module:**

The template protection module is responsible for converting a feature vector into a cryptographically secured biometric template. This module is a potential target

for attackers, who may aim to extract information related to the key used in the encryption algorithm. One type of attack on the template protection module is **a side channel attack (SCA)**. SCAs exploit information leakage from the implementation of a cryptosystem to recover the secret key. For example, an attacker may measure the power consumption or timing of the cryptosystem to extract information about the key.

- **J. Matcher module attacks:**

The matcher module is responsible for comparing two biometric templates and determining whether they match. Attackers may target the matcher module in an attempt to achieve a matching score above the predefined threshold for a non-enrolled fingerprint. Examples of matcher module attacks against biometric systems include:

1. **Hill climbing attack:** In a hill climbing attack, the attacker iteratively modifies a synthetically generated template or group of templates until the verification threshold is reached. The attacker uses the similarity score generated by the matcher to guide the modification process.
2. **Matcher override or false match:** In a matcher override attack, the attacker modifies the matching score generated for a genuine template comparison to be above the predefined threshold. This can be done by tampering with the matcher algorithm or by injecting malicious code into the matcher module.
3. **Side channel attacks:** Side channel attacks exploit information leakage from the implementation of a cryptosystem to recover the secret key. In the context of matcher module attacks, side channel attacks can be used to extract information about the matching algorithm. This information could then be used to manipulate the matcher algorithm to generate a match for a non-enrolled fingerprint.
4. **Trojan horse attack:** In a Trojan horse attack, the attacker replaces the vendor's matching algorithm code with a malicious matching algorithm. The malicious matching algorithm could be programmed to generate a match for a non-enrolled fingerprint.

- **K. Template database attacks:**

The template database is a critical component of any biometric system. It stores the biometric templates of all enrolled users. As such, it is a prime target for attackers. In this kind of attacks, the imposter can perform a man-in-the-middle attack, such as a replay attack. In addition, the attacker has the capability to execute a denial-of-service (DoS) attack by erasing stored templates. The Attack Point AP8 in Figure 2.2 illustrates an example of an attack targeting the database. This kind of attacks include :

1. **Template reconstruction:** In a template reconstruction attack, the attacker breaks into the online database where the encrypted biometric templates are stored and

modifies the existing encrypted biometrics or reconstructs a new template from it. The attacker can then use the reconstructed template to impersonate a legitimate user and mount a hill-climbing attack at AP 9 in Figure 2.2.

2. **Unauthorized template modification:** In an unauthorized template modification attack, the attacker steals or modifies the biometric templates stored in the database. The attacker can then use the stolen or modified templates to impersonate legitimate users or to disrupt the operation of the biometric system.
3. **Circumvention:** Circumvention attacks involve exploiting vulnerabilities in the biometric system to gain unauthorized access to the template database or to manipulate the system data. For example, an attacker could exploit a vulnerability in the biometric application to gain access to the template database or to modify the template data of a legitimate user.

- **L. Man-in-the-middle (MiTM) attack:**

A man-in-the-middle (MiTM) attack is a type of cyberattack where the attacker intercepts the communication between two parties and impersonates one of them. In the context of biometric systems, an attacker could use a MiTM attack to steal biometric data or to impersonate a legitimate user. In this case, the attacks points AP 3, AP 5, AP 7, AP 9, AP 10, and AP 12 represent examples of MiTM attacks against biometric systems, they include:

1. **Replay attack (AP 5):** In a replay attack, the attacker intercepts a valid biometric authentication and then replays it at a later time to gain unauthorized access.
2. **Reuse of residuals (AP 5):** In a reuse of residuals attack, the attacker uses biometric data that has been left in the memory of a biometric device to impersonate a legitimate user.
3. **System parameter override or modification (AP 12):** In a system parameter override or modification attack, the attacker changes the system parameters of a biometric system to make it more likely to accept a false biometric.
4. **Synthesized feature vector attack (AP 9):** In a synthesized feature vector attack, the attacker creates a fake biometric template and then injects it into the biometric system.
5. **Brute force attack (AP 4):** In a brute force attack, the attacker tries to guess the biometric template of a legitimate user by submitting a large number of possible templates to the biometric system.
6. **Storage channel intercept and data inject attack (AP 7 and AP 10):** In a storage channel intercept and data inject attack, the attacker intercepts the communication between the biometric database and the matcher module and then injects false data into the communication.

7. **Modify score attack (AP 12):** In a modify score attack, the attacker changes the matching score generated by the matcher module to make it more likely that a false biometric will be accepted.

- **M. Modify access rights :**

The system's security is compromised when a user, originally registered with limited access rights, is granted administrative privileges. In both scenarios, the adversary must first obtain the system administrator's credentials.

- **N. Decision override :**

An attacker can employ a Trojan horse to manipulate the decision module, skewing the outcomes in their favor. For instance, this kind of attack can be utilized to grant unauthorized access to the application depicted in Figure 2.2 or to deny access to other legitimate users.

- **O. Denial-of-service (DoS) attack:**

A potential threat involves launching a DoS attack against an online authentication server responsible for processing access requests. This is achieved by inundating the server with numerous fraudulent access requests, causing it to exhaust its computational resources to the extent that it becomes incapable of handling legitimate requests.

- **P. Intrusion :**

The act of an attacker gaining unauthorized access to the template storage through an external system is categorized as intrusion. Additionally, scenarios in which the system mistakenly recognizes an impostor as an authorized user are also considered instances of intrusion.

- **Q. Zero-effort attacks:**

Zero-effort attacks occur when predefined thresholds for biometric verification are adjusted to either lower or higher values. As shown in Figure 4 (AP 14 and AP 15), these attacks exploit the limited individuality and intraclass variations present in biometric features. Research efforts are currently underway to reduce such inherent errors by developing salient-feature detectors and robust matchers [5]. In this context, an online platform called EvaBio has been introduced by [67] for the security evaluation of biometric systems.

2.7.2.2. Ratha et al. model

Ratha et al . [3] analyzed possible attacks on a generic biometric system and grouped them into eight types. Figure 2.3 illustrates the locations of these types of attacks:

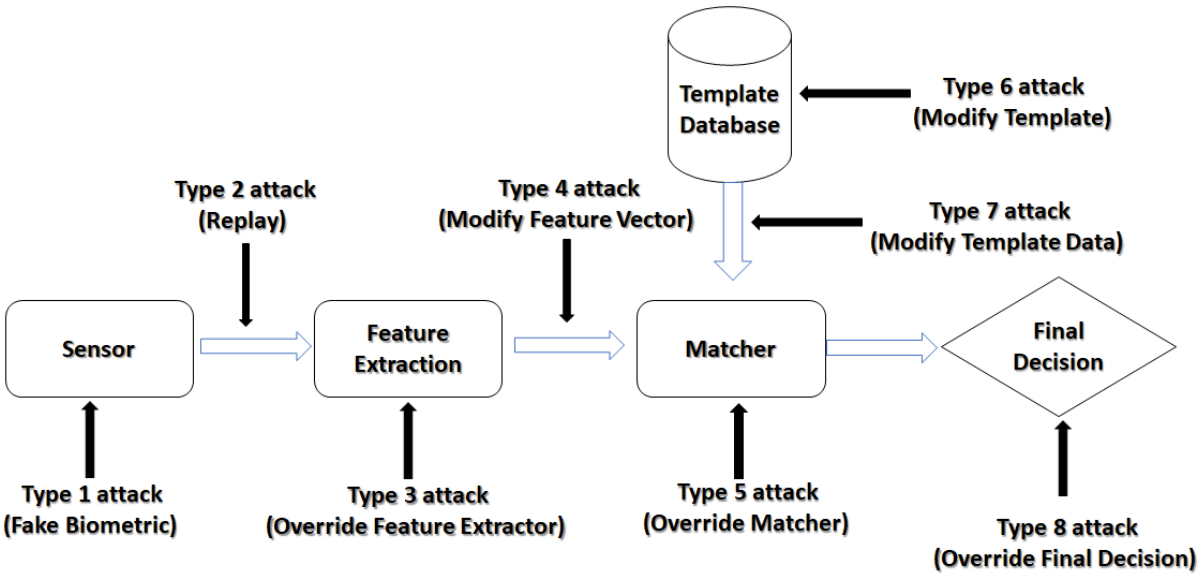


Figure 2.3: Ratha et al. [3] Threat Model : The arrows and numbers indicate the access point of information for mounting a specific type of attacks

- **Type (1) :** Attacks on the acquisition module: consists of presenting falsified biometric data to the capture module (such as a synthetic reproduction of a fingerprint).
- **Type (2) :** Attacks on the communication channel between the acquisition and extraction modules: consisting of replaying old biometric data previously recorded in the system by bypassing the biometric sensor (such as a fingerprint image or a recording sound).
- **Type(3) :** Attacks on the feature extraction module: consists of forcing the module to produce the features chosen by the attacker by infecting the system with malicious programs (generally Trojan horse).
- **Type (4) :** Attacks on the communication channel between the extraction and comparison modules: consisting in altering the results returned by this module, by modifying them or replacing them with other characteristics chosen by the attacker.
- **Type (5) :** Attacks on the comparison module: consists of altering this module with malicious programs (Trojan horse) to generate scores determined by the attacker.
- **Type (6) :** Attacks on the storage module: consists of altering (adding, modifying or deleting) one or more templates which are stored in the templates database in order to authorize access to an imposter or to prevent a legitimate users to access it.
- **Type (7) :** Attacks on the channel between the storage module and the Matcher: consists in altering the templates transmitted via this channel.

- **Type (8) :** Attacks on decisions: consists of changing the final decision produced by the comparison module, if successful, this attack would simply render the biometric system useless.

2.7.2.3. Adler model

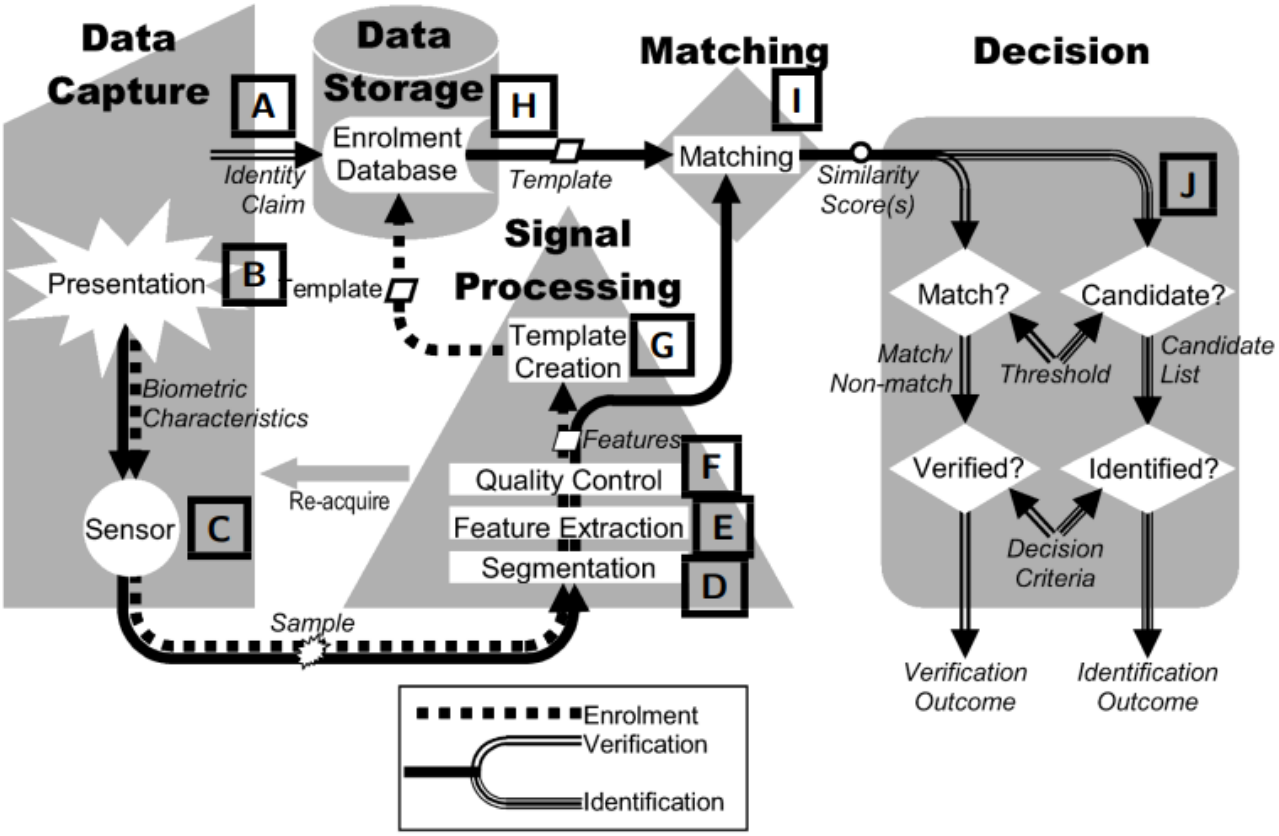


Figure 2.4: Adler's ten Vulnerabilities Model [4].

Adler's threat model [4] is a comprehensive framework that identifies and classifies ten (10) various attacks based on their characteristics and objectives in the context of biometric systems. The model, depicted in Figure 2.4, describes the various stages and components involved in the processing of biometric data within a biometric system and helps in understanding the potential threats and vulnerabilities associated with the use of biometrics for authentication purposes. In the Adler's model, each biometric sample (B) is acquired by a sensor (C) and then processed using segmentation (D) and feature extraction (D) algorithms to extract the unique features of the sample. If available, a quality assessment algorithm (E) is used to determine whether the sample is of sufficient quality to proceed. The extracted biometric features are then encoded into a template, which is a compact digital representation of the sample. The template can be stored in a database (H), on an identity card, or in secure hardware [4]. During enrollment, biometric samples are linked to a claimed identity (A). During subsequent verification or identification, new biometric samples are

tested against the enrolled templates using a matching algorithm (I). The matching algorithm generates a similarity score, which is used to make an identity decision (J). This decision can be made automatically or by a human agent reviewing the biometric system outputs.
Steps (A) to (J) are analyzed in the next sub-sections [68].

- **Identity Claim (A):** Identity claims play a crucial role in most biometric security systems, although they are not directly related to biometric properties. Identity claims typically rely on connections to government-issued identity documents, making them susceptible to various forms of document fraud. This vulnerability extends even to highly secure documents like passports, or birth certificates witch are issued by local government, hospitals, or religious authorities.
- **Presentation (B):** An attack on the biometric sensor, also known as the capture module, aims to introduce a fake biometric sample into the system. These attacks have two objectives: either to avoid detection, resulting in a false negative, or to impersonate someone else, leading to a false positive. The latter type of attack is commonly referred to as "spoofing." Generally, avoiding detection is easier than spoofing because changing the modalities significantly can create confusion in the segmentation module or the feature extraction module. Simple techniques like altering makeup, facial hair, wearing glasses, or wetting fingers have often been successful in evading detection. However, recent advancements in biometric algorithms have significantly reduced the effectiveness of these attacks.
- **Sensor (C):** Attacks on the biometric sensor involve techniques that alter or replace the hardware components of the sensor. Tampering with the sensor can potentially allow complete bypassing of the biometric system. In many cases, an attack on the sensor takes the form of a replay, where the connection between the sensor and the biometric system is manipulated to introduce biometric samples belonging to legitimate users.
- **Segmentation (D):** Segmentation is responsible for detecting and extracting the relevant image or signal from the biometric sample. A segmentation failure occurs when the system fails to detect valid biometric data. Segmentation attacks can be employed to evade surveillance or generate denial of service attacks.
- **Feature Extraction (E):** Attacks on the feature extraction module can be used to evade detection or create imposters. Knowledge of the extraction algorithms can be exploited to design specific features from acquired samples, leading to the calculation of false features.
- **Quality Control (F):** Evaluating the quality of biometric samples is essential to ensure low error rates. During enrollment, most systems assess the quality

of acquired samples. Attacks on the quality control algorithm can occur in two ways: classifying a good quality sample as bad or classifying a poor quality sample as good. The objective of the attack in the first case is to avoid detection, as poor quality images are not used for matching. In the second case, poor quality images are deliberately enrolled in the system.

- **Creation of Templates (G):** Biometric characteristics are encoded in templates, which are compact digital representations of essential biometric features. Template creation is typically considered a one-way function, making it infeasible to regenerate a sample from its template. However, recent research has shown that it can be possible to regenerate biometric samples from templates.
- **Data Storage (H):** Enrolled biometric templates are stored as references for authentication and identification operations. Vulnerabilities related to stored templates include altering the templates (adding, modifying, or deleting), duplicating templates for unauthorized use (identity theft), or changing the assigned identity. The data in a template can be stored in clear, encrypted, or digitally signed formats.
- **Comparison (I):** The comparison module calculates a similarity score that indicates the likelihood of two biometric samples belonging to the same person. While attacks at this level are rare and obscure, they can be possible in certain cases.
- **Decision (J):** Human operators are often responsible for reviewing the decisions of the biometric system. These operators are susceptible to fatigue and boredom, which can be exploited by attackers. For example, an attacker could launch a denial-of-service attack to overwhelm the system with requests, causing it to produce many errors. This would force the operators to abandon the system or to become suspicious of its results, making them more likely to override the system's decisions in favor of the attacker.

2.7.2.4. Jain et al. model

Jain et al. [5] proposed the Fish-bone model, that provides a cause-and-effect analysis of the vulnerabilities in a biometric system. The model identifies five causes as shown in Figure 2.5, represented in ovals, that contribute to the vulnerabilities, with arrows indicating the corresponding effects or possible attacks resulting from these causes.

- **Intrinsic Failure:** This cause encompasses the limitations and errors within the system that lead to false acceptance of a non-enrolled individual or false rejection of an enrolled user. Intra-user variations, such as improper finger placement on the sensor or lack of individuality/uniqueness (particularly in face recognition systems with twins), can result in false rejections.

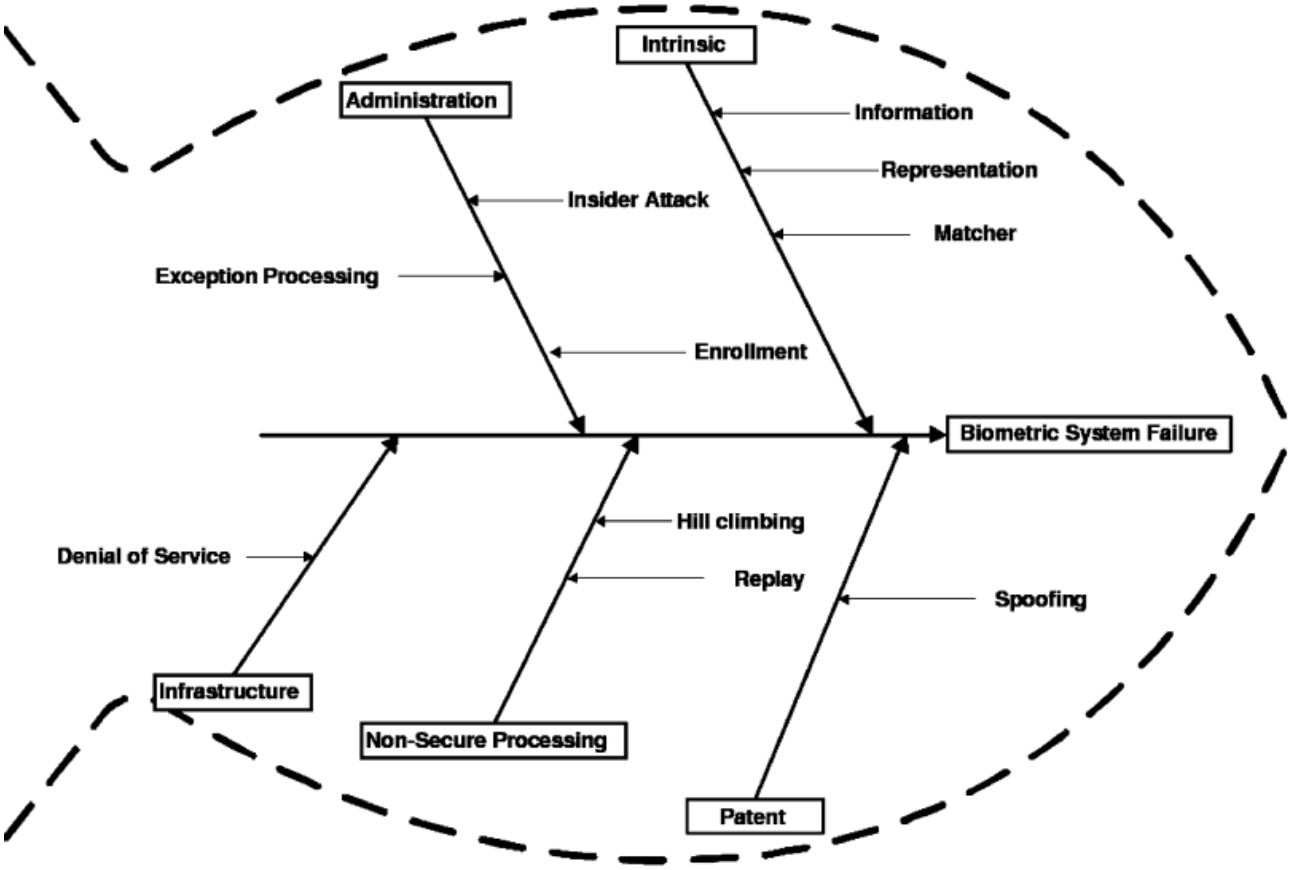


Figure 2.5: The Fiche-bone threat model proposed by Jain et al. [5]

- **Administration:** This cause relates to dishonest staff, particularly system administrators, who can act as adversaries. A disloyal administrator can assist in mounting successful attacks, engaging in enrollment fraud, or abusing exception processing mechanisms. Exception processing allows enrolled users to gain authorization in cases of intrinsic limitations or denial of service attacks (e.g., deleting their stored templates). Insider attacks involve cooperation between adversaries and disloyal staff to target the system.
- **Infrastructure:** This cause involves system design defects that make the system vulnerable to attacks. Hardware components, such as sensors, along with software modules like feature extraction and matching, as well as communication channels between system components, constitute the infrastructure of the biometric system. Adversaries may utilize Trojan horse programs to override the matcher or decision module.
- **Non-Secure Processing:** This cause highlights vulnerabilities arising from insecure enrollment and authentication processes. Adversaries can launch hill-climbing or replay attacks through non-secure communication channels, intercepting information and gaining unauthorized access without going through the sensor. Poorly secured databases can lead to leaks of stored templates, which adversaries can exploit for replay attacks.

- **Patent Cause or Biometric Overtness:** This cause pertains to the secrecy of biometric templates. Attackers covertly collect the biometrics of genuine users from sensors or public places they visit. These biometrics can be used to create fake physical or gummy fingers for spoofing attacks.

The Fish-bone model emphasizes the importance of avoiding general errors and implementing security techniques when designing a biometric system. The authors suggest that one effective approach to secure the system is to store templates and system modules on smart cards, resulting in systems known as "system-on-card" or "match-on-card" solutions.

2.7.2.5. The Bartlow and Cukic Framework

The Bartlow and Cukic framework [6], as an extension of the model proposed by Ratha et al. [3] and Wayman's subsystem architecture [66], focuses on the technical testing of biometric devices. The framework decomposes the model into three modules and identifies over twenty (20) vulnerable attack points. The three modules are the administrative observation/system management module, IT environment module, and the biometric subsystem module, as shown in 2.6. The biometric subsystem module further categorizes into five subsystems based on Wayman's suggestion. Attack points and vulnerabilities are represented by arrows in the subsystems and modules of the system.

- **The administrative observation/system management module :** in this module, the disloyal system administrators are represented, and attack points AP 1, AP 2, AP 18, AP 19, and AP 20 target various components of the data collection module and the manipulation of final decisions.
- **The IT environment subsystem :** includes applications like the operating system and database management system that interact with the biometric system. Attack point AP 3 indicates the possibility of attacks through these interacting applications, such as intrusion or execution of malicious code.
- **The biometric subsystem :** focuses on internal components, data transmission, and vulnerable points of the biometric system. The data collection module involves the collection and presentation of biometrics, and attack points AP 4, AP 5, and AP 6 represent vulnerabilities through input devices, susceptible to spoofing, replay, and masquerade attacks.

  The transmission module handles the compression and transmission of biometric data, and attack points AP 7, AP 8, AP 9, AP 10, and AP 11 indicate vulnerabilities to attacks like replay, parallel sessions, and masquerade, where intercepted information can be used for replay, hill-climbing, or brute force attacks.

  The signal processing module includes feature extraction, template matching,

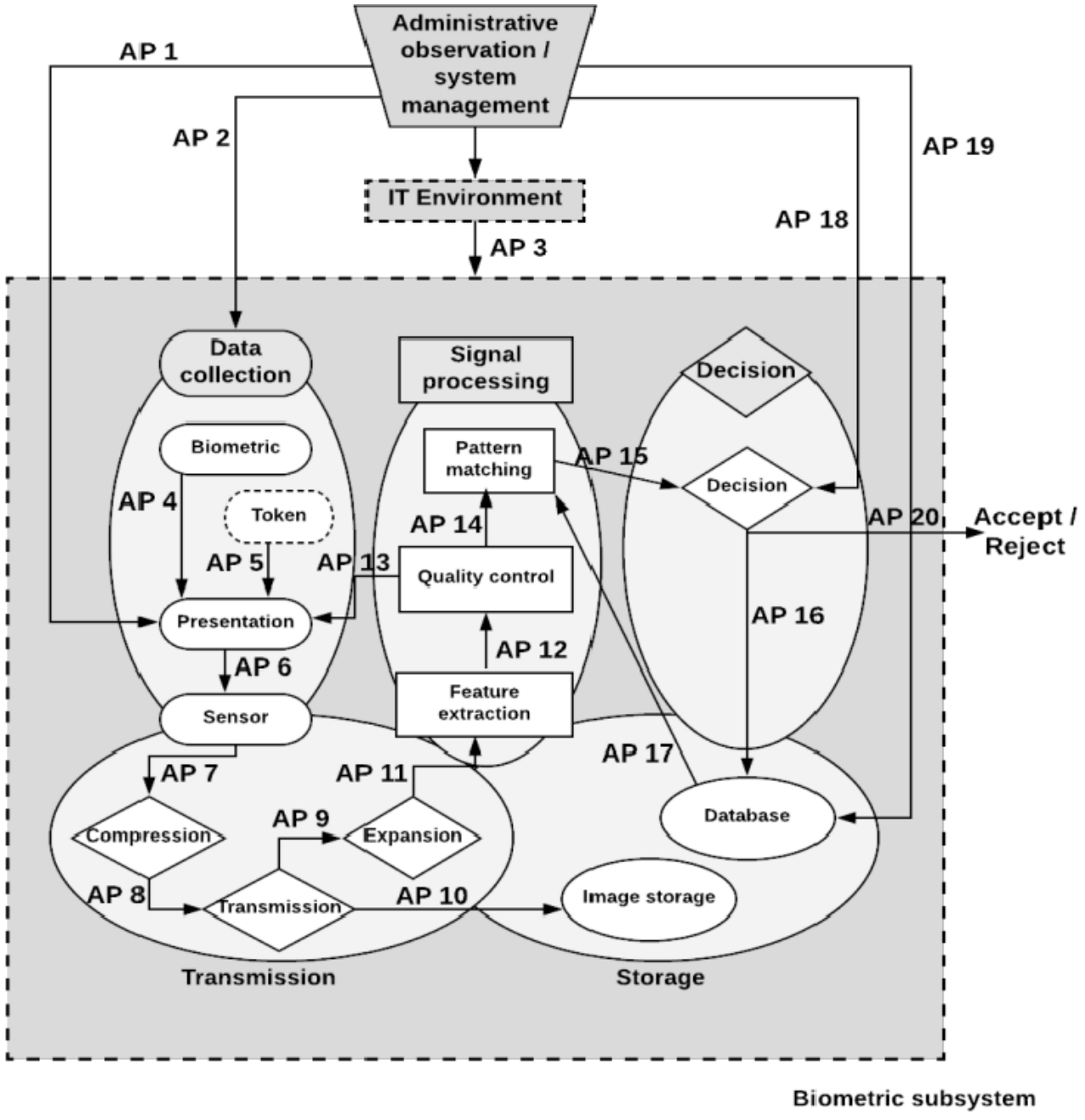


Figure 2.6: The Bartlow and Cukic threat Model [6]

and image quality control components. Attack points AP 12, AP 13, AP 14, and AP 15 represent vulnerabilities to attacks targeting the feature extractor, quality control unit, and interception of information between the pattern matching and decision modules. Attack points AP 16 and AP 17 involve capturing data through database leakage and retransmitting it to the matcher module. AP 20 represents an attempt to override the final conclusion of the decision module in cooperation with a dishonest administrator.

It's important to note that while this framework identifies potential vulnerable attack points, it does not propose specific methodologies for designing security techniques in a biometric system. Instead, it serves as a benchmark for testing and validating existing and proposed security techniques.

2.7.3 Suggested Solutions

In the literature, various solutions have been proposed to address the security vulnerabilities of biometric systems and mitigate their associated flaws. The following is a summary of the most recurring solutions for different types of attacks:

2.7.3.1. Solutions to Biometric Vulnerabilities

To address the irreplaceable and non-concealable of biometric characteristics, a new approach involves applying transformations to biometric samples and templates to generate alternative representations of the data. These transformed representations can be easily canceled or replaced when necessary. For example, in the case of fingerprints, Lee et al. [69] proposed a technique to secure minutiae by generating fictitious minutiae through geometric transformations applied to the coordinates and orientations of real minutiae. By modifying the parameters of the geometric function, new minutiae can be generated to replace compromised or stolen ones. For a more comprehensive study on generating cancelable biometric data, Patel et al. [70] have published an exhaustive research paper.

2.7.3.2. Solutions to Vulnerabilities Related to Biometric System Components

Several studies have been conducted to find solutions for vulnerabilities observed in different components of biometric systems.

- **A. Solutions to Physical Attacks:** These solutions aim to enable biometric systems, particularly sensors, to differentiate between real biometric characteristics and synthetic ones. Many of these solutions leverage vital signs (e.g., temperature, blood pressure, or heartbeat) or physical characteristics (e.g., conductivity, skin elasticity). For instance, Cappelli et al. [71] proposed a method to detect artificial or synthetic fingerprints by exploiting the elasticity of the skin,

specifically the distortions of the finger's epidermis when lifted from the sensor. Matsumoto et al. [72] introduced a technique based on the periodicity of sweat diffusion, while Baldissera et al. [73] suggested using an odor sensor to differentiate human skin odor from the odor of synthetic materials used in creating fake impressions. It's worth noting that most techniques based on vital properties used by biometric systems are publicly accessible, and the secrecy lies in the vital feature employed. However, an impostor who knows the property used by a system can potentially predict and mount a system-specific attack.

- **B. Solutions to Logical Attacks:** Apart from physical attacks, other attacks occur in the digital domain and require malicious techniques and programs. To counter these attacks, various techniques have been proposed:
  - Cryptography: Cryptographic techniques can secure biometric data during transmission or storage, addressing attacks such as eavesdropping (type 2), tampering (type 4), and replay attacks (type 6 and 7). While several cryptographic techniques have been presented, they often require decryption before processing, which introduces vulnerabilities, and managing long keys can be challenging.
  - Digital Watermarking: Digital watermarking, typically used for multimedia data, has been introduced to strengthen the security of biometric systems. It can enhance the security of biometric images during transmission, protect the originality of stored images, detect image alterations, and safeguard templates. Examples include techniques by Pankanti and Yeung [74] and Ratha et al. [75] for fingerprint image protection using digital watermarking techniques.
  - Challenge/Response: Ratha et al. [3] proposed a solution based on the Challenge/Response principle to counter type 2 attacks. The transaction server generates a random challenge sent to the acquisition module, which calculates the corresponding response using the biometric feature sample. The response is then compared with the one received, and any differences indicate a potential type 2 attack.
  - Perceptual Hashing: Perceptual hashing techniques, a novel approach to multimedia data security, have been proposed to protect biometric systems. This approach is discussed in detail in the subsequent chapter.

2.8 Conclusion

In this chapter, the basic concepts of biometric systems, their applications, architecture, and operational modes were presented. Additionally, the evaluation of biometric systems, including their performance, usability, and security aspects, was discussed. The chapter also provided an overview of the main approaches proposed in

the literature to enhance the security of biometric systems. The subsequent chapter will delve into the perceptual hashing approach, which offers a novel method for securing multimedia data.

Chapter 3

An Overview of Perceptual Hashing for Image Fingerprinting

3.1 Introduction

In today's digital era, the digitization of information brings numerous advantages, particularly in terms of data recognition, storage control, and efficient data exchange. This widespread availability of data, coupled with the Internet's prevalence, has resulted in a significant increase in multimedia data, such as images and audios. However, this surge in multimedia data has also given rise to several challenges, including the need to distinguish multimedia data from existing ones, identify similarities among different multimedia files, recognize altered multimedia data, and facilitate accelerated access through indexing. Perceptual hashing has emerged as a promising solution to address these challenges associated with multimedia data. Perceptual hashing techniques enable various applications, such as indexing multimedia databases and identifying multimedia data based on their content. Unlike cryptographic hashing, perceptual hashing focuses on generating hashes that capture perceptual similarities between multimedia files, even if they are not identical.

This chapter provides a comprehensive overview of perceptual hashing, highlighting its fundamental concepts, properties, and diverse applications. Additionally, it presents a summary of the current state-of-the-art approaches employed in the field and the advancements made in perceptual hashing techniques.

3.2 Cryptographic Vs Perceptual hashing Functions

The term "hash functions" has its origins in computer science, where it refers to a function that compresses a string of variable length into a fixed-length string [76]. The data being encoded is often referred to as the message, while the resulting hash value is sometimes called the message digest or simply digest [77]. There are various types of hash functions, but they can generally be categorized into two main groups: cryptographic hash functions and perceptual hash functions. However, it's important to note that the objectives of cryptographic hash functions and perceptual hash func-

tions are not exactly the same. While there may be some overlap in the areas where they are used, perceptual hash functions represent a distinct concept compared to traditional cryptographic hash functions.

- **Cryptographic Hash Functions :** Before delving into the definition of perceptual hash functions, it is essential to understand hash functions in general. Hash functions are specific functions that take variable-sized input data and produce a fixed-sized digital fingerprint, known as a hash, which serves to identify the original data. Cryptographic hash functions, in particular, are designed to yield significantly different hashes even if the input data is slightly modified. This property, known as the avalanche effect, ensures that the behavior of cryptographic hash functions is unpredictable.
- **Perceptual Hash Functions :** In our daily life, there are situations, such as copyright detection or digital forensics, where it is necessary to determine if one digital data has been derived from another. Cryptographic hashes are not suitable for this purpose. Instead, perceptual hash functions are used, specifically tailored for multimedia files. The key advantage of perceptual hashing is that the resulting hash is similar when the data is visually similar. For example, if an image has been retouched, a binary-level comparison would yield a completely different result from the original image. However, a perceptual hash function is designed to change only a few bits when the data is altered, resulting in a similar hash. By employing similarity functions, it becomes possible to determine whether two multimedia data originate from the same source or not.

It is worth noting that both cryptographic hash functions and perceptual hash functions serve similar purposes. They verify data authenticity and ensure data integrity. Cryptographic hash functions are ideal when exact matching of all bit values is crucial, such as in authenticating executable files. However, when authenticating multimedia data, visual content needs to be considered rather than the digital representation alone. Cryptographic hash functions are not well-suited for this task. Perceptual hash functions, on the other hand, are specifically designed to address the security requirements of digital images. Like cryptographic hash functions, perceptual hash functions should generate signatures that possess the following characteristics:

- ***Short*:** The signature should be relatively short, typically a few hundred bits.
- ***Robust*:** Multimedia data with the same visual content should produce the same signature.
- ***Secure:*** It should be practically impossible to reconstruct the original data from its signature, while ensuring that multimedia data with different visual content have completely different signatures.

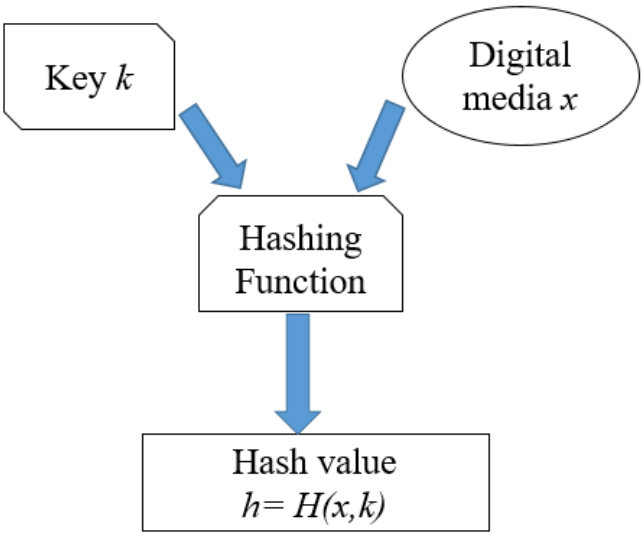


Figure 3.1: Keyed Hashing Function

3.2.1 Definitions and properties of perceptual hashing

To enable efficient multimedia search, it is crucial to have compact and effective representations of the data [78]. Perceptual hashing has emerged as a promising approach to achieve this objective. In general, the term "hashing" refers to the process of compressing data using a hash algorithm. A hash algorithm, denoted as $H(x)$, takes an input string x and transforms it into an output string y, which is commonly referred to as a hash value : $y = H(x)$.
Where x is represented as a sequence of elements $[x_1, ..., x_m]$, each element x_i belongs to the input alphabet X. Similarly, the output string y is represented as a sequence of elements $[y_1, ..., y_n]$, where each element y_i belongs to the output alphabet Y. Mathematically, hashing can be seen as a mapping from the input space X^m to the output space Y^n, where X represents the input alphabet and Y represents the output alphabet [79]. In the domain of multimedia, the input x is typically a real-valued feature vector, while the output y is a binary vector, implying that X is the set of real numbers (R) and Y is the set $\{0,1\}$ [80]. When comparing two hash values, h_1 and h_2, their dissimilarity is commonly measured using several methods, three popular metrics that can be used for this purpose are described in detail in section 3.3.5 bellow. It is worth noting that the choice of which method to use depends on the specific requirements of the application and the desired properties of the similarity/dissimilarity measure. We can distinguish two categories of hash functions [81] :

- **Keyed Hash functions:** generates a hash value h for an input data x and a key k, such as : $h = H(x,k)$, as shown in Figure 3.1

- **Unkeyed hash functions:** generate a hash of value h for an input data x, such as : $h = H(x)$, as depicted in Figure 3.2

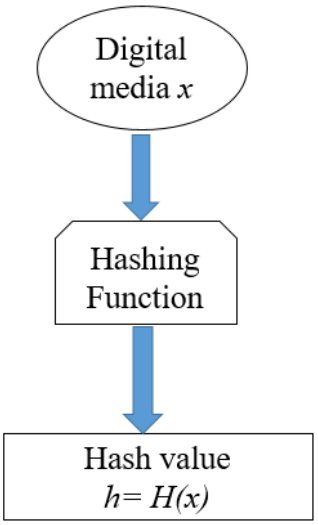


Figure 3.2: Unkeyed Hashing Function

3.2.2 Properties of Perceptual Hashing (PH)

Generally, perceptual hash functions possess four basic properties [82] as follow:

- Unpredictability: The hash value must be as unpredictable as possible, meaning that each bit of the hash should have an equal probability of being 0 or 1. This is achieved by making sure that the hash function combines the different bits of the input image in a complex and unpredictable way.
- Independence: The hash value of one image should be independent of the hash value of any other image. This means that it should be impossible to predict the hash value of one image based on the hash values of other images. This is achieved by making sure that the hash function does not take into account any external information, such as the order in which the images are processed.
- Robustness: The hash values of two perceptually similar images should be nearly identical. This means that the hash function should be robust to small changes in the input image, such as noise, compression, and cropping. This is achieved by extracting features of media objects which are invariant under insignificant content modifications.
- Discrimination: The hash values of two perceptually different images should be very different. This means that the hash function should be able to distinguish between images that are 'perceptually' distinguishable. This is achieved by using features that are sensitive to these types of changes.

In addition to these four properties, perceptual hash functions are also typically required to be compact and secure. The compactness property means that the generated hash value should be short numeric representations of images, which are much smaller in size compared to the original images. The compact image hashes facilitate the search process in a database of hashes and require less storage space [7]. Whereas

the security property of image hashing authentication involves the use of a pseudo-randomization "secret key" making it inherently secure and resistant to unauthorized access [8].

To facilitate the understanding of these properties, we define the following variables:

- H is a perceptual hash function that takes an image as input and generates a hash of length n as output.
- x is an input image.
- $\hat{x}$ is a modified version of x that is perceptually similar to x.
- y is another image that is perceptibly different from x.
- h_x and h_y are the hash values of x and y, respectively.
- $\{0|1\}^n$ represents a binary string of length n.

The four properties of perceptual hash functions can then be defined as follows:

1. Property 1: Unpredictability of the hash values:

$$Pr(H(x) = h_x) \approx \frac{1}{2^n}, \forall h_x \in \{0|1\}^n \tag{3.1}$$

2. Property 2: Independence of hash values for different images:

$$Pr(H(x) = h_x | H(y) = h_y) \approx Pr(H(x) = h_x), \forall h_x, h_y \in \{0|1\}^n \tag{3.2}$$

3. Property 3: Robustness or invariance of hash values for perceptibly similar images:

$$Pr(H(x) = H(\hat{x})) \approx 1 \tag{3.3}$$

4. Property 4: Discrimination or distinction between perceptibly different images:

$$Pr(H(x) = H(y)) \approx 0 \tag{3.4}$$

3.2.3 General framework of a perceptual image hashing system

A perceptual image hashing system typically comprises three key steps, as depicted in Figure 3.3: pre-processing, feature extraction, and post-processing [7, 8].

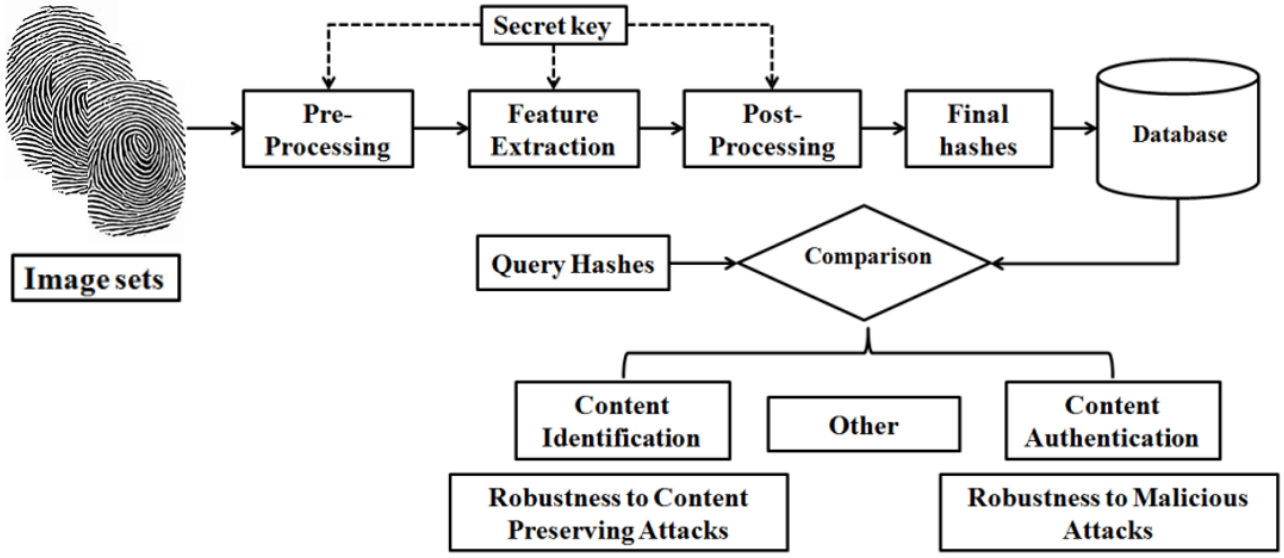


Figure 3.3: General Framework of a PIH system [7, 8]

The strength of an image hashing system lies in its ability to extract robust features, while the post-processing stage primarily contributes to the final hash. To enhance the security of the image hash, a secret key can be incorporated into one or both of these stages. When designing a digital image hashing algorithm, the specific application scenario plays a crucial role in determining the design framework. For instance, if the algorithm is intended for content identification or copy detection, its main focus is on robustness against content-preserving operations (CPOs) that maintain the perceptual quality of the image. Conversely, when the purpose is image authentication, the algorithm should primarily address malicious attacks involving content-changing operations (CCOs), such as object removal or insertion. In this case, the image hashes need to exhibit sensitivity to these perceptually significant attacks.

3.2.3.1 Pre-Processing step

Pre-processing is an essential step in the perceptual image hashing system, aimed at reducing the sensitivity of feature extraction to minor distortions in the image, such as noise and lossy compression. Various common pre-processing operations are typically applied to the images, including [83]:

1. Noise Reduction: Techniques such as Gaussian blurring or median filtering can be employed to reduce noise levels in the image prior to feature extraction.
2. Image Resizing: Resizing the image to a standard or optimal size can help normalize the input and improve the efficiency of subsequent processing steps.
3. Color Space Conversion: Converting the image to a different color space, such as grayscale, can simplify the subsequent feature extraction process and enhance

robustness against color variations.

4. Histogram Equalization: This technique aims to enhance the contrast of the image by redistributing pixel intensities, which can improve the effectiveness of subsequent feature extraction algorithms.
5. Compression Artifact Removal: If the image has undergone lossy compression, pre-processing steps can be employed to mitigate compression artifacts and restore image quality before feature extraction.

It is important to note that the specific pre-processing operations applied may vary depending on the requirements of the perceptual hashing system and the characteristics of the input images.

3.2.3.2 Feature extraction step

In the feature extraction step, the perceptual hashing system extracts the features of the image to generate the intermediate real feature vector . The feature extraction step plays an essential role in perceptual image hashing algorithms, aiming to extract robust and distinctive features that enable content identification. These features should be capable of resisting insignificant content-preserving operations (CPOs) while maintaining perceptual similarity to the human visual system (HVS). Consequently, the resulting hash values should exhibit high similarity for perceptually similar images and dissimilarity for perceptually different images. Robust feature extraction is thus crucial to meet the requirement of robustness.

3.2.3.3 Post-processing step

In the post-processing step of image hashing algorithms, the focus is on achieving compactness by compressing robust features into a fixed-length, real-valued, or binary sequence. This step serves as a dimension reduction process. Several typical techniques used in post-processing are summarized as follows:

1. Quantization: This technique involves converting a continuous feature space into a finite discrete feature space. Popular approaches in image hashing generation include interval quantization, binary quantization using ordinal measures or thresholds [84, 85].
2. Compression and Coding: Techniques used in communication, such as distributed source coding (e.g., Wyner-Ziv, Slepian-Wolf) and error-correcting coding (ECC), can be applied to compress robust features into short image hashes [86].
3. Random Projection: The random projection approach provides performance comparable to conventional dimension reduction methods like Principal Compo-

nent Analysis (PCA). It involves a pseudo-randomization process that enhances the security of the designed image hashing scheme [87, 88].

4. Clustering: This method involves dividing the feature space and mapping similar features into the same centroids of clusters [87].

3.2.3.4 The secret key incorporation

The incorporation of a security key into image hashing is crucial to ensure the unpredictability and confidentiality of the generated image hashes. By introducing a secret key into the hash generation process, image hashes become a pseudo-randomized result that cannot be easily guessed or generated without the correct key. This enhances the security of the image hashing algorithm and allows for controlled access by authorized users while preventing unauthorized access. This aspect is particularly valuable for applications such as copyright protection.
By changing the secret key, the resulting hashes should undergo significant changes, ensuring that even a small modification to the key can lead to drastically different hash values. This property strengthens the resilience of the image hashing system against attacks and unauthorized attempts to generate or manipulate image hashes.

3.2.4 Classification of perceptual image hashing techniques

Most of the existing perceptual image hashing methods mainly focus on the feature extraction step. The main objective for these methods is to extract visual characteristics which remain relatively constant in the face of a specific type of acceptable manipulation. These methods reduce a system of perceptual hashing of images to a system which is satisfied with the extraction of a few characteristics resistant to certain manipulations. These same characteristics will be used in the step of authenticating the image in question [89]. These methods can be roughly classified into five categories [8]:

- Methods based on image statistics.
- Methods based on transform invariant characteristics.
- Methods based on local characteristic points.
- Methods based on dimensionality reduction.
- Learning-based methods.

3.2.4.1 Methods based on image statistics

This type of hash methods extract a hash based on the most stable image statistics and which are invariant to small modifications of the hash image by calculating the image statistics, (visually negligible or not detectable by the human visual system). Statistical features are generally more robust against noise, blurring and compression

distortions [8]. In one of the first approaches, Schneider and Chang [90] proposed to use statistics of intensity, such as the mean and variance from blocks of the image to create an image signature. They observed that these statistics have good robustness properties against small perturbations on the image. The major disadvantage of this method is that it is easy to modify an image without modifying its histogram (for example, the permutation of pixels in a block). It threatens their system security properties. In 2000, Venkatesan et al. [91] developed a perceptual hash function based on a statistics vector extracted from the different subbands of a wavelet decomposition of the image. They observed that such statistics as the means of the subbands of approximations and the variance of the other subbands (of details) remain invariant with respect to a large class of acceptable manipulations. Even though wavelet coefficient statistics have been found to be much more robust than intensity statistics, they do not necessarily reflect content changes as well, especially those that are maliciously generated. Srivastava et al . [92] proposed an image hashing technique for copy detection based mainly on the statistical features of the image. After the preprocessing step, a Radon transformation with DCT is applied to generate a vector. Four statistical features, namely mean, variance, skewness and kurtosis coefficient, which define the central value, dispersion, skewness and peak, were chosen for the hash generation. Huang et al . [93] proposed an image hashing method based on two complementary functionalities, namely the global functionality via the texture and the local functionality via the DCT coefficients. The statistical features in the gray level co-occurrence matrix (GLCM) are extracted to form a global statistical feature, and the leading DCT coefficients of the first row/column of the non-overlapping blocks are extracted to build a feature matrix .

3.2.4.2 Methods based on transform invariant features

This type of method extracts robust features from transformations and then uses the coefficients to generate final hash values. The major advantages of these methods lie in their strong robustness against certain distortions and attacks [8]. In the frequency domain, the coefficients of a transform are important characteristics. The main advantage of these coefficients is their strong robustness against certain distortions and attacks. Currently, various transforms to extract robust features have been used for hash generation, such as Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), Radon Transform , etc. Lin and Chang [94] proposed a DCT block-based perceptual hashing method for image authentication. The method exploits the invariance of the relationship between DCT coefficients selected at the same position in separate blocks of the image to distinguish JPEG compression from malicious operations. The relationship is preserved even when the DCT coefficients are re-quantized. Therefore, the authentication method can distinguish malicious operations from JPEG compression

regardless of the compression ratio and the number of compression iterations. Such a feature is robust to JPEG compression, but is very fragile to other minor modifications. Swaminathan et al. [86] presented a robust hashing technique based on the Fourier-Mellin transformation, exploiting the property of invariance during the affine (2-D) transformation. The proposed method gives conclusive results in terms of robustness against geometric manipulations. Liu and Cheng [95] proposed a perceptual hashing method based on the "Wave Atom" transform. The original image is decomposed into blocky multi-scale coefficients using the "WaveAtom" transform. The perceptual signature is then extracted from the mean and variance of the third scale band block. The third scale band coefficients change slightly when the image content is not visually altered, but the coefficients vary significantly when malicious tampering takes place. Qing et al. [96] proposed a robust image hashing algorithm using non-uniform sampling in the discrete Fourier domain. First, a regularization step is applied to the input image to generate a secondary image . Then, the DFT is applied to have the magnitude matrix of the coefficients. Finally, features are first extracted by non-uniform sampling and then safely quantized into the final binary hash.

3.2.4.3 Methods based on local characteristic points

This category of approaches exploits invariant local features under certain content-preserving image manipulations. The advantages of these hashing methods lie in invariance under geometric transformations, especially against rotation attacks [8]. Feature points are a group of features important for generating robust hashes. These characteristic points generally include edges, corners, spots, protruding regions, etc. Monga et al. [97] proposed an image authentication technique based on feature point extraction, using distortion modeling via an affine transformation. This technique gives good results in the case of geometric attacks. It uses the Hamming distance to measure the similarity between images. Monga and Evans [98] proposed a perceptual hashing system using feature points based on the human visual system (HVS) to ensure the robustness of difficult features like corners and points of strong curvatures. First, they extract feature points using End-stopped wavelet transform, then an iterative procedure is applied to capture feature points with excellent invariance properties against insignificant perturbations. Yan et al . [99] proposed a multi-scale image hashing method using the feature point extraction techniques. First, they produced multiple image content-preserving attacks and then extracted features from the SIFT. These features are matched with the corresponding features extracted from the original image by a matching algorithm. Then, the Round Location-Context Hashing (RLCH) and Angular Location Context Hashing (ALCH) methods are used to build the image hash at multiple scales. Qin et al . [100] proposed a perceptual hashing algorithm with selective sampling for salient structure features. Based on the double-threshold edge detection with the Canny operator, a series of non-overlapping

blocks containing the richest structure information of the perceptual salient feature for the input image are selectively sampled. The two dominant DCT coefficients of the sampled blocks with their corresponding position information are obtained for the construction of the structure characteristics. The hash is generated based on the two extracted feature matrices, i.e. the position information matrix and the dominant DCT coefficient matrix. Next, a compression step is applied to the produced hash to produce the final hash, using the PCA method.

3.2.4.4 Methods based on dimensionality reduction

Dimensionality reduction techniques, such as SVD, Nonnegative Matrix Factorization (NMF), Fast Johnson-Lindenstrauss Transform (FJLT)), Locally Linear Embedding (LLE), Compressed Sensing (CS), are very important to have the essential characteristics which are invariable under many image manipulations and they have been successfully applied in the image hash. Kozat et al . [101] proposed a perceptual hashing scheme by retaining the strongest components of the SVD of the image blocks. Then, by applying an SVD decomposition to pseudo-randomly chosen semi-global regions and employing the singular vectors to extract the robust features (instead of using the singular values of the entire image), they generate the perceptual signature final. Monga and Mihçak [102] proposed a perceptual hashing technique based on NMF. First, NMF is applied to decompose randomly chosen sub-blocks of the initial image. A second NMF is applied to a random vector, followed by dimension reduction. The results obtained demonstrated good robustness to geometric attacks. Lv and Wang [103] proposed an algorithm based on the fast Johnson–Lindenstrauss transformation. This approach based on the coarse features of the image has good robustness properties against most acceptable manipulations. Tang et al . [104] proposed a perceptual hashing system that generates perceptual signatures based on a lexicographic dictionary. The dictionary consists of a large collection of feature vectors called "words" which represent the features of different blocks of the image. The generation of the perceptual signature is calculated from the sub-dictionaries. DCT and NMF are used in their system. The proposed scheme exhibits good robustness against content-preserving manipulations. Sun and Zeng [105] used Fourier-Mellin transform and CS (FMCS) and proposed secure and robust image hashing. The FMCS-based hash was robust to a large class of distortions and geometric attacks. Liu et al . [106] exploited the dimension reduction properties in CS and falsification recovery inherent in the low-rank representation for robust image hashing. The hashing algorithm was robust against content-preserving modifications and had better image retrieval performance.

3.2.4.5 Learning-based methods

In this category of approaches, efficient learning can be integrated into the image hash generation process by leveraging training data. These learning-based perceptual hashing methods use a compact binary representation from the training data. Depending on how the training data is used, these methods can be divided into two categories: semi-supervised and supervised methods [8].

- Semi-supervised methods: Lv et Wang [107] proposed an image hashing algorithm based on semi-supervised spectral integration. The basic principle is to apply a series of virtual attacks, such as additive noise, blurring, compression, etc., on the original images and to generate the corresponding distorted copies. The hashes of the original training images and their distorted copies compose a space, called Virtual Prior Attacked Hash Space (VPAHS). This space represents the distribution of hashes under certain distortions and virtual attacks. The authors proposed that image hashing is an infinite clustering problem, and the hash generated from the original image is not the centroid of the VPAHS space. Therefore, they estimated the exact centroids in VPAHS and incorporated in the training phase for the image hash generation. Using the VPAHS concept and the estimated hash centroids of the training images, they generated a binary hash for each image based on Laplacian spectral integration. Du et al. [108] proposed a Multi-View Semi-supervised Hashing algorithm with Perceptual Saliency (MV-SHPS), which simultaneously explores supervised information and multiple features in learning. Moreover, as falsified images are often obtained by combining multiple images, including adding or removing objects, which are highly relevant to human perception. The authors consider the variation between the original image and the test image as perceptual salience . So, instead of learning the metric distance on the global image, they explored the local hash distance by considering the perceptual salience effect between the different regions.
- Supervised methods: Chen et al. [109] and Qian et al. [110] proposed median filtering detection and steganalysis based on CNN. Bayar and Stamm [111] proposed a universal investigative approach to detect image manipulations, namely median filtering, Gaussian blur, white Gaussian noise, and resampling, using deep learning. The goal of their approach was to build a single deep learning model that could universally identify the presence of these common image processing operations, rather than requiring specialized detectors for each individual manipulation type. The proposed solution detects whether an image had been tampered with using techniques such as median filtering, Gaussian blurring, the addition of white Gaussian noise, or resampling. This universal approach contrasts with having to develop separate detection models for each

type of image manipulation, potentially making it a more efficient and effective method for identifying a range of common digital image forgeries. Choi et al. [112] proposed a composite manipulation detection method based on CNNs. They learned the statistical change of three types of attacks that frequently occur during image tampering, namely Gaussian blur, median filtering and gamma correction, through the proposed CNN architecture for forgery detection. composite.

3.3 Applications of Perceptual Hashing

Perceptual hashing has a wide range of applications, mainly focused on efficient content search, the detection of spam images, indexing of multimedia databases and the detection of illegal images of a pedophile nature on the Internet. Although the application scenarios differ, the following application areas can be distinguished [81, 113]

1. Content identification
2. Integrity check
3. Watermarking support
4. Content-based media retrieval and processing

Apart from these conventional scenarios, perceptual hashing is also finding applications in novel areas. The one-way nature of hash computation makes hash-based approaches suitable for privacy protection in multimedia search. The compactness of hash values also facilitates signal quality estimation with reduced reference. While transmitting hash values requires minimal effort, it is possible to further reduce data transmission by combining perceptual hashing with digital watermarking.
Recently, perceptual hashing has been applied by researchers in various applications related to online abuse detection and analysis [82] . For instance, it has been utilized to identify malicious websites by analyzing webpage screenshots. In addition, perceptual hashing has been employed to detect counterfeit apps by analyzing graphical user interfaces (UIs). More recently, researchers have utilized perceptual hashing to analyze manipulated images in misinformation campaigns, identify and flag child abuse content shared online, and detect identity impersonation in online social networks [114]. Perceptual hashing also plays a significant role in real-world reverse image search engines such as Google, Yandex, Microsoft Bing, and TinEye.

3.3.1 Content Identification:

The primary objective of content identification, also known as near-duplicate detection, is to find similar content despite incidental distortions such as noise addition or geometric distortions. Typically, similarity between multimedia objects is assessed

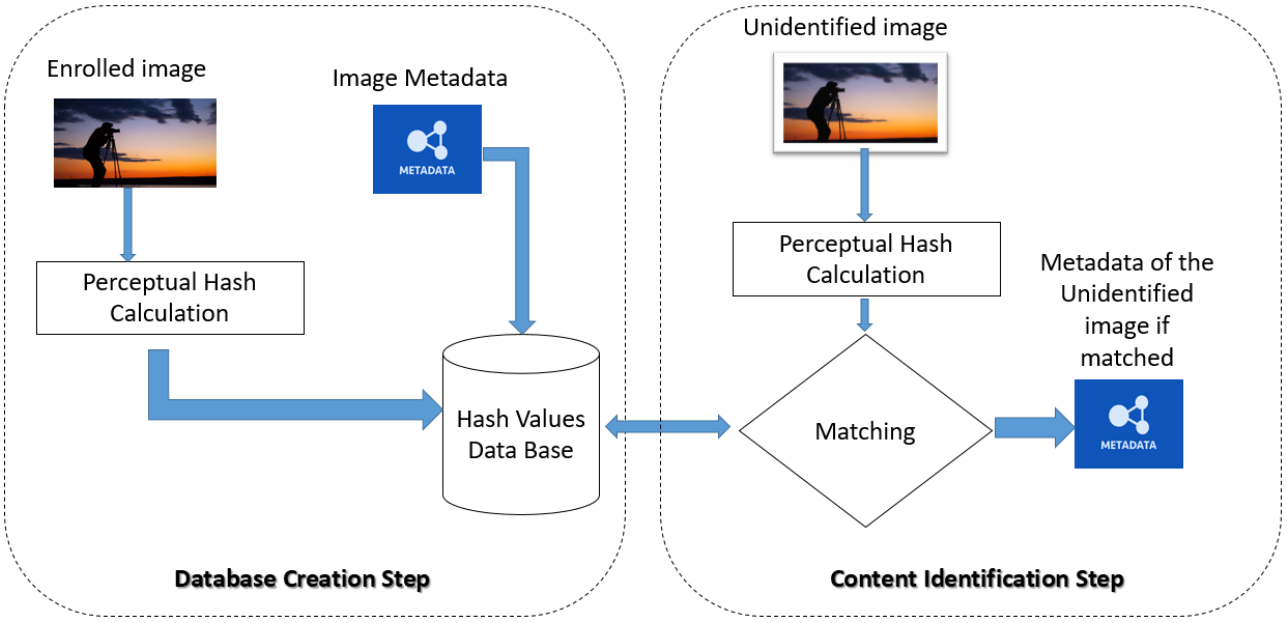


Figure 3.4: The usage of PH in Content Identification.

on a global level. Content identification can be considered a special case of classification with infinite classes [113]. One typical application of perceptual hashing in this domain is images searching among a large database. Rather than storing the whole image, which can be computationally expensive and storage-intensive, perceptual hashing generates a unique and compact hash value for each image. This hash value, along with relevant metadata such as the image name, is then stored in the database. This approach offers several advantages:

1. Significant Reduction in Database Size: By storing hash values instead of the full images, the database size is significantly reduced, enhancing storage efficiency and minimizing computational overhead.
2. Robustness to Image Alterations: Perceptual hashing is designed to be robust against minor image alterations, such as compression, color correction, and brightness adjustments. This ensures that similar images can be identified even after undergoing such modifications.

Figure 3.4 illustrates the usage of PH in the Content Identification, which involves two phases: "database creation" and "content identification." In the first phase, the database is populated with hashes of the images for future identification. In the second phase, when an unknown image is presented to the system, its hash is calculated and compared with the stored hashes in the database. If a match is found, the system retrieves the corresponding metadata information of the identified image. This first application of PH involves the following steps :

1. Pre-processing and feature extraction: In this initial step, the images undergo pre-processing techniques such as resizing, color conversion, and adjusting resolution. These transformations prepare the images to be introduced to a given

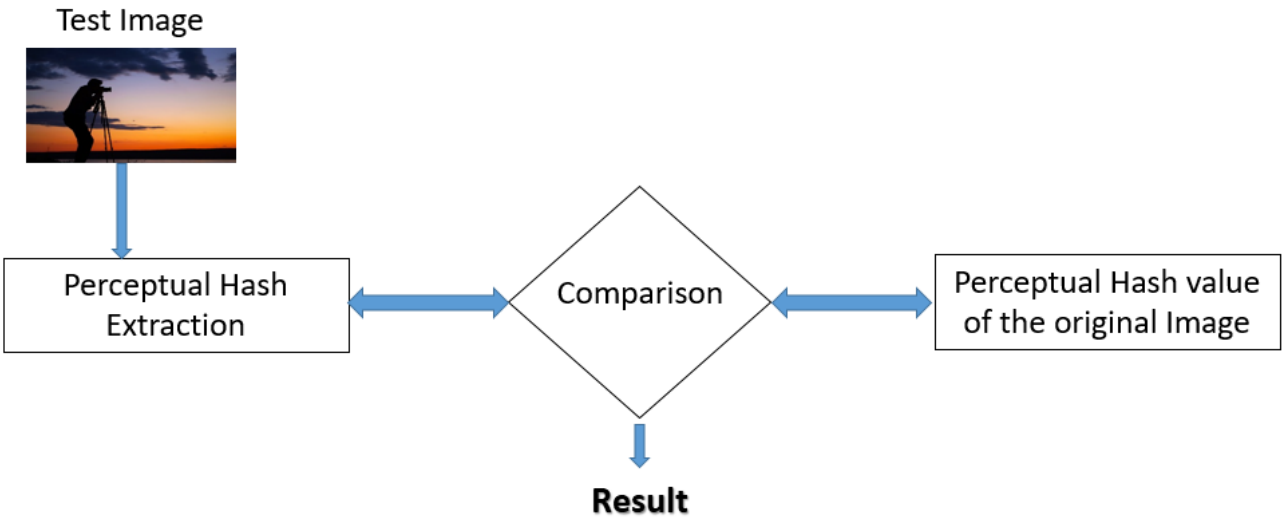


Figure 3.5: The General Principle of Integrity verification using PH

perceptual hash function. Additionally, the necessary features required for the perceptual hash function are extracted from the pre-processed images.

2. Hash modeling: The hash function leverage the extracted features from the previous step to calculate a unique hash value for each image. This hash represents a compact and distinctive fingerprint of each image.
3. Database search: To compare two hashes efficiently, specific search algorithms are employed, along with distance measures or similarity functions. These algorithms facilitate the search process by identifying potential matches within the database.
4. Hash comparison: A comparison function is used to determine the similarity between the computed hash and the hash values stored in the database. This function operates based on the principle of hypothesis testing, where H_0 represents no similarity and H_1 represents the presence of similarity. By utilizing a predefined threshold, the comparison function determines whether the computed hash match with any stored hashes in the database. It is worth noting that the choice of an appropriate threshold is crucial and depends on the specific application scenarios, where it determines the sensitivity and specificity of the hash comparison. For instance, a lower threshold may result in more false positives (matching hashes that are not truly similar), while a higher threshold may result in more false negatives (missing true matches). So, the optimal threshold should be selected based on the requirements of the application, balancing the trade-off between false positives and false negatives.

3.3.2 Integrity check

Perceptual hash functions are also used to ensure the integrity of an image, by following the steps depicted in Figure 3.5. Based on this principle, perceptual hash

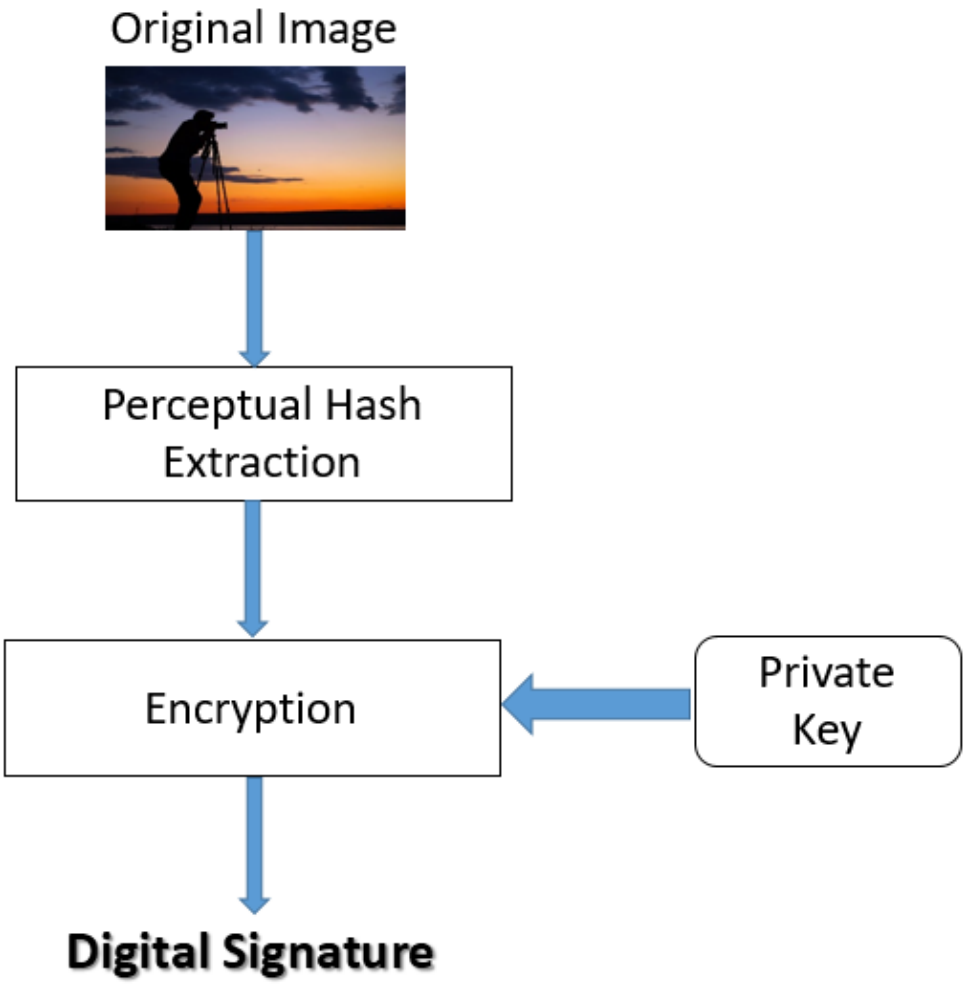


Figure 3.6: Creation of Digital Signature process

functions can be employed in two distinct ways. Firstly, the hashes can be directly inserted into the images using digital watermarking techniques. Secondly, the perceptual hash itself can be signed using a digital signature. The process of creating a digital signature is illustrated in Figure 3.6, while Figure 3.7 demonstrates its verification. Furthermore, a few hash functions are able to determine the nature and location of the modifications introduced to the original images.

3.3.3 Watermarking Support

Perceptual hash functions can efficiently support watermarking techniques, specifically through the creation of semi-fragile signatures [115]. These signatures involve embedding perceptual hashes into media objects using robust or semi-fragile watermarking methods. Figure 3.8 and Figure 3.9 illustrate this process. Semi-fragile watermarks are designed to authenticate only the image features they are embedded within. For example, in image watermarking, the watermark information is often embedded in the high-frequency coefficients of the block discrete cosine transform (DCT), as changes in these coefficients are less perceptible to the human eye compared to low-frequency coefficients [81]. This ensures that only modifications in the high-frequency coefficients can be detected by the watermark. However, these limi-

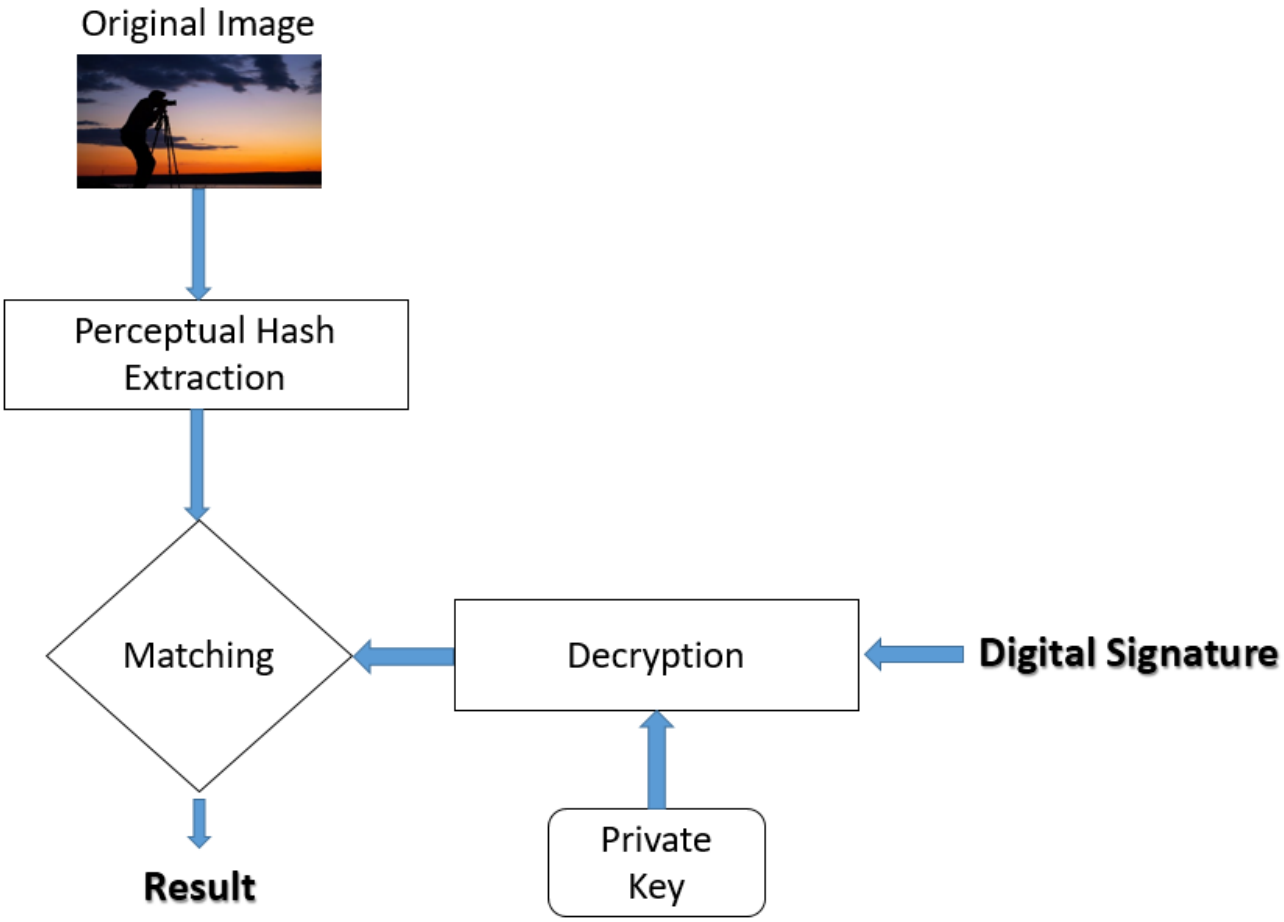


Figure 3.7: The Digital Signature checking process

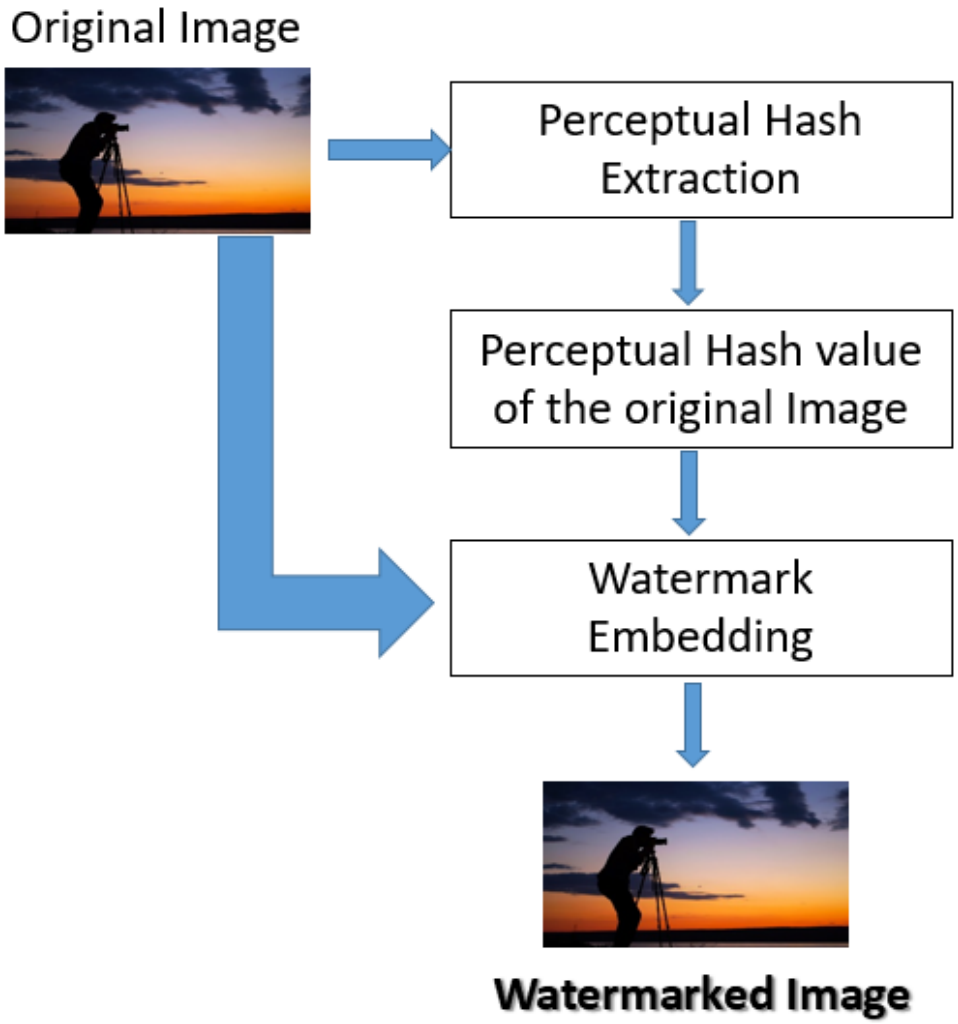


Figure 3.8: Watermark support: Perceptual hash watermark Embedding

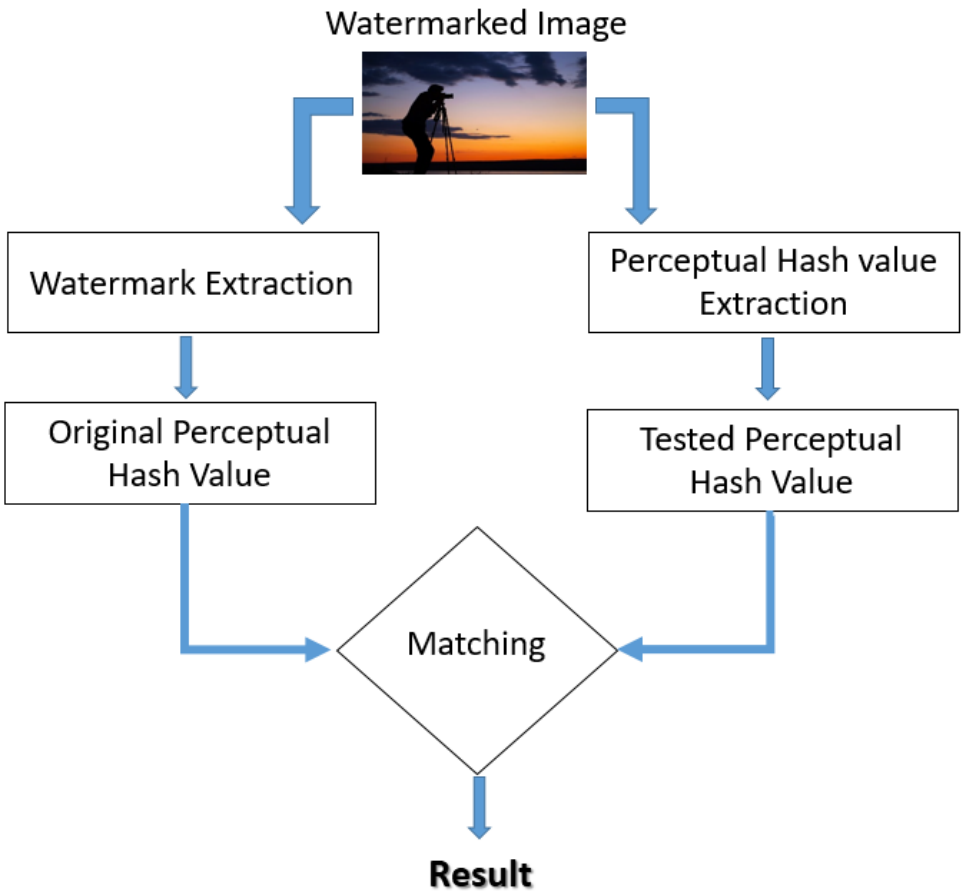


Figure 3.9: Watermark support: Perceptual hash watermark extraction

tations can be overcome by employing semi-fragile signatures, where a suitable robust or semi-fragile watermarking algorithm can embed any desired perceptual hash, allowing authentication of any image feature considered by the perceptual hashing algorithm. Therefore, in this way, perceptual hash functions enhance the capabilities of watermarking techniques.

3.3.4 Content-Based Media Retrieval and Processing

Content-Based Image Retrieval (CBIR) is a technique used to retrieve images from a database based on their content [116]. It involves selecting images that match a specific query, such as extracting road environment images. CBIR methods rely on extracting significant features from images, which can be either local or global in nature. These techniques draw inspiration from fields like statistics, pattern recognition, signal processing, and computer vision. In CBIR, statistical measures of similarity between image representations are used to correlate with subjective visual similarity. Perceptual hashing, when applied in the context of CBIR, involves several steps. These include indexing the images based on their content, extracting relevant features, compiling the extracted information, and performing searches based on user queries. The goal is to create a perceptual hash that captures the unique characteristics of an image and allows for efficient retrieval of similar or matching images from the database.

3.3.5 Similarity Metrics for Perceptual Hashes

When evaluating the performance and robustness of image hashing techniques we need to use distance or similarity metrics in order to calculate the perceptual differences between similar/dissimilar media objects. This is particularly important in image hashing, as it enables us to determine whether two given images are identical or completely different, especially when the generated hash for an image needs to be verified among a database. Alternatively, the hashes should reflect how "perceptually similar" the images are to each other. In image hashing, when a query hash is received, the similarity between hashes is determined by calculating the distance. This distance calculation results in a "similarity score," which is used to make decisions. The choice of distance metrics to measure the similarity between hashes is crucial in image hashing[7].
In the next sections, three popular metrics that can be used for this purpose are described in detail. Lets define two hash values $H_1(I_1)$ and $H_2(I_2)$ as described in Equations 3.5 and 3.6 below:

$$H_1(I_1) = h_1(1), h_1(2), ..., h_1(L) \tag{3.5}$$

$$H_2(I_2) = h_2(1), h_2(2), ..., h_2(L) \tag{3.6}$$

Where H_1 and H_2, represent the hashes of two distinct images I_1 and I_2 with a specific hash length L. These two hashes are represented as an array of bits denoted as h. The choice of a specific metric of distance calculation for comparing the similarity between the hashes depends on whether they are binary or decimal. Different approaches can be employed accordingly to calculate the distance [7, 81].

3.3.5.1 Hamming Distance

The Hamming distance (HD) is a metric used to calculate the similarity between two binary hash vectors by comparing them bit-by-bit and counting the differing bits. It represents the minimum number of substitutions needed to transform one string into another [117]. For instance, if we have two strings X and Y, the HD can be calculated using Equation 3.7:

$$HD = \sum |X_i - Y_i| \tag{3.7}$$

It is worth noting that a higher value of HB signifies more differences between the compared strings. Specifically, in our case, where the strings are hashes, a HD that equal to 0 means that the hashes are identical. As the Hamming distance increases, the differences between the two hashes become more pronounced, indicating a greater "distance" between them. However, it's important to note that the Hamming distance cannot exceed the length of the string being compared.
In some cases, when the strings can consist of elements from alphabets or other number systems, then the HD can be calculated based on the number of characters rather

X	Y	HD
1000101	1000101	0
1100**1**10	1100**0**10	1
2**4**58**7**	2**5**59**7**	2
M**a**a**m**a**r**	M**a**e**m**e**n**	**3**

Table 3.1: Examples of calculating the Hamming distance (HB) of two given strings X and Y having different systems (binary, Latin alphabet and decimal system).

than the number of bits. Table 3.1 provides examples illustrating the calculation of the Hamming distance using different systems, including binary, Latin alphabet, and decimal systems. Indeed, while Table 3.1 demonstrates that alternative systems such as the decimal system or alphabets can be employed, it is important to note that for our specific purpose of calculating the Hamming distance of fingerprint images, we only need to consider the binary representation of the data. By utilizing the XOR operator($\oplus$), we can denote H_1 and H_2 as two binary-coded numbers of equal length L. The Hamming distance can then be determined by counting the number of ones in the result of the XOR operation between H_1 and H_2. Expanding upon and adjusting the basic Equation 3.7, we arrive at the comprehensive equation for calculating the HD between two binary hashes as follows :

$$HD(H_1,H_2) = \sum_{i=1}^{L} |h_1(i) \oplus h_2(i)| \tag{3.8}$$

In Equation 3.8, the Hamming Distance (HD) represents the calculated distance between the two binary hashes H_1 and H_2. The XOR operation $(H_1 \oplus H_2)$ compares the corresponding bits in H_1 and H_2, and the sum is taken over all bits denoted as L in their binary representation. By summing the resulting operations, we obtain the total number of differing bits, which precisely quantifies the Hamming distance between the two binary hashes.

3.3.5.2 Normalized Hamming distance

The Hamming distance can be normalized by considering the length $'L'$ of the strings being compared. This normalization is particularly useful when we want to obtain the ratio of differing elements. By adapting Equation3.8 for the Hamming distance, we can derive the equation for its normalized version (Equation 3.9).

$$NHD(H_1,H_2) = \frac{1}{L}\sum_{i=1}^{L} |h_1(i) \oplus h_2(i)| \tag{3.9}$$

Where NHD represents the Normalized Hamming Distance, h_1 and h_2 are the binary hashes of two different images having the same length L. Also, $h_1(i)$ refers to the i^{th} bit of the first binary hash H_1, and $h_2(i)$ denotes the i^{th} bit of the second binary hash H_2. It is important to note that when calculating the NHD, the two compared hashes

H1	H2	NHD
1000101	1000101	0
1100**1**10	1100**0**10	0.14
1**1**10**1**1**0**	1**0**10**0**1**1**	0.42
1**0**1**0**1111	1**1**0**1**1011	0.5

Table 3.2: Examples of calculating the Normalized Hamming distance (NHD) of two given binary hashes H_1 and H_2.

must have the same length. Table 3.2 provides examples of calculating the Normalized Hamming distance specifically for the binary system. As shown in Table 3.2, the NHD value is expected to be close to 0 for similar images on the one hand, and close to 0.5 for dissimilar ones, on the other hand. As modifications could be done on different parts of an image, the manipulated image and the original image gradually become more dissimilar and divergent from each other. In an ideal hashing scheme, the normalized Hamming distance between the corresponding hashes should reflect this increasing divergence.

3.3.5.3 Bit Error Rate

Bit Error Rate (BER) is another commonly used distance metric, particularly in the context of binary systems. The BER can be represented by the following equation (Equation 2.6):

$$\rho = \frac{i}{L} \tag{3.10}$$

Where, ρ represents the BER and i is the number of bit errors in the hash, normalized by the hash length L. The BER should fall within the range of $0 \leq \rho \leq 1$ and $i \in \{0, 1, 2, ..., L\}$. Table 3.3 provides examples of calculating the BER for two given binary hashes. As depicted in Table 3.3, the number of bit errors i is equivalent to the HD between two different hashes, and the resulting value ρ is equal to the normalized Hamming distance.

H1	H2	BER
1000101	1000101	0
1100**1**10	1100**0**10	0.14
1**1**10**1**1**0**	1**0**10**0**1**1**	0.42
00101101	**11010010**	1

Table 3.3: Examples of calculating the Bit Error Rate (BER) of two given binary hashes H_1 and H_2.

In summary, the bit error rate (BER) quantifies the number of bit errors in a hash relative to its length. It provides a measure of similarity between two hashes, with the resulting value being equivalent to the normalized Hamming distance.

3.4 Conclusion

This chapter has presented a comprehensive overview of perceptual hashing techniques, covering various aspects such as their definition, properties, main framework, applications, and similarity metrics. By gaining a solid understanding of the concept and application of perceptual hashing, we can harness these techniques to significantly enhance the security, integrity, and efficiency of a wide range of applications, including biometric systems, multimedia forensics, and copyright protection. The utilization of perceptual hashing holds great potential in addressing critical challenges faced by these applications. For instance, in biometric systems, perceptual hashing can help enhancing security measures by mitigating the risk of digital replay attacks. By implementing advanced perceptual hashing algorithms and exploring their practical implementations, we can protect finger-print based biometric systems against one of the most devastating attacks, thereby ensuring the reliability and authenticity of biometric data.

In the next chapters, we will delve deeper into specific case studies, delve into advanced perceptual hashing algorithms, and explore their practical implementations. We will further explore the potential of perceptual hashing in securing biometric systems against digital replay attacks, thereby contributing to the advancement of secure and reliable authentication mechanisms.

In conclusion, the comprehensive exploration of perceptual hashing techniques in this chapter highlights their significance and applicability in various domains. In the subsequent chapters, we will further explore the potential of perceptual hashing techniques in order to strengthen security measures and provide enhanced protection against digital attacks.

Chapter 4

A Comparative Evaluation of Perceptual Image Hashing Approaches

4.1 Introduction

In this chapter, we extensively analyze and evaluate the performance of five commonly used perceptual hashing techniques when applied to fingerprint images, specifically focusing on their ability to verify image authenticity. The evaluated techniques encompass both spatial and frequency domain approaches. In the spatial domain, we assess the effectiveness of the Average hashing (A-Hash) and the Difference hashing (D-Hash) techniques. These methods operate by analyzing the pixel values and computing average and difference values, respectively, to generate perceptual hashes. In the frequency domain, we investigate three techniques: DCT-based hashing (P-Hash), Wavelet-based hashing (W-Hash), and SVD-based hashing (SVD-Hash). These methods employ transforms such as Discrete Cosine Transform (DCT), Wavelet Transform, and Singular Value Decomposition (SVD) to extract frequency-based features and generate perceptual hashes. To conduct our evaluation, we perform extensive experiments using real fingerprint images. Our assessment focuses on three key aspects: perceptual robustness, discrimination capability, and authentication properties of each evaluated technique. We aim to understand how well these techniques handle the unique characteristics of fingerprint images, which are often perceived by non-experts as alternating curved dark lines (representing ridges) and white lines (representing valleys) that appear visually similar.

By thoroughly analyzing the performance of these perceptual hashing techniques, we aim to shed light on their efficacy in differentiating fingerprint images and determining their authenticity. The obtained results will provide valuable insights into the strengths and limitations of each technique, assisting in the selection and implementation of appropriate perceptual hashing methods for fingerprint image verification.

4.2 The compared PIH Techniques

In this section, we detail five of the widely used perceptual hashing techniques. It is important to note that the emphasize here is specifically on perceptual image hashing algorithms that offer fast computation of image hashes while preserving the image identity. These perceptual hashing techniques have been carefully selected to reach a balance between computational efficiency and the ability to capture key visual characteristics of images. By generating perceptual hashes that encapsulate important visual features, these algorithms enable quick and reliable identification and comparison of images. Also, we evaluate the performance of these perceptual hashing techniques in terms of their speed and accuracy in distinguishing different images.

By analyzing and assessing these perceptual hashing techniques within the context of fingerprint images, we aim to provide valuable insights into the trade-offs between computational efficiency and the preservation of image integrity capability. This knowledge will assist researchers and practitioners in selecting perceptual hashing algorithms that best suit their specific requirements, seecking for an optimal balance between speed and image identity preservation.

4.2.1 A-Hash

The A-Hash technique, known as an average hashing technique, is a simple and fundamental method used for generating perceptual hashes of images [118–120]. It involves extracting the image's low frequencies, which represent the overall structure, while eliminating the higher frequencies that correspond to image details [121]. This technique employs several pre-processing operations, including blurring, resizing, color reduction, and normalization [119]. The primary objective of A-Hash is to determine the average color of all the image matrix values by calculating the mean of the matrix. For the implementation of this technique, we followed the steps outlined in the work of Popkov [119]. The steps can be summarized as follows:

1. The input image is resized to a dimension of 8x8 pixels.
2. A color space conversion is performed, transforming the image from the RGB color space to the gray-scale (YCbCr) color space.
3. The mean value of all luminance values in the resulting image matrix is calculated.
4. Each element of the image matrix is compared with the calculated mean value, resulting in a new binary matrix with 64 elements. A value of 1 indicates that the intensity of the element is greater than the average, while a value of 0 indicates otherwise.
5. The resulting binary matrix is used to construct a 64-bit long hash. This is achieved by creating a vector from the binary matrix, starting from the top left

and moving towards the bottom right.

The resulting hash can be compared with the hashes of other images, allowing for the calculation of a "similarity score" based on the distance metric between the two hashes. Figure 4.1 below illustrates the block diagram of the A-Hash technique.

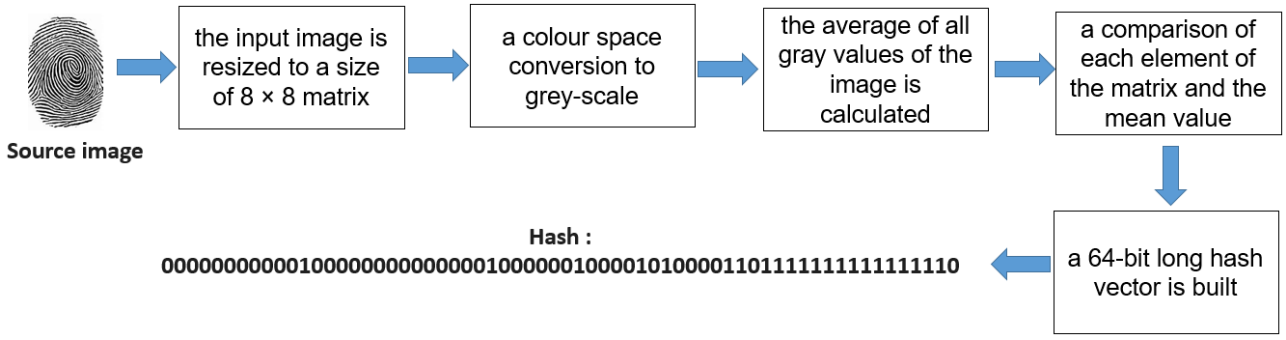


Figure 4.1: Block diagram of the A-Hash algorithm

It is worth noting that the A-Hash technique serves as a basic method for generating perceptual hashes and can be further expanded upon and optimized in more advanced hashing algorithms.

4.2.2 D-Hash

The D-Hash technique, also known as the difference hash technique, is an alternative method similar to A-Hash [122]. Like A-Hash, D-Hash focuses on capturing the image structure by reducing the image size and eliminating higher frequencies . However, the key distinction between the two techniques lies in how the hashes are generated. D-Hash computes the difference hash similarity of the image by analyzing the color gradient changes between adjacent pixels in the image matrix [123, 124]. To implement the D-Hash algorithm, we also followed the steps outlined in the research of Popkov et al., 2015 [119]. These steps can be summarized as follows:

1. The image is reduced to a block size of 9x8, resulting in a total of 72 pixels.
2. The image is converted to grayscale.
3. For each row, the differences between adjacent pixels are computed, resulting in a total of 8 differences per row.
4. The 64 differences obtained from the image are used to construct the image hash. If the difference value is negative, the corresponding hash bit is set to 0; otherwise, it is set to 1. This process generates a 64-bit hash.

Figure 4.2 below illustrates the overall diagram of the D-Hash technique.

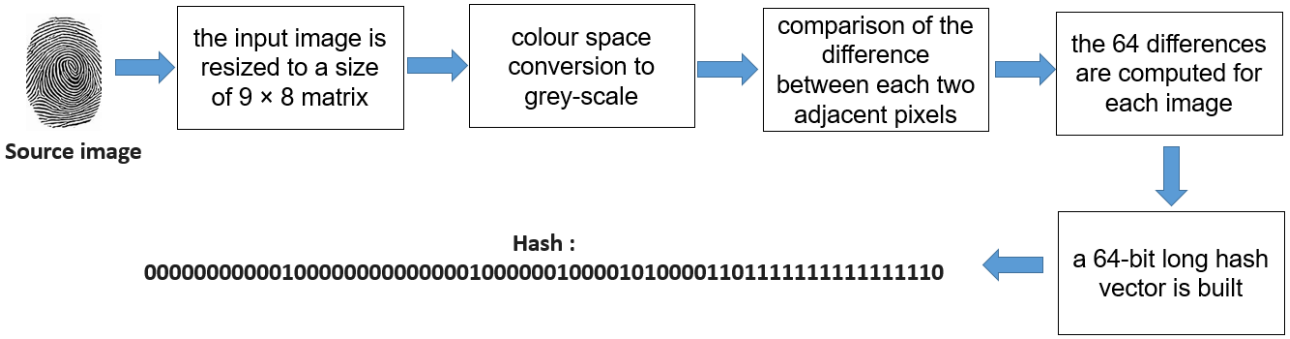


Figure 4.2: Block diagram of the D-Hash algorithm

It is important to note that while D-Hash shares similarities with A-Hash in terms of focusing on image structure, it incorporates a distinct approach by utilizing color gradient changes. The D-Hash algorithm provides an alternative method for generating perceptual hashes and can be further refined and optimized in advanced hashing algorithms.

4.2.3 P-Hash

The P-Hash technique, which is short for perceptual hash, builds upon the A-Hash method by incorporating the Discrete Cosine Transform (DCT) to capture the most sensitive information from the human visual system (HVS) [120]. Similar to A-Hash, P-Hash follows the approach of computing mean values and performing comparisons [119]. However, instead of utilizing image intensities for the hash generation process, P-Hash utilizes a range of low frequencies obtained through the application of the DCT technique [125]. The implementation of P-Hash involves the following steps:

1. The image is resized to a matrix of dimensions 32x32 pixels.
2. The resulting image is then converted to grayscale.
3. A 32x32 DCT is performed on the grayscale image, resulting in a DCT coefficient matrix of the same dimensions. This transformation gathers the energy of the image into a few low-frequency DCT coefficients.
4. A vector of length 64 is constructed by concatenating the DCT coefficients starting from position (1,1), which corresponds to the upper left corner of the 64x64 matrix, and ending at position (8,8), representing the lower right corner.
5. The mean value of the resulting array of DCT coefficients is computed.
6. A comparison is made between each of the 64 DCT coefficients and the mean value. If a coefficient is greater than the mean, the corresponding hash bit is set to 1; otherwise, it is set to 0.
7. Finally, a 64-bit binary hash is obtained.

Figure 4.3 illustrates the general diagram of the P-Hash technique.

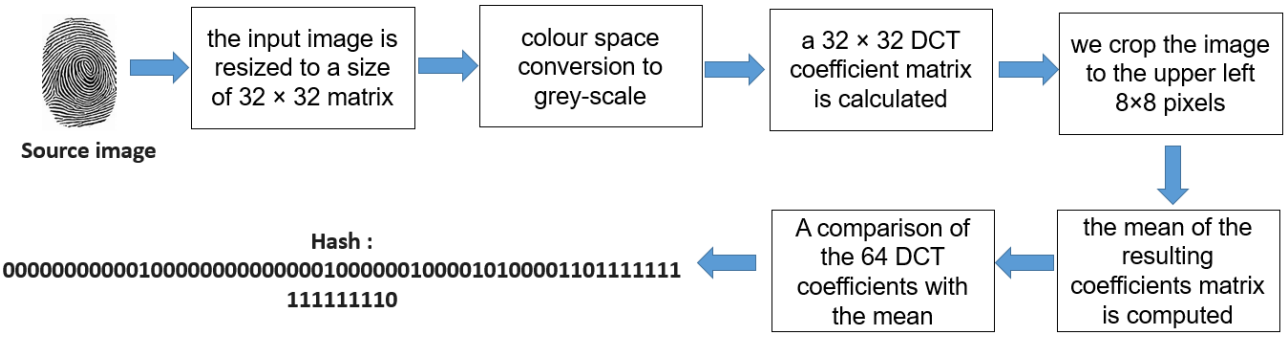


Figure 4.3: Block diagram of the P-Hash algorithm

It should be emphasized that P-Hash extends the A-Hash method by incorporating the DCT to capture low-frequency components that are more sensitive to the human visual system. By integrating these components into the hash generation process, P-Hash enhances the capabilities of perceptual hashing. The P-Hash algorithm provides an effective technique for generating perceptual hashes and can be further refined and optimized in advanced hashing methods.

4.2.4 W-Hash

The W-Hash technique, known as the wavelet hash technique, is a frequency domain hashing method that utilizes the Discrete Wavelet Transform (DWT) to generate perceptual hashes. It operates by analyzing the image in the wavelet domain while preserving temporal information [126]. The DWT is commonly employed to eliminate redundancy in data with highly correlated neighboring values, such as pixels in images [121]. The original W-Hash was introduced by Venkatesan et al. [91], who outlined the main steps for implementing this technique as follows:

1. Perform a wavelet decomposition of the image using the Haar wavelet, dividing it into sub-bands at multiple levels [127].
2. Calculate a randomly tilled statistics vector for each sub-band.
3. Quantize the resulting statistics vector using a randomized quantizer.
4. Decode the quantized statistics vector using a first-order Reed-Muller error-correcting decoder, producing a binary hash value of length n.
5. Pass the intermediate hash value through another decoding stage of a linear code with random parameters to transform it into an even shorter hash code.

Figure 4.4 illustrates the block diagram of the W-Hash technique.

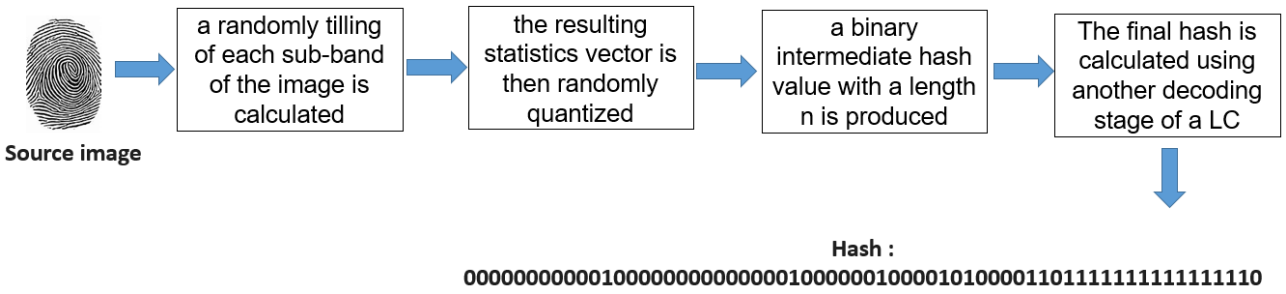


Figure 4.4: Block diagram of the W-Hash algorithm

It is noteworthy that W-Hash utilizes the wavelet domain and incorporates various encoding and decoding stages to generate the final hash code. This technique offers a robust approach for perceptual hashing by considering the wavelet coefficients and their statistical properties. Further research and optimization can be carried out to improve the performance of W-Hash in different applications.

4.2.5 SVD-Hash

The SVD-Hash technique, also known as Singular Value Decomposition hash, was initially proposed by Kozat et al. [101] as a robust hashing method. This technique involves deriving a secondary image from the original image by pseudo-randomly extracting features that capture semi-global geometric characteristics. The final features are then extracted and appropriately quantized to form the hash value. The implementation steps of the SVD-Hash algorithm are summarized as follows:

1. From the input image matrix of size $n \times n$, form p possibly overlapping blocks, so that each of them has the size of $m \times m$.
2. For each resulting block, generate the corresponding feature vector using the Singular Value Decomposition (SVD) transformation.
3. Generate a secondary image by combining all intermediate feature vectors using a pseudo-random (PR) technique.
4. Apply the same steps as 1 and 2 to the new resulting image.
5. Finally, combine the generated feature vectors from the second image to construct the final hash vector.

Figure 4.5 illustrates the general diagram of the SVD-Hash algorithm.

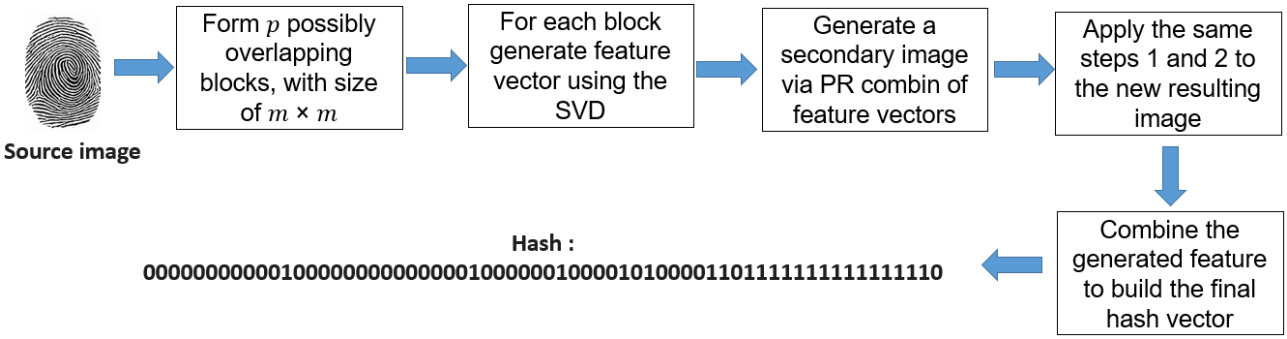


Figure 4.5: Block diagram of the SVD-Hash algorithm

The SVD-Hash technique utilizes the SVD transformation to extract relevant feature vectors from blocks of the image. These feature vectors are then combined to form a secondary image, which undergoes the same process to generate the final hash vector. SVD-Hash provides a robust approach for generating perceptual hashes by considering semi-global geometric characteristics. Further research and optimization can be conducted to enhance the performance of SVD-Hash in various applications.

4.3 Experimental Results

In this section, intensive experiments have been carried out to evaluate the performance of the techniques described in Section 4.2. In all experiments, real fingerprint images from the 'FVC 2000, DB3_a' database [128] have been used. This database contains 800 fingerprint images of size 448×478. Note that these images have been slightly resized to have a size of 448×480 (i.e., having a height and width divisible by 8). Three aspects are considered in our experiments: (i) perceptual robustness, (ii) discrimination capability, and (iii) authentication. Moreover, and in order to make the comparison as fair as possible between the evaluated techniques, the Normalized Hamming distance has been used as a metric to evaluate the similarity between extracted hashes. To perform perceptual robustness and authentication testing, eleven kinds of commonly used content-preserving image manipulations were utilized to produce visually similar images. The details of image manipulations and the corresponding parameter settings are given in Table 4.1.

4.3.1 Perceptual Robustness

An efficient perceptual hashing technique should be robust and resist content-preserving manipulations with moderate strength. In other words, for visually similar images, it should produce the same or similar hashes even their digital representations are no longer the same. This characteristic can be measured by evaluating the perceptual robustness. In this work, we conducted a set of experiments to assess the perceptual robustness of the five evaluated techniques. Hence, for every evaluated technique, we extracted the original (reference) hashes from the original 800 fingerprint images and their corresponding manipulated versions under the eleven content-preserving

Table 4.1: Content-preserving manipulations and parameters setting

Manipulation	Parameters setting
Gaussian noise (GN)	variance = 0.07
Average filtering (AF)	filter size = 9
Salt & pepper noise (SP)	density = 0.20
Gaussian blurring (GB)	standard deviation = 7
Gamma correction (GC)	gamma = 0.8
Motion blurring (MB)	len = 40, theta = 55
JPEG compression (JC)	quality factor = 40
Median filtering (MF)	filter size = 7
Wiener filtering (WF)	filter size = 7
Image sharpening (SH)	alpha = 0.49
Image scaling (SC)	factor = 0.5

manipulations listed in Table 4.1. Then, the Normalized Hamming distance is calculated between each original hash and its corresponding manipulated hash. The minimum, maximum, and mean values of the resulting distances after each operation are presented in Table 4.2. Note that, to make the comparison as fair as possible, we have normalized the obtained results by mapping them to the same interval. The obtained results clearly show that, in most cases, the A-Hash technique provides the lowest mean values of the Normalized Hamming distances, computed between reference hashes and their manipulated ones. The P-Hash comes in the second place in most cases. Moreover, the D-Hash shows an acceptable level of robustness against the most manipulations. The least robust technique is the SVD-Hash, where the results clearly show that the applied manipulations affects the Hamming distances significantly. It is worth mentioning that in the absence of manipulations, the Normalized Hamming distances should be equal to 0.

Table 4.2: Hamming distances under different content-preserving manipulations. Best values are in bold.

	A-Hash			D-Hash			P-Hash			W-Hash			SVD-Hash		
	Min	**Max**	**Mean**	**Min**	**Max**	**Mean**	**Min**	**Max**	**Mean**	**Min**	**Max**	**Mean**	**Min**	**Max**	**Mean**
GN	0.000	0.214	**0.035**	0.000	0.375	0.135	0.000	0.273	0.110	0.043	0.385	0.138	0.037	0.567	0.134
AF	0.000	0.156	0.018	0.000	0.119	0.019	0.000	0.097	0.025	0.006	0.035	**0.015**	0.010	0.200	0.058
SP	0.000	0.117	**0.013**	0.000	0.217	0.041	0.000	0.136	0.033	0.022	0.165	0.055	0.021	0.338	0.066
GB	0.000	0.097	**0.008**	0.000	0.127	0.079	0.000	0.097	0.012	0.000	0.029	0.010	0.012	0.214	0.070
GC	0.000	0.097	**0.011**	0.000	0.098	0.016	0.000	0.117	0.020	0.024	0.159	0.071	0.003	0.163	0.032
MB	0.000	0.136	**0.020**	0.000	0.177	0.046	0.000	0.195	0.064	0.000	0.035	0.014	0.014	0.221	0.073
JC	0.000	0.078	**0.001**	0.000	0.039	0.003	0.000	0.039	0.005	0.000	0.010	0.002	0.001	0.012	0.003
MF	0.000	0.253	**0.036**	0.000	0.217	0.048	0.000	0.195	0.055	0.011	0.189	0.043	0.014	0.329	0.066
WF	0.000	0.078	**0.007**	0.000	0.138	0.015	0.000	0.078	0.009	0.000	0.043	0.015	0.009	0.106	0.030
SH	0.000	0.078	0.007	0.000	0.079	0.007	0.000	0.058	**0.003**	0.000	0.030	0.008	0.006	0.083	0.015
SC	0.000	0.078	**0.007**	0.000	0.059	0.010	0.000	0.058	0.008	0.000	0.011	0.001	0.026	0.944	0.235

4.3.2 Discrimination capability

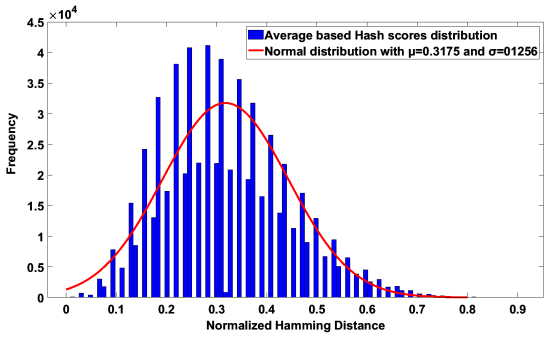
The discrimination capability of a hashing algorithm, also known as anti-collision capability [100], can be defined as its ability to generate significantly different hashes for visually distinct images. This means that an algorithm with high discrimination

capability has a very low probability to generate similar hashes for two perceptually different images [129]. In general, the discrimination capability is evaluated by computing the collision probability of two hashes for two visually distinct images, which is in our case equals to the probability that the Normalized Hamming distance is smaller than the predetermined threshold. According to [100], the collision probability P_c of the hashes for two visually distinct images is given by:

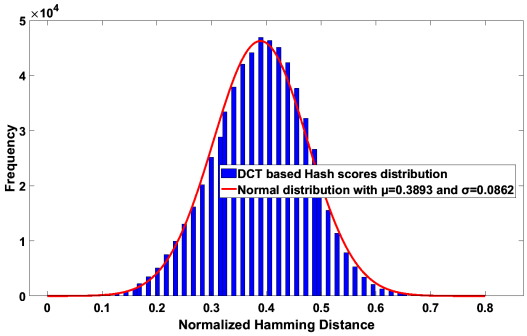
$$\begin{aligned} P_c(T) &= \frac{1}{\sqrt{2}\delta}\int_{-\infty}^{T} exp\left[-\frac{(x-\mu)^2}{2\delta^2}\right]dx \\ &= \frac{1}{2}erfc\left(-\frac{T-\mu}{\sqrt{2}\delta}\right) \end{aligned} \tag{4.1}$$

where $erfc(.)$ is the complementary error function, μ is the mean value, δ is the standard deviation and T is the predetermined threshold. Note that the values of T have been empirically calculated as described in [129]. In order to compute P_c, we first need to estimate the parameters μ and δ corresponding to the NHD values computed from a large set of visually distinct images. To do so, and for each evaluated technique, we first extracted the reference hashes for all the 800 fingerprint images and then calculated the NHD for each hash with the other 799 hashes. Consequently, we obtained $800 \times (800-1)/2 = 319600$ NHDs. Then, the values of μ and δ of the obtained NHDs are computed by assuming that they all follow a normal distribution. The distribution of these 319 600 distances between hashing pairs, along with the obtained values of μ and δ, for the five evaluated techniques are shown in Fig. 4.6, where the abscissa is the Normalized Hamming distances and the ordinate represents their frequency. Note that this assumption is widely adopted in the literature [100, 129, 130]. As can be noticed, only the D-Hash and the P-Hash algorithms generate Normalized Hamming distances that follow a normal distribution. The other algorithms are not really approximated by this distribution, and this fact, may lead to a less accurate evaluation of the discrimination capability by the collision probability measure.

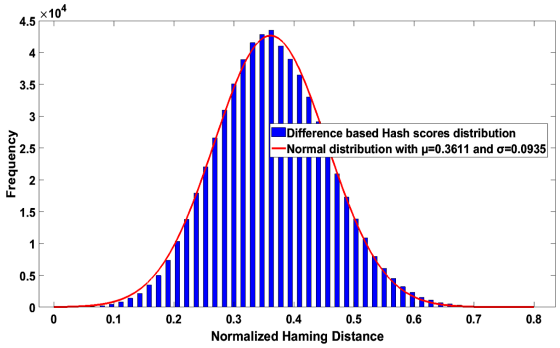
After obtaining the parameters μ and δ, we applied Eq. 4.1 to compute the collision probability P_c for different values of thresholds T. The obtained results are given in Table 4.3. From these results, one can observe that, the smaller the threshold T is set, the smaller the collision probability is. Furthermore, the collision probability P_c values are close to each other, with a slight superiority of the P-Hash technique.



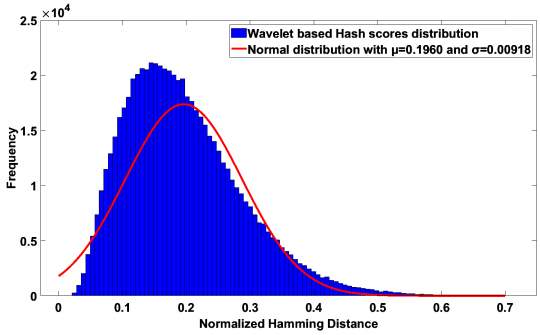
(a) A-Hash technique



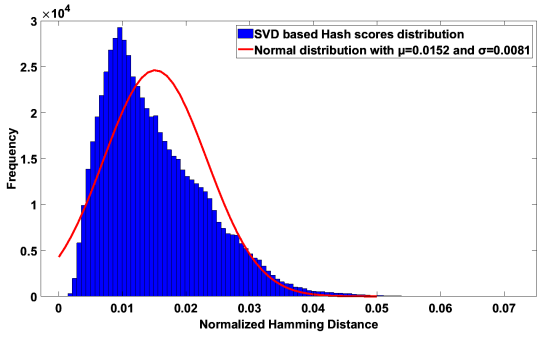
(c) P-Hash technique



(b) D-Hash technique



(d) W-Hash technique



(e) SVD-Hash technique

Figure 4.6: Distribution of Normalized Hamming distances corresponding to the five tested techniques.

Table 4.3: Collision Probability of the evaluated algorithms with different thresholds T

Technique	T	P_c	Technique	T	P_c
A-Hash	0.02	0.89×10^{-2}	**P-Hash**	0.04	0.00×10^{-2}
	0.04	01.36×10^{-2}		0.08	0.02×10^{-2}
	0.06	02.02×10^{-2}		0.12	0.09×10^{-2}
	0.08	02.93×10^{-2}		0.16	0.39×10^{-2}
	0.10	04.17×10^{-2}		0.20	01.40×10^{-2}
	0.12	05.79×10^{-2}		0.24	04.16×10^{-2}
D-Hash	0.04	0.03×10^{-2}	**W-Hash**	0.004	01.82×10^{-2}
	0.08	0.13×10^{-2}		0.008	02.03×10^{-2}
	0.12	0.50×10^{-2}		0.012	02.25×10^{-2}
	0.16	01.57×10^{-2}		0.016	02.50×10^{-2}
	0.20	04.24×10^{-2}		0.020	02.76×10^{-2}
	0.24	09.76×10^{-2}		0.024	03.05×10^{-2}
SVD-Hash	0.0008	03.77×10^{-2}			
	0.0010	03.98×10^{-2}			
	0.0012	04.20×10^{-2}			
	0.0014	04.42×10^{-2}			
	0.0016	04.66×10^{-2}			
	0.0018	04.90×10^{-2}			

4.3.3 Authentication results

Since perceptual robustness and discrimination are contradictory with each other, we evaluate, in this section, the overall performance of the five evaluated techniques in terms of tampering detection accuracy. This evaluation has been carried out by computing three metrics, namely: the Recall, Precision and F1-measure. The precision expresses the proportion of the images that the hashing algorithm identifies as visually similar and they actually are. Whereas, The Recall is the proportion of similar images that the algorithm identifies as similar. Therefore, Precision measures the accuracy of the algorithm to detect visually similar images, while Recall measures the ability of the algorithm to find all the similar images among the dataset [131]. These two metrics are calculated by

$$Precision = \frac{TP}{TP+FP}, Recall = \frac{TP}{TP+FN} \tag{4.2}$$

where TP is the true positive value which represents the images that have been labeled as similar by the algorithm and they actually are. FP is the false positive value, corresponding to the images that have been labeled as similar by the algorithm, but they are actually not; and FN is the false negative value, representing the images that have been labeled as different by the algorithm, but they are actually similar. The F1-score is the weighted harmonic average of the Precision and the Recall and it can be calculated as follows [131]

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4.3}$$

Note that higher values indicate better performance for these three metrics. The same set of tests described in Section 4.3.1 has been performed to compute the three

aforementioned metrics and the predetermined thresholds are chosen to provide the best results and are set as follows : A-Hash: $T = 0.04$, D-Hash : $T = 0.06$, P-Hash: $T = 0.08$, W-Hash: $T = 0.018$ and SVD-Hash: $T = 0.0018$. The obtained results are given in Table 4.4.

Table 4.4: Quantitative comparisons using the Recall, Precision and F1-measure metrics. Best results are in bold.

	A-Hash			**D-Hash**			**P-Hash**			**W-Hash**			**SVD-Hash**		
	Prec.	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**
GN	0.456	0.722	0.559	0.818	0.101	0.180	0.886	0.494	**0.651**	NaN	0.000	NaN	NaN	0.000	NaN
AF	0.461	0.899	0.609	0.912	0.971	0.941	0.878	1.000	0.935	0.984	0.977	**0.981**	0.240	0.159	0.191
SP	0.474	0.947	0.632	0.895	0.827	0.860	0.916	0.994	**0.953**	0.750	0.007	0.015	1.000	0.145	0.253
GB	0.472	0.976	0.636	0.894	0.992	0.941	0.886	1.000	0.939	0.979	0.997	**0.988**	0.087	0.100	0.093
GC	0.477	0.967	0.639	0.911	0.985	**0.946**	0.877	0.999	0.934	0.428	0.004	0.007	0.752	0.589	0.660
MB	0.441	0.889	0.590	0.906	0.784	0.840	0.944	0.862	0.901	0.979	0.979	**0.979**	0.064	0.072	0.068
JC	0.441	0.996	0.640	0.909	1.000	0.952	0.884	1.000	0.938	0.979	1.000	**0.989**	0.930	1.000	0.964
MF	0.427	0.707	0.532	0.901	0.766	0.828	0.872	0.914	**0.892**	0.935	0.162	0.277	0.342	0.242	0.284
WF	0.467	0.986	0.634	0.902	0.991	0.944	0.881	1.000	0.937	0.979	0.920	**0.948**	0.724	0.554	0.627
SH	0.471	0.989	0.638	0.909	0.996	0.950	0.889	1.000	0.941	0.977	0.994	**0.985**	0.956	0.877	0.915
SC	0.482	0.987	0.648	0.911	1.000	0.953	0.882	1.000	0.937	0.978	1.000	**0.989**	0.052	0.002	0.005
Mean	0.461	0.915	0.614	0.897	0.856	0.849	0.890	0.933	**0.905**	0.815	0.640	0.651	0.468	0.340	0.369

As it can be observed, and by evaluating the performance of each technique against a specific content-preserving operation, the results obtained from Table 4.4 can be described as follows:

- For the Gaussian noise manipulation, the P-Hash is the most powerful technique with a F1-score value of 0.651, whereas the W-Hash and SVD-Hash have not resisted to this attack and were unable to calculate the F1-score with Recall and Precision values of zero.
- For the Average filtering, the three techniques W-Hash, D-Hash and P-Hash performed very well and provide F1-score values of 0.981, 0.941 and 0.935 respectively. The worst technique is the SVD-Hash which generates a poor value of F1-score, equals to 0.191.
- For Salt & pepper noise addition, the P-Hash technique is in the first place with a F1-score value of 0.953, followed by the D-Hash technique with a F1-score value of 0.860, while the the SVD-Hash and W-Hash are the last-rank techniques with F1-score values of 0.253 and 0.015, respectively.
- For the Gaussian blurring, the most powerful techniques are W-Hash, D-Hash and P-Hash with F1-score values of 0.988, 0.941 and 0.939, respectively. The worst technique is SVD-Hash with a F1-score value of 0.093.
- For the Gamma correction manipulation, both D-Hash and P-Hash techniques performed well with F1-score values of 0.946 and 0.934, respectively. The W-Hash technique is the worst one with a F1-score value of 0.007.

- For the Motion blurring operation, the most successful techniques are the W-Hash, the P-Hash and the D-Hash with F1 score values of 0.979, 0.901 and 0.840, respectively. The worst performance is obtained by the SVD-Hash algorithm, which produces a F1-score value of 0.068.
- For the JPEG compression, all techniques performed well and provide good F1-score values, particularly, the W-Hash and the SVD-Hash, which have the highest F1-score values of 0.989 and 0.964, respectively. The A-Hash has generated the lowest F1-score value of 0.640.
- For the Median filtering, the best techniques are the P-Hash and the D-Hash with F1-score values of 0.892 and 0.828, respectively. The worst techniques are the SVD-Hash and the W-Hash with F1-score values of 284 and 0.277, respectively.
- For the Wiener filtering, almost all techniques performed well and the most successful ones are the W-Hash, the D-Hash and the P-Hash with F1-score values of 0.948, 0.944 and 0.937, respectively. The worst technique is the SVD-Hash which generates F1-score value of 0.627.
- For Image sharpening operation, except the A-Hash technique which yields a low F1-score value of 0.638, the other techniques were robust to this operation, and they all produce F1-score values over 0.951.
- For Image scaling operation, almost all techniques have resisted to this operation, except for the SVD-Hash algorithm which performed very poorly and provides a F1-score value of 0.005. The most robust algorithms are the W-Hash, the D-Hash and the P-Hash with F1-score values of 0.989, 0.953 and 0.937, respectively.

Overall, the most successful technique is P-hash, which achieves a F1-score value of 0.905 at a Precision value of 0.890 and a Recall value of 0.933. The second one is the D-hash, which reaches a F1-score value of 0.849 at a Precision value of 0.892 and a Recall value of 0.856. The next successful technique is the W-hash, which yields a F1-score value of 0.651 at a Precision value of 0.815 and a Recall value of 0.640. The forth place was assigned to the A-Hash algorithm, with a F1-score value of 0.614 at a Precision value of 0.461 and a Recall value of 0.915. Finally, the worst performance is obtained by the SVD-hash algorithm, which generates a F1-score value of 0.369 at a Precision value of 0.468 and a Recall value of 0.340.

4.4 Conclusion

In this chapter, we have examined and evaluated the performance of five commonly used and fast perceptual hashing techniques in the context of fingerprint images, which exhibit similar visual patterns due to the alternating ridges and valleys. The techniques studied include A-Hash, D-Hash, P-Hash, W-Hash, and SVD-Hash. Through

extensive experiments conducted on real fingerprint images, we focused on assessing three key aspects: perceptual robustness, discrimination capability, and authentication capacity.

The results obtained from our study are quite interesting. Regarding perceptual robustness, the A-Hash technique demonstrated superior performance compared to the other techniques, exhibiting greater resilience against various image manipulations. In terms of discrimination capability, all evaluated techniques showed comparable performance in terms of collision probability, with a slight advantage observed for the P-Hash technique. When considering the authentication property, which encompasses overall performance, the P-Hash technique yielded the best results across the range of applied manipulations.

While these results are promising, it is important to note that there is no universally superior technique, and the choice of a perceptual hashing technique should be based on the specific context of its application. In the upcoming chapter, we will delve into the utilization of a perceptual image hashing technique to enhance the security of a fingerprint-based biometric system against one of the most devastating attacks, namely the digital replay attack.

Chapter 5

A Replay Attack Detection Scheme based on Perceptual Image Hashing

5.1 Introduction

In this chapter, our focus is on the application of perceptual image hashing to safeguard biometric systems against digital replay attacks. We present a novel Replay Attack Detection (RAD) scheme that ensures the authenticity of received biometric images while also detecting if they have been resubmitted. A significant advantage of our proposed scheme is its ability to operate in uncontrolled environments without requiring the cooperation of end users. It leverages the Challenge/Response principle and combines perceptual image hashing with data hiding techniques.

Furthermore, we introduce a new perceptual image hashing system specifically designed for fingerprint images. This system offers both content authentication and identification capabilities. It utilizes a shift-invariant calibration signal technique that employs the Discrete Cosine Transform (DCT) and Discrete Sine Transform (DST) of the Differential Block Luminance Mean (DBLM) features to generate the hash sequence.

To evaluate the performance of our proposed system, we conducted extensive experiments using real fingerprint images. The results clearly demonstrate the system's ability to achieve high levels of authentication and identification accuracy. Furthermore, our proposed system surpassed existing state-of-the-art image hashing techniques in terms of performance.

Overall, this chapter presents a comprehensive approach to protecting biometric systems against digital replay attacks. Our RAD scheme and the novel perceptual image hashing system offer robust security features, making them valuable tools in safeguarding biometric data integrity and preventing unauthorized access.

5.2 The Proposed solution

Digital replay attacks are listed in the Common Weakness Enumeration (CWE) as weakness number 294 [132], and are described as "a capture-replay flaw exists when the design of the software makes it possible for a malicious user to sniff network traffic and bypass authentication by replaying it to the server in question to the same effect as the original message (or with minor changes)." They are a subset of network injection attacks that depend on observing previously sent valid commands, then slightly changing them if necessary and resending the same commands to the server [132]. In other words, DRA are a type of security attack that occurs by using biometric data sent from remote acquisition stations, which are equipped with sensors, to the central servers over an open network. In this kind of attacks, the imposter with unauthorized access, captures the image of the biometric modality (fingerprint, palm-print, iris, face, etc.), modifies the captured image and then resubmits it to its original destination, acting as a genuine user. The main feature of replay attacks is that the data are sent twice, and therefore the name replay attack. For example, suppose that a legitimate user wants to access a protected remote system using his fingerprints as a means of authentication. The system asks him to introduce and send his fingerprints images to confirm his identity. Meanwhile, an attacker eavesdrops on the communication and stores the fingerprints images of the authentic user. Once the transaction is completed, the attacker requests access to the remote system, pretending to be the legitimate user; when the system asks for his fingerprints, the attacker sends the previously intercepted fingerprint images. Therefore, the system believes to be dealing with a legitimate user and grants access to the attacker. An illustrative diagram of the replay attack is shown in Fig. 5.1.

Although this type of attack appears difficult to carry out in practice, studies pub-

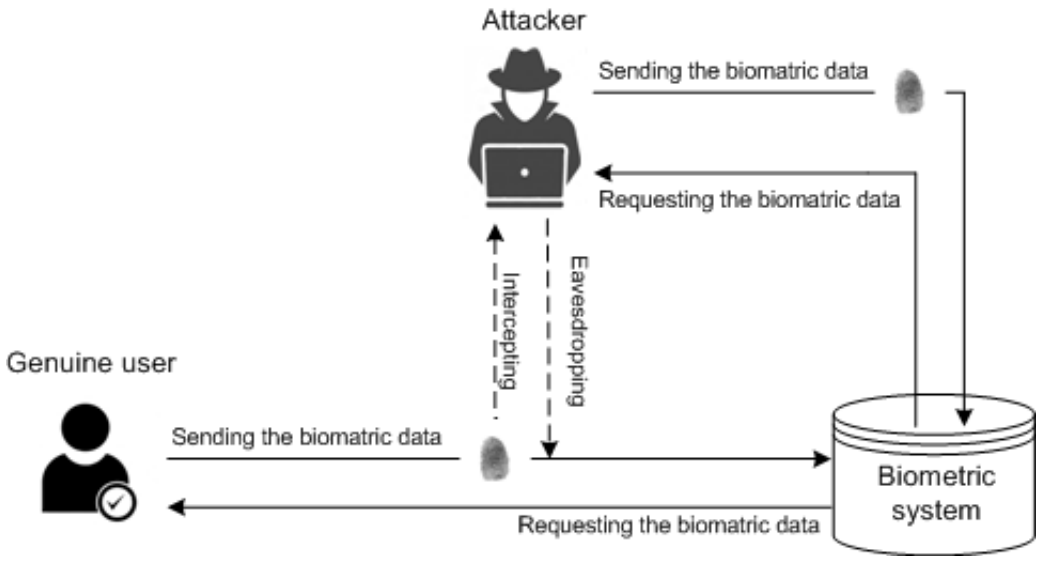


Figure 5.1: Illustrative diagram of a basic DRA.

lished in the literature have clearly shown the feasibility of these attacks. Adler [133],

proposed a replay attack on facial recognition systems and has shown that a coarse image of the source image can be reconstructed from biometric templates and by only using the scores returned by the algorithm of similarity between an input image and the image of the target user in an iterative manner. Hosseinzadeh *et al.* [134] have analyzed the effect of replay attacks on constrained cyber-physical systems which are subject to linear probabilistic constraints. In this case, the attacker hijacks the sensor by injecting an exogenous control input without being detected. Then he observes and records their readings for a certain amount of time and repeats them afterwards while carrying out his attack. The authors cited in their work the conditions under which the attacker can induce perturbations in the control loop without being detected. Furthermore, and as is well known, optical biometrics scanners are widely used to capture biometric samples, but the main stumbling block is that they do not have bidirectional communication and are not equipped with security modules or mechanisms that prevent replay attacks. Indeed, mostly available, biometric devices only capture the image, the video, or the audio signal, process it, and happily send it to the destination for the identification process.

5.2.1 The proposed RAD scheme

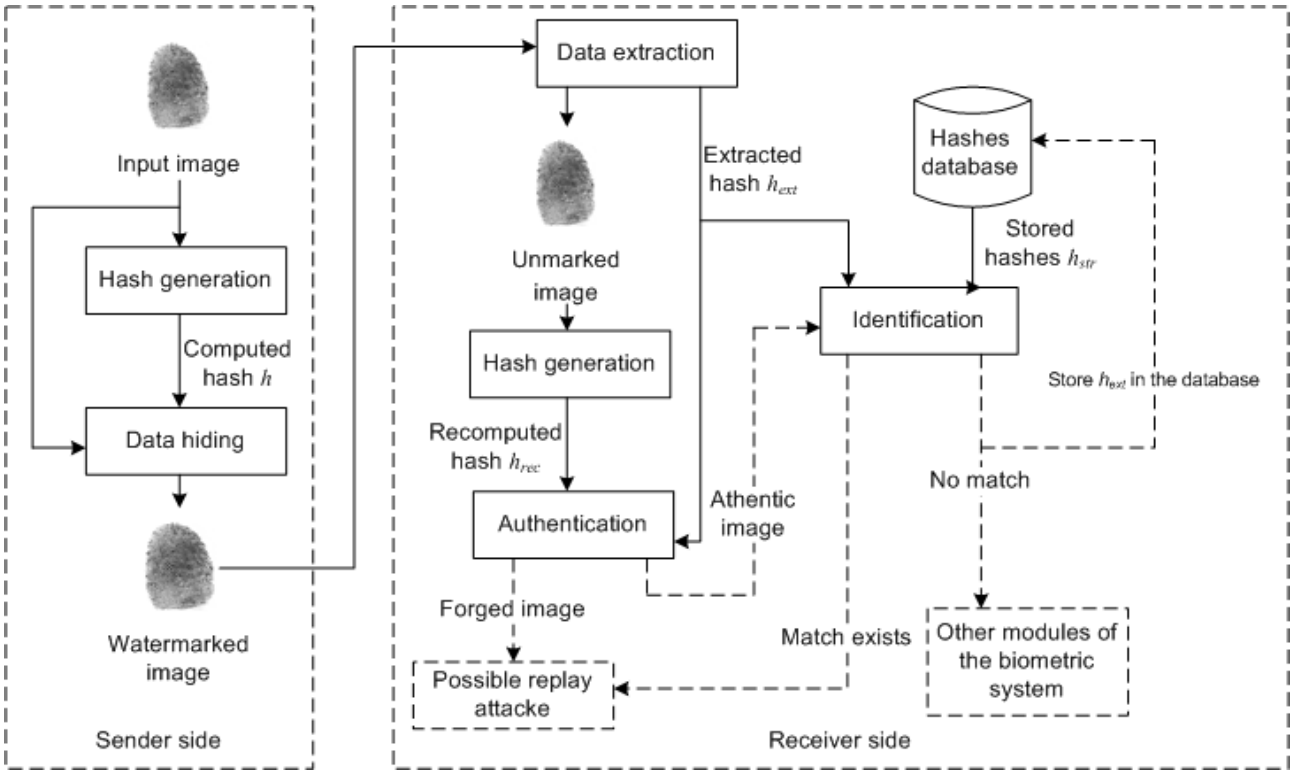


Figure 5.2: Block diagram of the proposed scheme.

In this work, an effective RAD scheme is proposed that relies on the Challenge/Response principle and combines perceptual image hashing and data hiding techniques to thwart digital replay attacks. The block diagram of the proposed scheme is depicted in Fig. 5.2, which comprises two main parts implemented at the sender and the receiver sides. It should be noted that for our scheme, the challenge corresponds

to calculate the robust hash of the acquired image and send them together. On the sender side (that is, the sensor), a robust hash sequence h is first calculated for the biometric input image and then sent along with its corresponding image to the receiver side (that is, the remote server or the processing module). Note that there are many ways to send the biometric image and its hash sequence. The simplest way is, for example, to send them separately on the same communication channel or on different channels. In our scheme, it is proposed to hide the hash sequence into its corresponding image by using an appropriate data hiding technique, to obtain the *watermarked image*. We believe that hiding the hash sequence into the image would strengthen the security of the proposed scheme and optimize the performance of the system (i.e., reducing computational complexity, optimizing channel bandwidth and saving storage space). It is recommended to use a Reversible Data Hiding (RDH) technique, since such techniques do not affect the biometric images and consequently preserve the hash sequences, which in turn, does not affect the performance of the authentication and identification operations. Note that, unlike the other data hiding schemes, RDH techniques can restore the original images after extracting the hidden data at the receiver side. In addition, we suggest using RDH techniques that are specifically designed for biometric images in order to get optimal performance. Although the data hiding technique is beyond the scope of this work, the readers can refer to the work of [135, 136] to get more information on the techniques that can be used in the proposed scheme.

On the receiver side, when the watermarked image is received, the hidden hash sequence h_{ext} is first extracted from the received image and then another hash sequence h_{rec} is recalculated from the unmarked image, using the same hash technique with the same parameters used by the sensor. After that, the two hash sequences h_{ext} and h_{rec} are submitted to the verification process to first authenticate the received image and then to identify if this image has been submitted before.

The proposed RAD scheme comprises two important steps, the authentication (one-to-one) and the identification (one-to-many). During the first step, the acquisition module sends the image with its hidden hash value to the target module that recalculates the robust hash of the received image using the same technique and the same parameters used by the sensor, then the two hashes are compared to each other (one to one) by calculating the measure of similarity between the recalculated hash sequence h_{rec} and the one sent by the sender h_{ext}. If the two sequences are not similar, then the received image may be forged and an alert of possible replay attack is sent to the system administrator. Otherwise, the received image is considered authentic and the extracted hash h_{ext} is transmitted to the identification process.

During the identification step, the similarity measure between the extracted hash sequence h_{ext} and those stored in the database $(h_{str1}, \cdots, h_{strk})$ is calculated (one to many) to ensure that this hash sequence has not been played before. If the hash sequence h_{ext} exists in the database (i.e, a match is detected between the received

hash sequence and one of the hash sequences stored in the database), then a possible replay attack is detected, and an alert is triggered. Otherwise, the hash sequence is inserted into the database and the received image can be further processed by the other modules of the biometric system.
To evaluate the efficiency of the proposed scheme, the following possible scenarios are analyzed :

- *Scenario 1*: The attacker has no knowledge about the perceptual hashing and the data hiding algorithms. Thus, he/she intercepts the watermarked image and replays it as is without any alterations. In this case, the extracted hash will be the same as the recalculated one, and the replayed image will pass the authentication step. However, at the identification stage, the extracted hash will be easily found in the database, and a replay attack will be detected.
- *Scenario 2*: The attacker knows all information about the data hiding algorithm including its secret keys generation process, but does not know the perceptual hashing algorithm. So, he/she can retrieve the hash from the watermarked image, apply some modifications to the image and re-hide the original extracted hash into the modified image. Depending on the intensity of the modifications, two cases are possible. If the modifications are somehow strong, the authentication step will notice the modifications (i.e. the forgery), and the replay attack is detected. Otherwise, if the modifications are weak, the replayed image may pass the authentication phase, but the identification process will easily detect that the extracted hash exists in the database, and a replay attack is detected.
- *Scenario 3*: The attacker knows all details about the perceptual hashing algorithm, but does not know the data-hiding algorithm and/or the secret keys. In this case, he/she intercepts the biometric image, applies some modifications and recompute the hash sequence of the modified image. However, he/she cannot hide it properly in the image. When replying the modified biometric image, the authentication process extracts the hidden hash (i.e. the one of the original image), which does not match the recalculated hash since the image has been modified, thus, a replay attack will be detected.
- *Scenario 4*: The attacker knows all details about the perceptual hashing and data hiding algorithms including any secret keys. In this scenario, he/she intercepts the biometric image, extracts the hidden hash, applies some modifications to the image, regenerates the hash, hides it into the modified image, and finally replays the watermarked image to the attacked system. The authentication process will not be able to detect the replay attack, since the extracted and recalculated hashes are similar, regardless of the strength of the modifications. For the identification process, its ability to detect replay attack will depend on the strength of the alterations. If they are not too strong, the identification can detect the attack;

otherwise, it will fail to detect it.

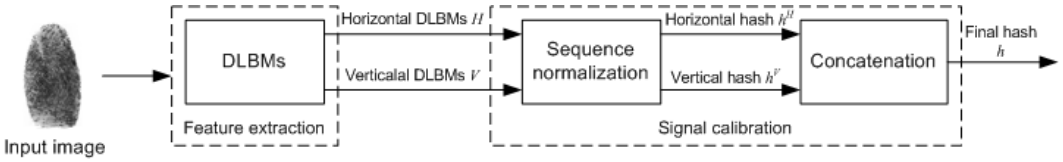
As can be noted, the only possible case where the proposed scheme may fail to detect DRA is in *Scenario 4*, which implies knowing both the perceptual hashing and the data hiding algorithms and their corresponding secret keys, in addition to applying strong modifications to the replayed image. Note that this is the less likely scenario because having knowledge of secret keys used by the two techniques is a very challenging issue for an attacker. Furthermore, if the applied modifications are too strong and exceed some limits, the recognition features used by the biometric system may be affected to a point that makes them useless, and the biometric system rejects the access to the attacker. Nevertheless, even if this scenario is extremely difficult to be performed in practice, the identification process should be designed to provide high performance, making the attacker to apply strong modifications (i.e., even the replayed data succeeds in bypassing the proposed countermeasure, it will be rejected by the biometric system and no access is granted to the attacker).

5.2.2 Proposed perceptual hashing system

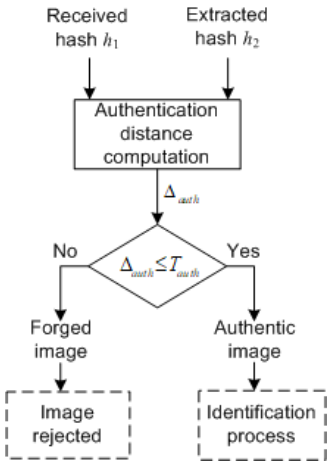
In order to implement the proposed scheme, a new perceptual image hashing system for fingerprint images that operates in the transform domain is also proposed in this work. The proposed system presents the advantage of performing both image identification and authentication operations using the same hash sequence. The block diagram of this system is depicted in Fig.5.3, which comprises three main stages, namely: (i) hash generation, (ii) content authentication, and (iii) content identification. In fact, the proposed system is built upon the works of Ghouti [37] and Khelifi *et al.*[38], which are video hashing systems for both content identification and authentication. Indeed, the hash generation stage is based on a shift signal calibration of the Discrete Cosine Transform (DCT) and the Discrete Sine Transform (DST) coefficients, in order to compute the hash values. The basic principle is to find the number of the normalization shift values, that allow to shift the digital signal in a way that its modified versions match a predefined pattern according to the DCT/DST coefficients. Two different similarity measures, and consequently two different thresholds, are used to perform authentication and identification. It is worth mentioning that to the best of our knowledge, only very few published works have proposed perceptual hashing systems that simultaneously target content authentication and identification applications : one for still images [137], and two for videos [37] and [38].

5.2.3 Hash generation

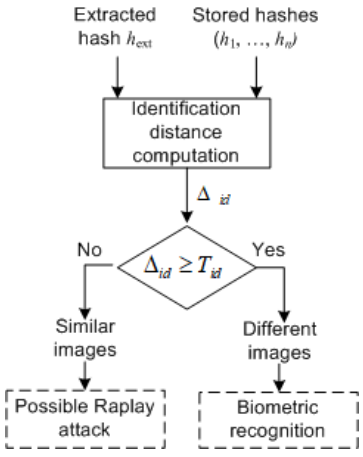
As illustrated in Fig.5.3a, the hash generation technique comprises two main stages, namely feature extraction and signal calibration. In the first stage, the Differential Luminance Block Means (DLBM) of the input image, representing the image features, are computed, whereas in the second stage, a shift-based signal calibration technique is applied to the extracted features to obtain the final hash sequence.



(a) Hash generation



(b) Image authentication



(c) Image Identification

Figure 5.3: Proposed perceptual hashing system.

It should be noted that our hash generation technique is an adapted version of the technique proposed in [38], where the following two major modifications have been made.

- Firstly, the pre-processing step has been omitted because this step, as proposed in [38], consists of reducing the video signal to a small frame rate (i.e. 5 frames/second), which is not applicable in our case.
- Secondly, the dimensions of the space have been reduced from three dimensions (3D) to two dimensions (2D) by eliminating the time space; hence, the whole mathematical equations have been modified.

A. Feature extraction: In this stage, the Differential Luminance Block Means (DLBM) are used as robust features to describe image information in both horizontal and vertical directions. This choice is justified by their robustness, their low complexity and their efficiency in representing textured areas [38, 138] which is suitable for highly textured images such as fingerprints.
Let I be the input image. First, I is divided into $M \times N$ non-overlapping blocks of size $h \times w$. Note that the size $h \times w$ is defined so that the image is divided into $M \times N$ blocks. Then, the mean of each block is calculated to form a 2D matrix of size $M \times N$, denoted by $A(i, j)$, with $0 \leq i \leq M-1$ and $0 \leq j \leq N-1$, as follows:

$$A(i,j) = \frac{\sum_{k=i\times h}^{i\times h+h-1} \sum_{l=j\times w}^{j\times w+w-1} I(k,l)}{h \times w} \tag{5.1}$$

Afterwards, the differences in the horizontal H and vertical V directions of A are computed, to generate two 2D matrices of the same size, as follows:

$$H(i,j) = \begin{cases} A(i,j+1) - A(i,j) \; if \; j < N-1 \\ A(i,0) - A(i,j) \; if \; j = N-1 \end{cases} \tag{5.2}$$

$$V(i,j) = \begin{cases} A(i+1,j) - A(i,j) \; if \; i < M-1 \\ A(0,j) - A(i,j) \; if \; i = M-1 \end{cases} \tag{5.3}$$

B. Signal calibration The final hash is generated by applying a shift-based calibration that aims to determine the normalization shift for the feature sequences. In fact, the normalization shift represents the number of samples by which a sequence can be shifted in a way that matches a certain pattern. In this work, this is achieved by following a shift-invariant signal normalization method that uses the DCT and DST coefficients of the sequence to represent the pattern.
Let $S(n)$ with $n \in \{0, 1, \cdots, L-1\}$ be a signal that represents one of the two matrices H and V defined above, after crossing it in one direction. The two DCT and DST transforms used in this work are defined, respectively, as follows:

$$S^{DCT}(m) = \sum_{n=0}^{L-1} S(n) \cos\left(\frac{\pi m(n+\frac{1}{2})}{L}\right); \quad m,n \in \{0,1,\cdots,L-1\} \tag{5.4}$$

and

$$S^{DST}(m) = \sum_{n=0}^{L-1} S(n) \sin\left(\frac{\pi(m+1)(n+\frac{1}{2})}{L}\right); \quad m,n \in \{0,1,\cdots,L-1\} \tag{5.5}$$

As suggested in [38], the normalization technique provides the same sequence in the case of a circular translation of the input signal. By considering the following sequence of samples:

$$S_0 = \{S(0), S(1), \cdots, S(L-1)\}, \tag{5.6}$$

Its shifted version by one sample can be expressed by:

$$S_1 = \{S(1), S(2), \cdots, S(L-1), S(0)\}, \tag{5.7}$$

By letting $\omega = \frac{\pi m}{L}$, it has been shown in [139] that:

$$S_1^{DCT}(m) = \cos(\omega) S_0^{DCT}(m) + \sin(\omega) S_0^{DST}(m-1) + S(0)\cos\left(\frac{\omega}{2}\right)((-1)^m - 1), \tag{5.8}$$

and

$$S_1^{DST}(m-1) = -\sin(\omega) S_0^{DCT}(m) + \cos(\omega) S_0^{DST}(m-1) + S(0)\sin\left(\frac{\omega}{2}\right)(1-(-1)^m) \tag{5.9}$$

In the following, only even values of m are considered (that is $m = 2,4,\cdots,2p$) because, as explained in [38], odd values of m do not produce any results. Therefore, Eq.5.8 and Eq.5.9 can be simplified as follows:

$$S_1^{DCT}(m) = \cos(\omega) S_0^{DCT}(m) + \sin(\omega) S_0^{DST}(m-1), \tag{5.10}$$

and

$$S_1^{DST}(m-1) = -\sin(\omega) S_0^{DCT}(m) + \cos(\omega) S_0^{DST}(m-1). \tag{5.11}$$

As can be seen, the DCT and DST coefficients of a sequence shifted by one sample can be obtained as a function of the DCT and DST coefficients of the original one [38]; therefore, the following recursive equations can be defined.

$$S_{r+1}^{DCT}(m) = \cos(\omega) S_r^{DCT}(m) + \sin(\omega) S_r^{DST}(m-1); \tag{5.12}$$

and

$$S_{r+1}^{DST}(m-1) = -\sin(\omega) S_r^{DCT}(m) + \cos(\omega) S_r^{DST}(m-1). \tag{5.13}$$

where $r \in \{0,1,\cdots,L-1\}$.

From Eq.5.12 and Eq.5.13, it has been proven in [38] that the DCT and DST coefficients of the shifted version by r samples can be expressed by the DCT and the DST coefficients of the original sequence and these coefficients follow a cosine function with the same magnitude as follows:

$$S_r^{DCT}(m) = \sqrt{(S_0^{DCT}(m))^2 + (S_0^{DST}(m-1))^2}\cos\left(\left(\omega\, r - \arctan\left(\frac{S_0^{DST}(m-1)}{S_0^{DCT}(m)}\right)\right)\right) \tag{5.14}$$

and

$$S_r^{DST}(m-1) = \sqrt{(S_0^{DCT}(m))^2 + (S_0^{DST}(m-1))^2}\cos\left(\omega\, r - \arctan\left(\frac{S_0^{DST}(m-1)}{S_0^{DCT}(m)}\right) + \frac{\pi}{2}\right) \tag{5.15}$$

In view of Eq.5.14 and Eq.5.15, it is observed that the DCT and DST coefficients of a shifted sequence arise at a shift factor of $\frac{2L}{m}$. Therefore, for $m = 2$, there is only one pair of coefficients $(S_r^{DCT}(2), S_r^{DST}(1))$ in the interval $[0, L-1]$, this is because there is one period of the cosine function with the variable r in the interval $[0, L-1]$. In other words, by setting $m = 2$, each shifted sequence by one sample among the L possible shifted sequences corresponds to one pair of coefficients $(S_r^{DCT}(2), S_r^{DST}(1))$. As mentioned above, the goal of the normalization shift step is to find the number of shifts that provides the same DCT/DST coefficients of the original sequence and its shifted version and, as explained in [38], for a given pair of coefficients $(S_{r^*}^{DCT}(2), S_{r^*}^{DST}(1))$, this can be achieved simply by finding its corresponding shifted sequence, which is known as the normalized sequence. Consequently, and considering Eq.5.14, the normalization shift r^* can be obtained by setting a reference angle θ in $[0, 2\pi]$, as follows:

$$\left(\omega r^* - \arctan\left(\frac{S_0^{DST}(1)}{S_0^{DCT}(2)}\right)\right) = \theta \tag{5.16}$$

Finally:

$$r^* = \left\lfloor \frac{L\arctan\left(\frac{S_0^{DST}(1)}{S_0^{DCT}(2)}\right) + L\theta}{2\pi} \right\rfloor \mod L \tag{5.17}$$

As can be observed in Eq.5.17, the signal calibration operation determines the normalization shift using a predetermined value of the reference angle θ. This operation is applied to the two feature matrices H and V to obtain two 1-D sequences denoted by h^H and h^V, respectively. These two sequences represent the hash in each direction and are described as follows:

$$h^H(i) = \left\lfloor \frac{N\arctan\left(\frac{H_i^{DST}(1)}{H_i^{DCT}(2)}\right) + N\theta}{2\pi} \right\rfloor \mod N \tag{5.18}$$

$$h^V(j) = \left\lfloor \frac{M\arctan\left(\frac{V_j^{DST}(1)}{V_j^{DCT}(2)}\right) + M\theta}{2\pi} \right\rfloor \mod M \tag{5.19}$$

where $H_i = H(i,0), \cdots, H(i,N-1)$ and $V_j = V(j,0), \cdots, V(J,M-1)$. X^{DCT} and X^{DST} are the DCT and TSD coefficients of the sequence X, respectively. It should

be noted that the elements of h^H and h^V are delimited because they represent the values of the normalization shifts (i.e., $0 \leq h^H(i) \leq N-1$ and $0 \leq h^V(j) \leq M-1$). The final hash h can be obtained simply by concatenating the two sequences h^H and h^V (i.e. $h = \{h^H, h^V\}$). The hash generation algorithm can be described by Algorithm 1.

Algorithm 1: Hash generation

Input: Input image I, Reference angle θ, Number of vertical blocks M, Number of horizontal blocks N,
Output: Final hash vector h.

1. — Compute the 2-D DBLM arrays H and V, representing the features of I in the horizontal and vertical directions, respectively, using Eq.5.2 and Eq.5.3.
2. — Calculate the normalization shift sequences h^H and h^V for the arrays H and V using Eq 5.18 and Eq. 5.19, respectively.
3. — Concatenate the two sequences h^H and h^V to get the final h.
4. — Return the final hash h.

5.2.4 Content authentication

A hashing-based authentication system should be able to differentiate between a maliciously forged image and a manipulated one with common and tolerated image manipulation operations such as compression and coding. In this case, the following requirements are considered:

- 1^{st} *requirement* : $\forall I; \Delta_{auth}(h, \hat{h}) \geq T_{auth}$
- 2^{nd} *requirement* : $\forall I; \Delta_{auth}(h, \bar{h}) < T_{auth}$

where I and h are, respectively, the original image and its corresponding hash. $\hat{h}$ is the hash of the forged image $\hat{I}$, whereas $\bar{h}$ is the hash of the manipulated image with tolerated operations $\bar{I}$. The first requirement indicates that the distance Δ_{auth} between the original hash and the one of the forged images should exceed the authentication threshold T_{auth}. However, in the case of image manipulation, the original hash is very similar to the hash of the manipulated image, where the distance between them is less than T_{auth}.

In our case, the authentication operation is performed by evaluating the similarity between the received hash and the extracted hash from the received image. As explained earlier, in replay attacks, it is very likely that an attacker introduces some modifications to the biometric data, trying to make it different from the original version. However, since the replayed data is used to gain access after being processed by the biometric system, a minimum number of modifications would be applied, and their intensity are kept as low as possible in order to preserve the main characteristics used by the biometric system, otherwise the biometric data becomes useless. For example, in the case of fingerprint images, we believe that geometric manipulations such as rotation and shifting are the most applicable modifications because in practice, each time a finger is scanned, the acquired fingerprint image is slightly different

from the other images of the same finger (in most cases looks like rotated, shifted or both), and biometric systems are designed to successfully match these different images.
In this work, the authentication distance is computed using the Euclidean distance as follows:

$$\Delta_{auth} = \sum_{i=0}^{M-1} (h_1^H(i) - h_2^H(i))^2 + \sum_{j=0}^{N-1} (h_1^V(j) - h_2^V(j))^2 \tag{5.20}$$

where $h_1 = \{h_1^H\, h_2^V\}$ and $h_2 = \{h_1^H\, h_2^V\}$ are the hashes of I_1 and h_2, respectively.
The above requirements can be formulated as a binary hypothesis testing as follows:

$$\begin{aligned} \Delta_{auth} \leq T_{auth} &\implies H_0 \\ > T_{auth} &\implies H_1 \end{aligned} \tag{5.21}$$

where H_0 represents the hypothesis of authentic image, whereas H_1 is the hypothesis of forged images. The threshold T_{auth} can be empirically obtained using the Neyman-Pearson criterion in a way to minimize the False Negative Probability P_{FN} for a fixed False Positive Probability (P_{FP}^*) [140].

$$\begin{aligned} P_{FP} &= Prob(\Delta_{auth} > T_{auth} | H_0) \\ &= \int_{T_{auth}}^{\infty} f_{\Delta_{auth}}(t|H_0)dt \end{aligned} \tag{5.22}$$

Where $f_{\Delta_{auth}}(t|H_0)$ is the probability density function (PDF) of Δ_{auth} under hypothesis H_0. By assuming that Δ_{auth} follows a Normal distribution under H_0, the threshold T_{auth} can be calculated by the following equation:

$$T_{auth} = \sqrt{2\sigma_{\Delta_{auth}}^2}\, erfc^{-1}(2P_{FP}^*) + \mu_{\Delta_{auth}} \tag{5.23}$$

where $efrc(.)$ is the complementary error function. $\mu_{\Delta_{auth}}$ and $\sigma_{\Delta_{auth}}$ are, respectively, the statistical mean and standard deviation of the authentication distance Δ_{auth}. The two later can be computed from the empirical distance between the original images and their manipulated versions (by tolerable image manipulation operations).
It is worth noting that if no manipulation is tolerated on the received image (i.e., the received image is exactly the same as it has been acquired by the sensor), then the value of T_{auth} is set to zero (0).
We summarize the authentication step in the following Algorithm 2:

5.2.5 Content identification

As mentioned earlier, one characteristic of fingerprint images, and biometric images in general, is that they visually look similar, especially for non-expert persons. Therefore, an efficient identification process should be able to detect dissimilarities between different fingerprint images. Furthermore, identification process should be

Algorithm 2: Authentication

Input: Received image I, Hash sequence sent by the sender h_{rec}, Authentication threshold T_{auth}
Output: Authentic value Av, Extracted hash sequence h_{ext}.

1. — Calculate the hash sequence h_{ext} from the received image I using Algorithm 1.
2. — Compute the authentication distance Δ_{auth} between the recalculated hash sequence h_{ext} and the received one h_{rec} using Eq 5.20.
3. — If $\Delta_{auth} < T_{auth}$ then
4. Set the Authentic value Av to true.
5. Transmit the extracted hash h_{ext} to the identification process.
   else
6. Set the Authentic value Av to false.
7. Trigger an alert of possible replay attack.
   end If.

able to differentiate the images acquired from the same finger, and this in fact is a very challenging task. It is worth mentioning that unlike the biometric identification which aims at identifying if the fingerprint images are acquired from the same finger, the identification in the context of perceptual hashing intends to verify if an image has been derived from another one by applying some image processing operations. In our case, the identification process is used to compare the hash of the received image with all the hash values stored in the database, in order to detect whether the image has been submitted before or not. Consequently, the identification process should take into account the following requirements:

- 1^{st} *requirement* : $\forall I_i, I_j;\ if\ I_i \nsim I_j\ then\ \Delta_{id}(h_i, h_j) \geq T_{id}$
- 2^{nd} *requirement* : $\forall I_i, I_j;\ if\ I_i \sim I_j\ then\ \Delta_{id}(h_i, h_j) < T_{id}$

where I_i and I_j are two images and h_i and h_j are their hashes, respectively. $\sim$ denotes perceptually similar, while $\nsim$ represents perceptually different. Δ_{id} is the function that calculates the distance between two hashes and T_{id} is the identification threshold. The binary hypothesis testing is also used here to represent the above requirements as follows:

$$\begin{aligned} \Delta_{id} &\geq T_{id} \implies H_0 \\ &< T_{id} \implies H_1 \end{aligned} \tag{5.24}$$

where H_0 is the hypothesis that the compared images are perceptually different, whereas H_1 is the hypothesis of perceptually similar images.
In this work, the identification distance Δ_{id} is defined as described in [38] with the introduction of some modifications. Let $\{h_1^H, h_1^V\}$ and $\{h_2^H, h_2^V\}$ be the hashes of two

images I_1 and I_2, respectively. The identification distance can be given as follows:

$$\begin{aligned}\Delta_{id} = &\sum_{i=0}^{M-1} \min\{|h_1^H(i) - h_2^H(i)|, |h_1^H(i) - h_2^H(i) - N|, |h_1^H(i) - h_2^H(i) + N|\} \\ &+ \sum_{j=0}^{N-1} \min\{|h_1^V(j) - h_2^V(j)|, |h_1^V(j) - h_2^V(j) - M|, |h_1^V(j) - h_2^V(j) + M|\}\end{aligned} \tag{5.25}$$

The threshold T_{id} can also be determined empirically using the Neyman-Pearson criterion, and hence the P_{FP} can be given by:

$$\begin{aligned}P_{FP} &= Prob(\Delta_{id} < T_{id}|H_0) \\ &= \int_{-\infty}^{T_{id}} f_{\Delta_{id}}(t|H_0)dt\end{aligned} \tag{5.26}$$

Although $f_{\Delta_{id}}(t|H_0)$ is the PDF of Δ_{id} under H_0. Suppose that Δ_{id} follows a Normal distribution under H_0, then the threshold T_{id} can be defined as follows:

$$T_{id} = \sqrt{2\sigma_{\Delta_{id}}^2} erfc^{-1}(2 - 2P_{FP}^*) + \mu_{\Delta_{id}} \tag{5.27}$$

where $\mu_{\Delta_{id}}$ and $\sigma_{\Delta_{id}}$ are, respectively, the statistical mean and standard deviation of the identification distance Δ_{id}.
We summarize the identification step in the following Algorithm 3:

Algorithm 3: Identification

Input: Extracted hash value h_{ext},
Output: Database insert operation / alert.

1. — Computes the similarity measure between the extracted hash sequence h_{ext} and those stored in the database ($h_{str1}, \cdots, h_{strk}$);.
2. — If the hash sequence h_{ext} exists in the database, then
3. Trigger an alert of a possible replay attack.
   else
4. Insert the hash sequence h_{ext} into the database.
5. Allow the received image to be further processed by the receiver.
   end If.

5.3 Experimental results and discussions

To assess the performance of our proposed perceptual system, a set of experiments has been carried out on real fingerprint images from the database 'FVC 2000, DB3 _a' [141], which consists of 800 fingerprint images of size 448×478. Actually, the test database contains the images of 100 different fingers, and each finger has eight images. Note that these images have been slightly resized to have a size of 448×480 (that is, to obtain height and width values divisible by 8). A sample of four images with different quality levels is shown in Fig.5.4. For all experiments, the angle value α is set to $\frac{\pi}{20}$. In our analysis, the following five state-of-the-art perceptual

(a) Image 22_1

(b) Image 20_1

(c) Image 2_5

(d) Image 16_1

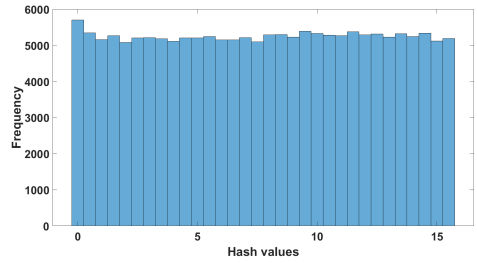
Figure 5.4: Examples of images used in our tests.

hashing schemes have been considered for reference: (i) the Local Binary Pattern (LBP) operator based hashing technique that operates in the spatial domain [142], (ii) the Noise Resistant Local Binary Pattern (NRLBP)-based scheme [143], (iii) the Singular Value Decomposition (SVD)-based NRLBP scheme, denoted by SVD-NRLBP [144], (iv) SVD-based Center-Symmetric Local Binary pattern scheme, denoted by SVD-CSLBP [145], and (v) DCT-NRLBP scheme [146]. The obtained results and discussions have been divided into: generated hash analysis, identification performance and authentication performance subsections.

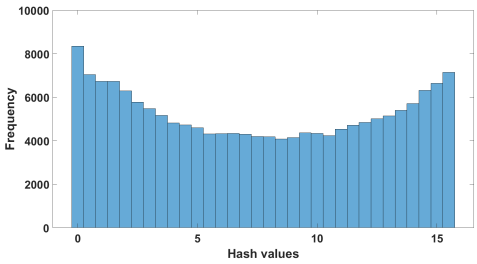
5.3.1 Generated Hash analysis

In this section, the statistical distribution of the generated hash values was analyzed to evaluate the discriminative capability of the proposed technique. In fact, as reported in [38], generating hash values that are almost uniformly distributed over the interval of possible values would guarantee a high level of discriminative capability. Consequently, generating some hash values with higher probability than others would increase the possibility of getting similar hashes for visually different images, and this fact decreases the discriminative capability. In this test, the hashes from 800 images of the test database were extracted, for different block sizes (i.e., 8×8, 16×16 and 32×32) and their distribution were analyzed. Fig. 5.5 shows the his-

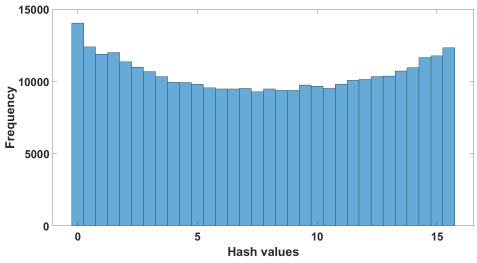
togram of the extracted hashes values in each direction for a block size of 32×32.



(a) In the horizontal direction



(b) In the vertical direction



(c) Final hash (i.e., after concatenation)

Figure 5.5: Histogram of the generated hash values. Block size: 32×32.

As can be seen, Fig. 5.5a clearly indicates that the values of the horizontally extracted hashes are equally distributed, while Fig. 5.5b shows that the values of the vertically extracted hashes are slightly concentrated around the limit interval values. For the final hashes, the histogram of their values, presented in Fig. 5.5c, reveals that, in general, they are more or less evenly distributed over the range of possible values. This result indicates that the proposed hash-generation technique allows for a fairly equitable distribution of the extracted hash values across all possible values. Note that the results for the other block sizes (that is, 8×8 and 16×16) are very similar to those shown in Fig. 5.5.

In the second test, we have compared the length of the generated hash using our technique and the other reference techniques, and the results obtained are given in Table 5.1. Note that the hash length is a very important parameter in real situations, since the generated hashes are stored into a database for identification purposes, hence, generating short hash sequences would optimize our proposed system. In our case, the length of the generated hashes depends on the size of the blocks; the larger the block size, the shorter the hash length and vice versa. From Table 5.1, it is observed that, except for the NRLBP scheme[143], the proposed hash generation

Table 5.1: Hash length comparison

Scheme	Hash length
LBP [142]	16384
NRLBP [143]	59
SVD-NRLBP [144]	3776
SVD-CSLBP [145]	4096
DCT-NRLBP [146]	3776
Proposed (8×8)	**3360**
Proposed (16×16)	**840**
Proposed (32×32)	**210**

technique provides the smaller hash lengths for all block sizes, when compared to the other schemes.

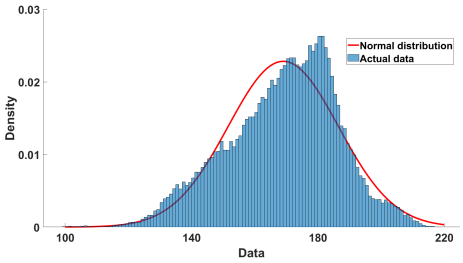
5.3.2 Authentication Performance

As mentioned above, an efficient authentication system should be able to differentiate between a forged image and a manipulated image by tolerated operations. In this work, we have chosen JPEG compression with a quality equal to 80 as the tolerated manipulation. In this subsection, the performance of the authentication process was assessed by : (i) analyzing the authentication distances, and (ii) evaluating the robustness in the presence of malicious distortions.

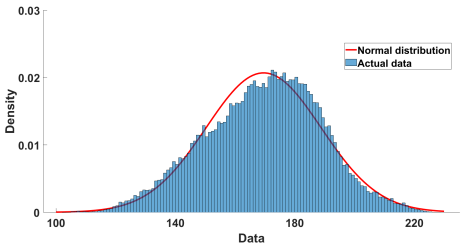
- **A. Authentication distance analysis**

As assumed in Section 5.2.4, the authentication distance Δ_{auth} can statistically be modeled by a Normal distribution, when tolerated manipulations are intentionally applied to the images. Under this assumption, the authentication thresholds T_{auth}, which are crucial parameters that control the system's accuracy, are computed; hence, it is important to verify this fact, which is the aim of this test. To achieve this, the test was performed as follows: first, the original hashes were extracted from the 800 images of the database, for different block sizes (i.e., 8×8, 16×16 and 32×32). The test images were then compressed, and for each block size, the hashes were generated from the resulting images. After that, the authentication distance between each original hash and its corresponding hash extracted from the compressed image is calculated. Finally, the distance sets obtained have been examined using Quantile-Quantile (Q-Q) plots, which is a graphical technique used to visually compare and determine if two data sets come from populations with a common distribution [147]. To obtain the Q-Q graphs, the parameters $\mu_{\Delta_{auth}}$ and $\sigma_{\Delta_{auth}}$ were first estimated from each set and then a large number of random samples drawn from a Normal distribution with the estimated parameters $\mu_{\Delta_{auth}}$ and $\sigma_{\Delta_{auth}}$ were generated. The quantiles of the real authentication distances against the quantiles of the random generated samples are plotted. Fig. 5.6 represents the distribution of the obtained identification distances sets, whereas Fig. 5.7 shows the Q-Q plots.
As one can observe, the majority of the values are close to the curve in red (i.e.,

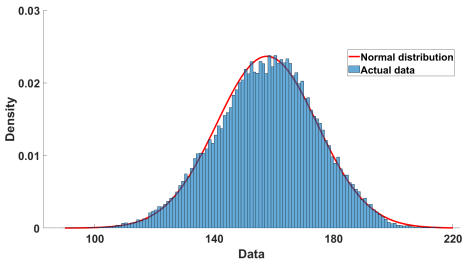
the reference line), which represents the theoretical distribution according to the estimated parameters, with a set of points that move away from the theoretical curve. This set is small for block size 32×32; a little more significant for size 16×16, and even larger for size 8×8. Overall, these results confirm that the authentication distances for their majority follow the Normal law, with a superiority for the results obtained for size 32×32.



(a) Block size: 8×8



(b) Block size: 16×16



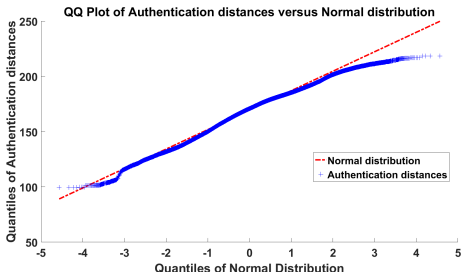
(c) Block size: 32×32

Figure 5.6: Authentication distances histograms for different block sizes.

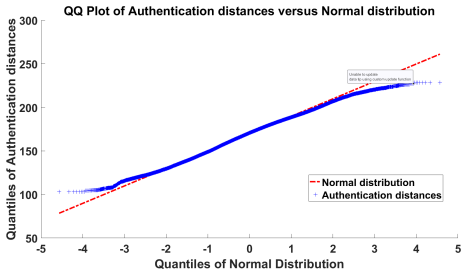
- **B. Authentication accuracy**

The authentication accuracy was evaluated in terms of two factors: (i) the False Positive Rate (FPR), which represents the rate of declaring an authentic image as forged, and (ii) the True Positive Rate (TPR), which is the rate of successfully detecting a forged image.

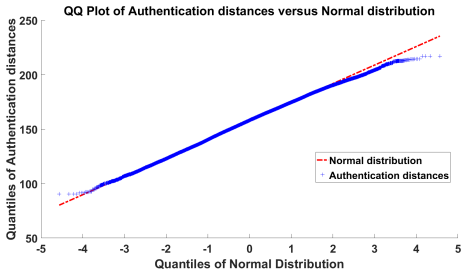
In the first test, FPR of the authentication process was measured between the original images and their corresponding manipulated versions (that is, compressed by JPEG with a quality of 80). To achieve this, for each block size, the authentication thresholds T_{auth} for the theoretical Probability of False Positive (P^*_{FP}) varied from 10^{-1} to 10^{-5} are calculated as defined by Eq. 5.23. Then, the sets of authentication distances obtained in the previous test are compared with the corresponding threshold, and each time a forge is detected, it is considered as an authentication error (i.e.,



(a) Block size: 8×8



(b) Block size: 16×16



(c) Block size: 32×32

Figure 5.7: Quantile-Quantile plots of the authentication distances for different block sizes.

+1 false positive). The results obtained are reported in Table 5.2 and clearly show that, for all block sizes, no false positive detection was recorded for P^*_{FP} values of 10^{-4} and higher. We can also notice a slight superiority for the block size 8×8, for which the lowest FPR values have been obtained. In the second experiment, the

Table 5.2: False Positive Rate (FPR) of the authentication process for different block sizes.

Block size	Theoretical probability of false positive (P^*_{FP})				
	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
8×8	7.8×10^{-2}	3.4×10^{-3}	0.0	0.0	0.0
16×16	8.7×10^{-2}	7.1×10^{-3}	2.0×10^{-5}	0.0	0.0
32×32	9.9×10^{-2}	5.8×10^{-3}	4.5×10^{-4}	0.0	0.0

TPR of the authentication process is computed in the presence of malicious distortions. In our case, we have particularly focused on image rotation and shifting as geometric distortions because as explained in Section 5.2.4, these two distortions are the most likely distortions that an attacker would apply in DRA. So, to compute the TPR values, we proceeded as follows: first, we took an image from the set of the test images and applied slight distortions, namely: image rotation with different angles (from -2° to $+2^\circ$) and image shifting with different values (from $[2,2]$ to $[5,5]$); this resulted in a set of distorted images. Then, for each block size, the hashes of the original and the distorted images were generated. Subsequently, the authentication distances Δ_{auth} between these hashes were calculated and compared with the thresholds obtained in the previous test. The results obtained for the image rotation and shifting are given in Table 5.3 and Table 5.4, respectively. It is worth mentioning that the aim of applying modifications with slight strength is to assess the degree of sensitivity of the authentication process to small distortions.

With respect to image rotation, it can be seen in Table 5.3 that for different block sizes, and even the rotation angles are very small, the authentication process succeeded in detecting all forged images with different rotation angles, and this for P^*_{FP} values of 10^{-5} and above. Particularly, the results have shown a clear superiority for block size of 32×32 over the other block sizes, where the authentication process has almost detected all forged images for the different values of P^*_{FP} and for the tested rotation angles. Another interesting observation is that, the results obtained for block size of 8×8 are slightly better than those obtained for block size of 16×16. With respect to image shifting, the results reported in Table 5.4 shows that the best performance was reported for the block size 32×32, for which the authentication process has detected all forged images for P^*_{FP} values from 10^{-1} to 10^{-5}. For the other block sizes, good values of TPR were recorded for P^*_{FP} values of 10^{-7} and above. Overall, the authentication process has shown a high sensitivity to image rotation and shifting operations. For the sake of completeness, we have also evaluated the accuracy of the authentication process when other manipulations are applied to the images. In these

Table 5.3: True Positive Rate (TPR) of the authentication process under rotation distortion with different angle values.

Bk size	Strength	Theoretical probability of false positive (P^*_{FP})									
		10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
8×8	-2°	1.0	1.0	1.0	1.0	1.0	1.0	0.9862	0.6787	0.0813	0.0187
	-1°	1.0	1.0	1.0	1.0	1.0	1.0	0.9500	0.5150	0.0437	0.0163
	$+1^\circ$	1.0	1.0	1.0	1.0	1.0	1.0	0.9537	0.5350	0.0525	0.0175
	$+2^\circ$	1.0	1.0	1.0	1.0	1.0	1.0	0.9838	0.7025	0.0838	0.0175
16×16	-2°	1.0	1.0	1.0	1.0	1.0	1.0	0.9175	0.3262	0.0400	0.0150
	-1°	1.0	1.0	1.0	1.0	1.0	0.9930	0.6887	0.1113	0.0238	0.0050
	$+1^\circ$	1.0	1.0	1.0	1.0	1.0	0.9912	0.6925	0.1175	0.0225	0.0013
	$+2^\circ$	1.0	1.0	1.0	1.0	1.0	0.9988	0.8825	0.3275	0.0437	0.0088
32×32	-2°	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9988	0.9750	0.8725
	-1°	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	$+1^\circ$	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	$+2^\circ$	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table 5.4: True Positive Rate (TPR) of the authentication process under image shifting distortion with different shift values.

Blk Size	Strgth	Theoretical probability of false positive (P^*_{FP})									
		10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
8×8	[2,2]	1.0	1.0	0.9988	0.9962	0.9875	0.9712	0.9463	0.6575	0.0625	0
	[3,3]	1.0	1.0	0.9988	0.9975	0.9888	0.9750	0.9513	0.7163	0.0825	0.0013
	[4,4]	1.0	1.0	0.9988	0.9975	0.9900	0.9738	0.9437	0.6725	0.0700	0.0013
	[5,5]	1.0	1.0	1.0	0.9988	0.9900	0.9750	0.9487	0.6775	0.0600	0.0013
16×16	[2,2]	1.0	1.0	0.9988	0.9988	0.9975	0.9875	0.8812	0.3275	0.0288	0.0013
	[3,3]	1.0	1.0	1.0	0.9988	0.9975	0.9900	0.8925	0.3463	0.0300	0.0013
	[4,4]	1.0	1.0	1.0	0.9988	0.9988	0.9912	0.8825	0.3113	0.0312	0
	[5,5]	1.0	1.0	1.0	1.0	1.0	0.9988	0.8825	0.3275	0.0437	0.0088
32×32	[2,2]	1.0	1.0	1.0	1.0	1.0	1.0	0.9962	0.9825	0.9437	0.7925
	[3,3]	1.0	1.0	1.0	1.0	1.0	1.0	0.9988	0.9938	0.9475	0.8300
	[4,4]	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9962	0.9762	0.8850
	[5,5]	1.0	1.0	1.0	1.0	1.0	1.0	0.9988	0.9950	0.9838	0.9213

tests, the most common and widely adopted manipulations, namely: JPEG compression, mean filtering, and Additive White Gaussian Noise (AWGN) were used. The TPR values are obtained in the same manner as the geometric distortions explained above, and the results are reported in Table 5.5.

As it can be noted, in the case of JPEG compression, the identification process did not detect all the forged images, except for the two cases where the block size was 32×32, the compression quality was equal to 10 and the P^*_{FP} values were set to 10^{-1} and 10^{-2}. These results for the JPEG compression were expected, since this distortion was adopted as the tolerated manipulation in this work. In other words, by using the JPEG compression even with a quality of 80, a sort of confusion is introduced to the authentication process, which affects the correct detection of forged images. In the case of mean filtering, the best performances were recorded for a block size of 32×32, for which the authentication process succeeded in detecting all filtered images using a filter of size 7×7, and this for P^*_{FP} values of 10^{-5} and higher. Concerning the other block sizes, 16×16 has slight superiority over the 8×8 block size. In the case of the AWGN, similar results as those of mean filtering were obtained, and the best performance was also provided by setting the block size to 32×32. It is worth mentioning that the results reported in this subsection were obtained by considering JPEG compression with a quality of 80 as a tolerated manipulation. Obviously, if another manipulation (or no manipulation) was used, different performances were obtained.

Table 5.5: True Positive Rate (TPR) of the authentication process under image manipulations.

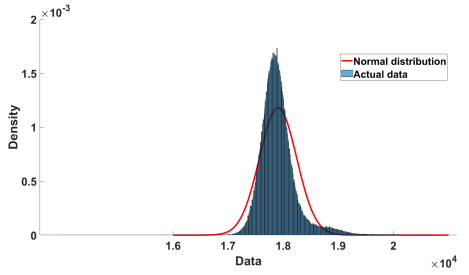
Blk Size	Dist	Theoretical probability of false positive (P^*_{FP})									
		10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}
8×8	JPEG(30)	0.9587	0.8200	0.6062	0.3563	0.1187	0.0375	0.0125	0	0	0
	JPEG(10)	0.9988	0.9962	0.9862	0.9625	0.9163	0.8400	0.7725	0.6587	0.5425	0.4075
	M Filter(3)	0.9988	0.9663	0.8187	0.5425	0.1737	0.0150	0	0	0	0
	M Filter(7)	1.0	1.0	0.9988	0.9925	0.9413	0.7063	0.1388	0.0050	0	0
	AWGN(25dB)	0.9500	0.7850	0.5725	0.2938	0.1000	0.0163	0.0025	0.0013	0.0013	0
	AWGN(15dB)	1.0	1.0	1.0	0.9838	0.8950	0.7150	0.4300	0.1138	0.0163	0.0037
16×16	JPEG(30)	0.9413	0.6675	0.3538	0.0800	0	0	0	0	0	0
	JPEG(10)	0.9988	0.9938	0.9413	0.8063	0.5475	0.2437	0.0.550	0.0100	0.0025	0
	M Filter(3)	1.0	0.9463	0.7000	0.3525	0.0537	0.0013	0	0	0	0
	M Filter(7)	1.0	1.0	1.0	0.9975	0.9838	0.7900	0.1737	0.0050	0	0
	AWGN(25dB)	0.9587	0.7750	0.5050	0.2025	0.0338	0.0013	0.0013	0	0	0
	AWGN(15dB)	1.0	1.0	0.9988	0.9375	0.7362	0.3563	0.0675	0.0088	0.0013	0
32×32	JPEG(30)	0.9800	0.8675	0.6038	0.3300	0.1475	0.0275	0.0037	0	0	0
	JPEG(10)	1.0	1.0	0.9950	0.9688	0.9075	0.7575	0.5325	0.3100	0.1550	0.0575
	M Filter(3)	1.0	0.9950	0.9600	0.8063	0.5350	0.2625	0.1150	0.0413	0.0100	0.0037
	M Filter(7)	1.0	1.0	1.0	1.0	1.0	0.9900	0.9325	0.8063	0.5650	0.3025
	AWGN(25dB)	0.9938	0.9225	0.7550	0.5050	0.2950	0.1175	0.0475	0.0187	0.0013	0.0013
	AWGN(15dB)	1.0	1.0	1.0	1.0	0.9800	0.9325	0.7975	0.6012	0.4088	0.2125

5.3.3 Identification Performance

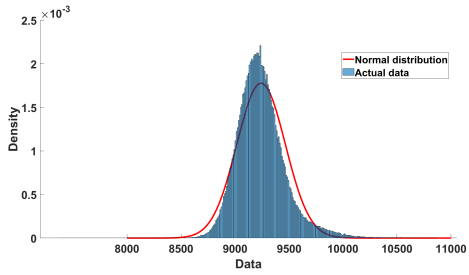
In this subsection, a set of experiments were carried out to gauge the performance of the identification process. First, the statistical distribution of the identification distances was examined. Then, the ability of the identification process to differentiate between fingerprint images, even those of the same finger, was assessed. Finally, the robustness of the identification process to common manipulations was evaluated.

- **A. Identification distance analysis**

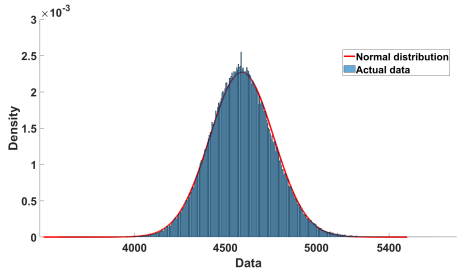
As for the authentication distances, it is also assumed that the identification distances between hashes of visually different images follow a Normal distribution, and this test has been carried out to validate this assumption. To achieve this, we have extracted the hashes of the 800 images from the test database, for different block sizes (i.e. 8×8, 16×16, and 32×32), and computed the identification distances between the extracted hashes, which generates a set of $(800 \times (800-1))/2 = 319600$ identification distances per block size. These sets have also been examined using Q-Q plots. The distribution of the obtained identification distances sets and their corresponding Q-Q plots are shown in Fig. 5.8 and Fig. 5.9, respectively.



(a) Block size: 8×8



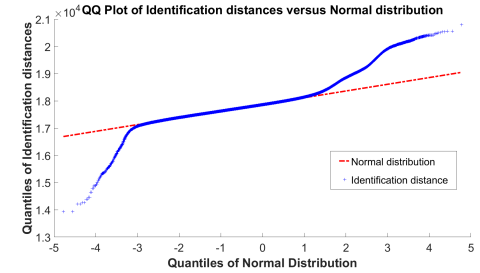
(b) Block size: 16×16



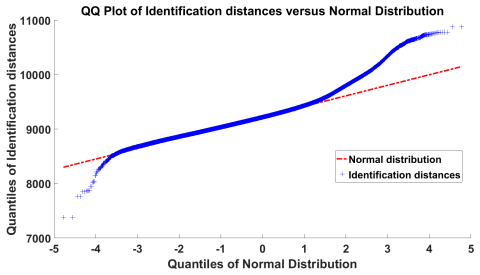
(c) Block size: 32×32

Figure 5.8: Identification distances histograms for different block sizes.

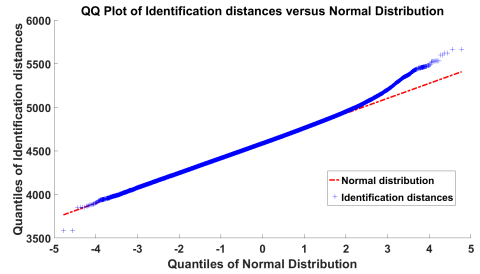
From Fig. 5.8, almost the same observations can be made as for the authentication



(a) Block size: 8×8



(b) Block size: 16×16



(c) Block size: 32×32

Figure 5.9: Quantile-Quantile plots of the identification distances for different block sizes.

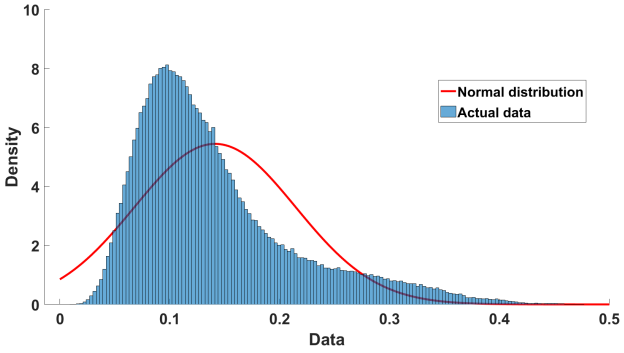
distances. In the case of a block size of 32×32 (Fig. 5.9c), a small set of values diverge from the reference line (i.e., the red line). This set of divergent values is larger for block size 16×16 (Fig. 5.9b) and becomes much larger for blocks of size 8×8 (Fig. 5.9a). In conclusion, the identification distances can be modeled with the Normal law, in order to calculate the identification thresholds T_{id}.
For the sake of illustration, we have also analyzed the statistical distribution of the test schemes in order to check if their distances between compared hashes follows a Normal distribution. To achieve this, the same experiment described above has been carried out for each test scheme and the obtained results are given in Fig.5.10. As it can be observed, except the SVD-NRLBP scheme, the other schemes do not follow a Normal distribution, and hence, their thresholds cannot be estimated as proposed in our system. In fact, for these schemes, the identification thresholds are empirically estimated, which is not accurate and increases the computational complexity.

- **B. Identification accuracy**

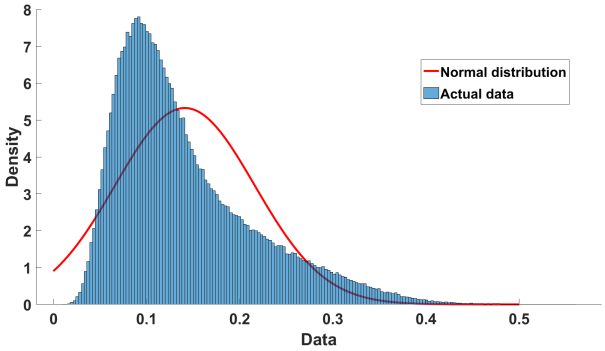
As explained above, the identification process should differentiate between fingerprint images, even they are acquired from the same finger. Therefore, the aim of this test is to evaluate the ability of the identification process to distinguish between different fingerprint images by measuring the FPR, which in this case represents the rate of declaring different images as similar, and this in the absence of any manipulation. To perform this test, the identification thresholds T_{id} for different values of the P^*_{FP} are first calculated as defined by Eq. 5.26, and this for each block size. Then, the sets of identification distances obtained in the previous test are compared with the corresponding threshold, and each time an identification is positive, it is considered as an identification error (i.e., +1 false positive). The results obtained are reported in Table 5.6, and clearly show that, in all cases, the identification process has made errors in detecting the dissimilarities between different fingerprint images, especially for block sizes 8×8 and 16×16, where their FPR values were in the order of 10^{-4}. The best performance has been obtained for a block size of 32×32, which generated the lowest FPR values. These results confirm that designing a perceptual hashing technique that identifies similarities between biometric images is a challenging task that requires particular attention.

- **C. Identification robustness**

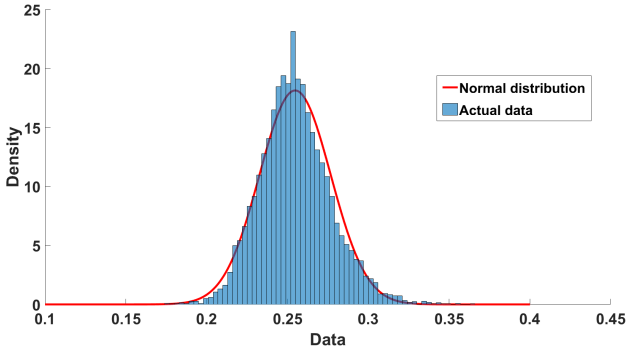
In order to assess the robustness of the identification process against malicious distortions, the Receiver Operating Characteristics (ROC) curve was used. This curve represents the variation of the TPR (i.e., correct detection of similar image content) versus the FPR (i.e., false detection of similar image content). To obtain the ROC curve, we set the block size to 32×32 and proceeded as follows: first, the identification thresholds T_{id} were calculated, by varying the P^*_{FP} between 10^{-1} and 10^{-10}. Then, we took an image from the set of test images and applied a series of distortions



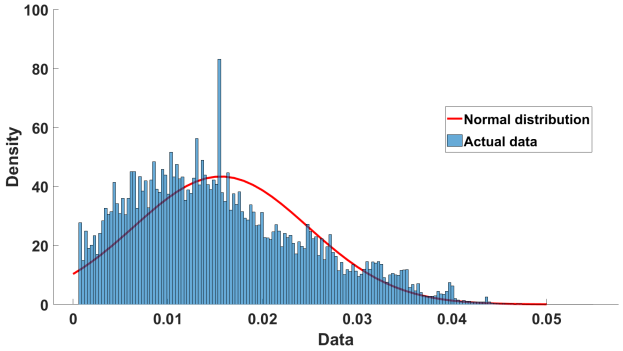
(a) LBP [142]



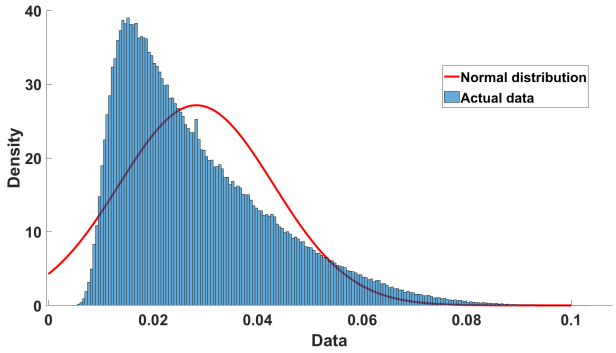
(b) NRLBP [143]



(c) SVD-NRLBP [144]



(d) SVD-CSLBP [145]



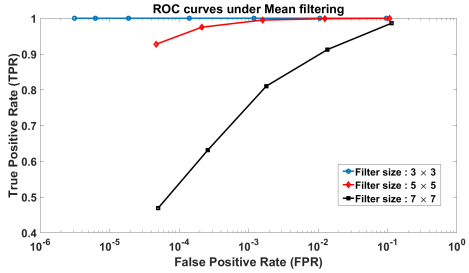
(e) DCT-NRLBP [146]

Figure 5.10: Identification distance histograms of the compared schemes.

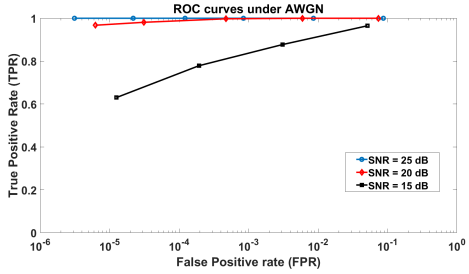
Table 5.6: False Positive Rate (FPR) in the identification process in the absence of distortions.

P^*_{FP}	**Block size**		
	8×8	16×16	32×32
10^{-1}	4.9×10^{-2}	9.6×10^{-2}	9.9×10^{-2}
10^{-2}	1.8×10^{-3}	1.0×10^{-2}	1.1×10^{-2}
10^{-3}	6.2×10^{-4}	2.1×10^{-3}	1.5×10^{-3}
10^{-4}	4.3×10^{-4}	9.6×10^{-4}	2.4×10^{-4}
10^{-5}	3.7×10^{-4}	6.9×10^{-4}	6.2×10^{-5}
10^{-6}	2.9×10^{-4}	5.1×10^{-4}	2.0×10^{-5}
10^{-7}	2.3×10^{-4}	4.2×10^{-4}	1.2×10^{-5}
10^{-8}	1.9×10^{-4}	3.6×10^{-4}	3.1×10^{-6}
10^{-9}	1.6×10^{-4}	3.0×10^{-4}	3.1×10^{-6}
10^{-10}	1.3×10^{-4}	2.7×10^{-4}	3.1×10^{-6}

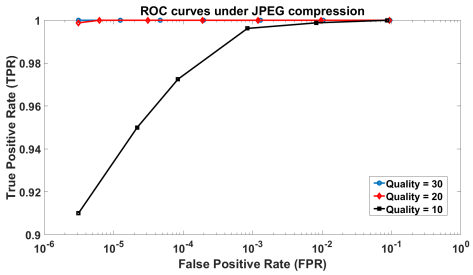
on it (i.e., mean filtering with different filter sizes, adding white noise with different strength, and JPEG compression with different qualities); this resulted in a set of altered images. After that, the hashes of the original and altered images were extracted. Then, the identification distances between these images were calculated and compared with the identification thresholds to obtain the TPR values. In the meantime, the identification distances between the hash of the original image with the altered images of the other test images were calculated and compared with the identification thresholds to obtain the FPR. Finally, the ROC curves were constructed for each type of attack using TPR against FPR. These curves are given in Fig. 5.11.



(a) Mean filtering



(b) Additive White Gaussian Noise (AWGN)

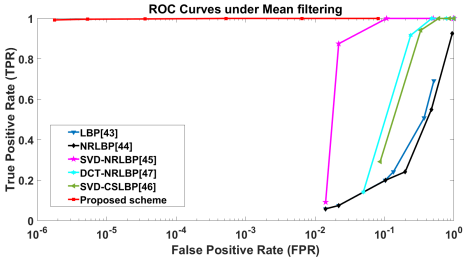


(c) JPEG compression

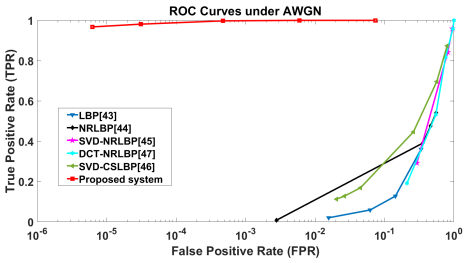
Figure 5.11: Robustness under different kinds of attacks.

With regard to mean filtering, the obtained results (Fig.5.11a) clearly show that the identification error rate is insignificant for the filter size of 3×3, and very small for the filter size of 5×5. However, by increasing the filter size to 7×7, this error becomes more important (around 0.5 for a $P^*_{FP} = 10^{-10}$). For the AWGN, it is observed from Fig.5.11b that the identification process is very robust, and the error rate is less than 0.4 for a $P^*_{FP} = 10^{-10}$, in the case of a very strong attack ($SNR = 15dB$). Regarding JPEG compression, the process is very robust and produces an insignificant error rate for quality values of 30 and 20. This rate tends to increase slightly and

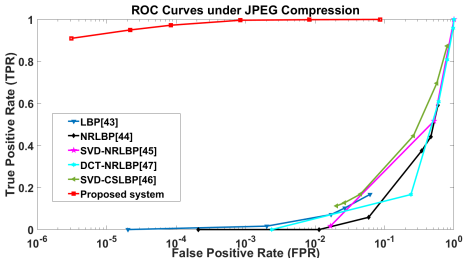
reaches a value less than 0.1 for a $P^*_{FP} = 10^{-10}$, with a compression quality equal to 10. Note that mean filtering with a filter size of 7×7, adding white Gaussian noise with $SNR = 15$ and compression with a quality equal to 10 are considered as very strong attacks and cause considerable degradation to the images. Overall, one can conclude that the identification process provides excellent results in the presence of noise addition and compression attacks; and produces acceptable results against filtering attack with an averaging filter.



(a) Mean filtering (3×3)



(b) Additive White Gaussian Noise (SNR = 20dB)



(c) JPEG compression (Quality = 10)

Figure 5.12: Performance comparison using ROC curves.

For comparison purposes, the robustness performance of our identification process was compared to the five test schemes, namely: the LBP scheme [142], the NRLBP scheme [143], the SVD-NRLBP [144], the SVD-CSLBP [145], and the DCT-NRLBP scheme [146]. The experiment was carried out as described above, and for all tested schemes, the parameters have been set in such a way as to yield the best performance. The ROC curves, illustrated in 5.12, were used to show the robustness of these schemes against mean filtering, AWGN and JPEG compression. As one can observe, our identification scheme provides the best robustness performance for all distortions, and hence clearly outperforms the other schemes.

Furthermore, the Area Under Curve (AUC) values have been computed for the tested systems from the ROC curves provided previously. the obtained results are given in Table 5.7. Note that, when an AUC value is equal to $= 0.5$, the scheme is randomly

Table 5.7: AUC performance comparison

Scheme	AUC (%)		
	Mean filtering	AWGN	JPEG compression
LBP [142]	59,98	50,11	63,79
NRLBP [143]	44,33	49,69	37,74
SVD-NRLBP [144]	92,74	50,28	43,21
SVD-CSLBP [145]	81,13	58,68	60,20
DCT-NRLBP [146]	85,16	47,84	50.19
Proposed	**96,40**	**95,75**	**86,50**

classifying the images. Therefore, the higher the *AUC* is, the better the performance of the scheme at distinguishing between perceptually similar and different images. As can be noted, the obtained values of AUC confirm the superiority of the proposed identification technique over the other tested schemes for all assessed attacks, particularly for the AWGN attack.

5.4 Conclusion

In this chapter, we presented a novel Replay Attack Detection scheme that Leverages the Challenge/Response principle and combines perceptual image hashing and data hiding techniques. The scheme effectively detects replay attacks through a two-step verification process. First of all, an authentication process ensures that the biometric image has not been forged. Then, an identification process checks if the received image has not been re-submitted before, by comparing it with a database of perceptual hashes of previously received images. We conducted an analysis of various attack scenarios and demonstrated that our proposed scheme efficiently detects replay attacks, even in less probable scenarios.

Additionally, we introduced a new perceptual image hashing system that offers the advantage of serving both image authentication and identification using the same hash sequence. The system is based on a shift-based signal calibration technique that calculates the normalization shifts of the DCT and DST coefficients of the DBLM values to generate the perceptual hash. Experimental results have confirmed the high performance of our proposed system for both image authentication and identification tasks.

In terms of authentication performance, our scheme effectively detects minute malicious manipulations, particularly image rotation and translation, which are commonly employed in replay attacks. As for identification performance, our system robustly resists image processing operations such as mean filtering, noise addition, and JPEG compression. Furthermore, our experiments have demonstrated that our proposed system outperforms other state-of-the-art image hashing techniques.

Chapter 6

General Conclusion and Perspectives

Biometric systems have emerged as a safer and more reliable means of identifying and authenticating individuals, finding applications in confidential and critical domains such as biometric identity document management, e-commerce, criminal identification, and access control to sensitive areas. These systems have demonstrated clear superiority over traditional identification/authentication systems. However, despite their advantages, biometric systems are still facing security vulnerabilities that can be exploited by attackers.

The objective of this research work is to propose a solution to secure biometric systems against replay attacks, where an attacker intercepts legitimate users' fingerprint images and replays them to gain unauthorized access. To address this challenge, we developed a solution inspired by the challenge/response method and incorporated perceptual hashing techniques. Perceptual hashing served as a means of verifying the image source and ensuring that the images were not replayed.

To achieve our objective, we have first provided an overview of biometric systems, focusing on their security aspects. We have then presented an overview of perceptual hashing techniques, surveying the literature and selecting the most suitable technique to counter replay attacks. We subsequently explained the stages of a replay attack on a biometric system and detailed our proposed solution based on perceptual hashing.

The chosen perceptual hashing technique was originally designed for video content identification and authentication. We adapted and applied this technique to fingerprint images, making modifications where necessary. We conducted a series of tests to analyze the statistical distribution of the generated hashes, determine optimal parameters such as block size, and evaluate the technique's robustness against various manipulations, including filtering, white noise addition, and JPEG compression.

The results of our experiments demonstrated that the values of the generated hashes exhibited a uniform distribution, minimizing the likelihood of two different hashes appearing similar by chance. We also found that the identification distances could be modeled by the Normal distribution, allowing for more reliable identification thresh-

olds. Moreover, splitting images into 32 x 32 blocks yielded the lowest false positive error rates. In terms of robustness against different attacks, the proposed technique showed excellent performance against noise addition and JPEG compression, and satisfactory performance against filtering using an averaging filter.

In conclusion, this research work has contributed to securing biometric systems against replay attacks by leveraging perceptual hashing techniques. The proposed solution has demonstrated promising results in terms of identification/authentication accuracy and robustness against various attacks. By addressing the security vulnerabilities of biometric systems, we aim to enhance their reliability and facilitate their deployment in critical applications.

As a future direction, we propose some perspectives for further research and development in the field of securing biometric systems using perceptual hashing techniques as follows:

1. **Enhancing resistance to advanced attacks**: While the proposed solution has shown effectiveness against various attacks, including replay attacks, there is still work for further improvement in terms of robustness against advanced attack strategies. Future research can focus on developing more sophisticated perceptual hashing techniques that can withstand advanced manipulations such as morphing, spoofing, and deepfake attacks.
2. **Multi-modal biometric systems**: Many real-world applications require the integration of multiple biometric modalities to enhance security and accuracy. Future work can explore the integration of perceptual hashing techniques with multi-modal biometric systems, such as combining fingerprints with other biometric traits like iris, face, or voice. This can provide a more comprehensive and robust approach to biometric system security.
3. **Adaptive and dynamic techniques**: The effectiveness of perceptual hashing techniques can be further improved by incorporating adaptive and dynamic mechanisms. Research can focus on developing algorithms that can adaptively adjust the hashing parameters based on the specific characteristics of the biometric data, environmental conditions, and potential attack scenarios. This would enhance the resilience of the system against evolving threats.
4. **Privacy-preserving techniques**: As biometric data become more pervasive, ensuring privacy protection becomes crucial. Future research can explore the integration of privacy-preserving techniques with perceptual hashing to enable secure and privacy-enhanced biometric systems. Techniques such as secure multi-party computation, homomorphic encryption, and differential privacy can be investigated to protect sensitive biometric information during the hashing and matching processes.

5. **Real-time implementation and scalability**: To deploy perceptual hashing techniques in practical biometric systems, real-time implementation and scalability are essential factors to consider. Future work can focus on optimizing the computation efficiency of perceptual hashing algorithms to enable real-time processing of large-scale biometric datasets. Additionally, research can explore distributed and parallel computing techniques to handle the computational demands of large-scale biometric systems.
6. **Standardization and interoperability**: Standardization plays a crucial role in ensuring compatibility and interoperability among different biometric systems. Future efforts can be directed towards standardizing perceptual hashing techniques, metrics, and evaluation methodologies. This would facilitate the adoption and integration of secure biometric systems across different platforms, applications, and organizations.
7. **Combining cryptography an PIH :** The integration of cryptography with perceptual hashing can offer valuable contributions to research in multimedia security and content identification. Combining these techniques can enhance the security and authenticity of multimedia content by using cryptographic primitives to secure the perceptual hash values, ensuring their integrity and making it harder for attackers to forge or tamper with the content. This approach can also enable privacy-preserving content identification and secure content retrieval and indexing, as well as develop more robust and tamper-resistant media forensic systems that can detect and analyze media manipulations and forgeries. The synergy between cryptography and perceptual hashing can lead to innovative solutions that address the evolving challenges in the digital media landscape.

By addressing these perspectives, researchers can contribute to the ongoing advancements in securing biometric systems, enhancing their reliability, accuracy, and resistance to attacks.

Bibliography

[1] International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC). Information technology — biometric performance testing and reporting — part 1: Principles and framework. International Standard, 05 2021. Technical Committee: ISO/IEC JTC 1/SC 37.

[2] M. Joshi, B. Mazumdar, and S. Dey. Security vulnerabilities against fingerprint biometric system. *CoRR*, abs/1805.07116, 2018. Mon, 13 Aug 2018 16:48:42 +0200.

[3] N. K. Ratha, J. H. Connell, and R. M. Bolle. Enhancing security and privacy in biometrics-based authentication systems. *IBM Systems Journal*, 40:614–626, 2001.

[4] A. Adler, A. K. Jain, P. Flynn, and A. Ross. *Biometric System Security*, pages 381–402. Springer US, Boston, MA, 2008.

[5] A.K. Jain, A. Ross, and A. Nagar. Biometric template security. *EURASIP J. Adv. Signal Process.*, 2008:113:1–113:17, 2008.

[6] B. Cukic and N. Bartlow. The vulnerabilities of biometric systems – an integrated look at old and new ideas. Technical report, West Virginia University, 2005.

[7] S. Prungsinchai. *Robust and Secure Perceptual Image Hashing in the Transform Domain*. PhD thesis, Northumbria University, 2014.

[8] Ling D., Anthony T.S. H., and Runmin C. Perceptual hashing for image authentication:a survey on presentation. *Signal Processing: Image Communication*, 81:115713, 2020.

[9] Y. Li Liu, W. Yan, and B. Hu. Resistance to facial recognition payment in china: The influence of privacy-related factors. *Telecommun. Policy*, 45(5):102155, 2021.

[10] P. Singh. Aadhaar and data privacy: Biometric identification and anxieties of recognition in india. *Inf. Commun. Soc.*, 24(7):978–993, 2021.

[11] K. Kruglov. Biometric data processing and storage system threats. *Malware Reports*, 2019.

[12] N. K. Ratha, J. H. Connell, and R. M. Bolle. An analysis of minutiae matching strength. In *Third International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA)*, pages 223–228, 2001.

[13] Z. Rui and Z. Yan. A survey on biometric authentication: Toward secure and privacy-preserving identification. *IEEE Access*, 7:5994–6009, 2019.

[14] N. K. Ratha, J. H. Connell, and R. M. Bolle. Enhancing security and privacy in biometrics-based authentication systems. *IBM Syst. J.*, 40(3):614–634, 2001.

[15] C. B. Tan, M. H. A. Hijazi, N. Khamis, N. E. b N. Puteri, Z. Zuraini, C. Frans, and G. Abdullah. A survey on presentation attack detection for automatic speaker verification systems: State-of-the-art, taxonomy, issues and future direction. *Multimed. Tools Appl.*, 80(21):32725–32762, 2021.

[16] M. K. Rusia and D. K. Singh. A comprehensive survey on techniques to handle face identity threats: challenges and opportunities. *Multimed Tools Appl*, 82(2):1669–1748, 2023. PMID: 35702682.

[17] D. F. Smith. *Countering Digital Replay Attacks for Face Verification on Consumer Smart Devices using Structured Illumination*. PhD thesis, The University of Queensland, 2016.

[18] Y. Ren, Z. Fang, D. Liu, and C. Chen. Replay attack detection based on distortion by loudspeaker for voice authentication. *Multimedia Tools and Applications*, 78:8383–8396, 2019.

[19] H. A. Patil and M. R. Kamble. A survey on replay attack detection for automatic speaker verification (asv) system. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1047–1053, 2018.

[20] R. M. Bolle, J. H. Connell, and N. K. Ratha. Biometric perils and patches. *Pattern Recognition*, 35(12):2727–2738, 2002.

[21] R. W. Frischholz and A. Werner. Avoiding replay-attacks in a face recognition system using head-pose estimation. In *IEEE International SOI Conference. Proceedings (Cat. No.03CH37443)*, pages 234–235, 2003.

[22] M. De Marsico, M. Nappi, D. Riccio, and J. L. Dugelay. Moving face spoofing detection via 3d projective invariants. In *5th IAPR International Conference on Biometrics (ICB)*, pages 73–78, 2012.

[23] A. Anjos, J. Komulainen, S. Marcel, A. Hadid, and M. Pietikainen. Face anti-spoofing: Visual approach. In *Handbook of Biometric Anti-Spoofing*, Advances in Computer Vision and Pattern Recognition, pages 65–82. Springer, 2014.

[24] I. Sluganovic, M. Roeschlin, K. B. Rasmussen, and I. Martinovic. Analysis of reflexive eye movements for fast replay-resistant biometric authentication. *ACM Transactions on Privacy and Security*, 22(1):1–30, 2019.

[25] A. Czajka, A. Refaey, and A. Pacut. Replay attack prevention for iris biometrics. In *42nd Annual IEEE International Carnahan Conference on Security Technology*, pages 247–253, 2008.

[26] K. Loukhaoukha, A. Refaey, K. Zebbiche, and A. Shami. Efficient and secure cryptosystem for fingerprint images in wavelet domain. *Multimedia Tools and Applications*, 77:9325–9339, 2018.

[27] D. Maltoni, D. Maio, A. J. Jain, and S. Prabhakar. Securing fingerprint systems. In *Handbook of Fingerprint Recognition*, pages XXIV, 522. Springer Cham, 3 edition, 2022.

[28] V. B. Joshi, M. S. Raval, D. Gupta, P. P. Rege, and S. K. Parulkar. A multiple reversible watermarking technique for fingerprint authentication. *Multimedia Systems*, 22:367–378, 2016.

[29] R. Gupta and P. Sehgal. Optimized singular value decomposition (svd) based watermarking of iris biometric data to mitigate replay attack. In *ICT for Competitive Strategies*, pages 371–416. CRC Press, 2020.

[30] X. Yuan and Y. Zhao. Perceptual image hashing based on three-dimensional global features and image energy. *IEEE Access*, 9:49325–49337, 2021.

[31] M. Schneider and Shih-Fu C. A robust content based digital signature for image authentication. In *Proceedings of 3rd IEEE International Conference on Image Processing*, volume 3, pages 227–230, 1996.

[32] C. Y. Lin and S. F. Chang. A robust image authentication method distinguishing jpeg compression from malicious manipulation. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(2):153–168, 2001.

[33] R. Muthu, A. Bouridane, and F. Khelifi. Minutiae based fingerprint image hashing. In *2014 International Conference on Control, Decision and Information Technologies (CoDIT)*, pages 696–700, 2014.

[34] R. Muthu and C. Rani. Perceptual hashing for efficient fingerprint based identification. In *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pages 1–6, 2017.

[35] S. M. Abdullahi, H. Wang, and A. Malik. Fingerprint image hashing based on minutiae points and shape context. *International Journal of Digital Crime and Forensics*, 10(4):1–20, 2018.

[36] W. Birouk, A. Lahoulou, A. Melit, and A. Bouridane. Robust perceptual fingerprint image hashing: A comparative study. *International Journal of Biometrics*, 15(1):59–77, 2023.

[37] E. Ghouti. A new perceptual video fingerprinting system. *Multimedia Tools and Applications*, 77(23):6713–6751, 2018.

[38] F. Khelifi, A. Bouridane, and A. Wang. Perceptual video hashing for content identification and authentication. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(1):50–67, 2019.

[39] E. Boutellaa, Z. Boulkenafet, J. Komulainen, and A. Hadid. Audiovisual synchrony assessment for replay attack detection in talking face biometrics. *Multimedia Tools and Applications*, 75(9):5329–5343, 05 2016.

[40] Q. Gui, W. Yang, Z. Jin, Mariella V. Ruiz-B., and S. Laszlo. A residual feature-based replay attack detection approach for brainprint biometric systems. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6, 2016.

[41] M. S. Saranya and H. A. Murthy. Decision-level feature switching as a paradigm for replay attack detection. In *Interspeech 2018*, pages 686–690, 2018.

[42] Andre R. Avila, Jahangir Alam, Douglas O'Shaughnessy, and Tiago H. Falk. Blind channel response estimation for replay attack detection. In *Interspeech 2019*, pages 2893–2897, 2019.

[43] Y. Wang, Y. Liu, P. Gao, and Y. Wang. Icdsp 2021: 5th international conference on digital signal processing. In *Proceedings of the 5th International Conference on Digital Signal Processing (ICDSP 2021)*, pages 266–271, 2021.

[44] D. Karmakar, R. Sarkar, and M. Datta. Spoofed replay attack detection by multidimensional fourier transform on facial micro-expression regions. *Signal Processing: Image Communication*, 93:116164, 2021.

[45] K. P. Bharath and Rajesh K. M. New replay attack detection using iterative adaptive inverse filtering and high frequency band. *Expert Systems with Applications*, 195:116597, 2022.

[46] M. Dua, A. Sadhu, A. Jindal, and R. Mehta. A hybrid noise robust model for multireplay attack detection in automatic speaker verification systems. *Biomedical Signal Processing and Control*, 74:103517, 2022.

[47] A. K. Jain and A. Ross. *Introduction to Biometrics*. Springer US, Boston, MA, 2008.

[48] A. K. Jain, P. Flynn, and A. Ross, editors. *Handbook of Biometrics*. Springer New York, NY, 1 edition, 2008.

[49] Citizen conference report on biometrics and security. Conference Report, 2006.

[50] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar. *Handbook of Fingerprint Recognition*. 2nd edition, 2007.

[51] S. Guennouni, A. Mansouri, and A. Ahaitouf. *Biometric Systems and Their Applications*. IntechOpen, 2020.

[52] S. Prabhakar, S. Pankanti, and A. K. Jain. Biometric recognition: Security and privacy concerns. *IEEE Security & Privacy*, 1(2):33–42, 2003.

[53] M. El Abed. *Evaluation of Biometric Systems*. PhD thesis, Rijeka, 2012.

[54] M. El-Abed, C. Charrier, and C. Rosenberger. Evaluation of biometric systems. In *New Trends and Developments in Biometrics*. InTech, 2012.

[55] P. Bonazza. *Système de sécurité biométrique multimodal par imagerie, dédié au contrôle d'accès*. Doctoral dissertation, Université de Bourgogne, 2019.

[56] R. O. Belguechi. *Sécurité des systèmes biométriques : révocabilité et protection de la vie privée*. Ph.d. thesis, Ecole nationale Supérieure en Informatique Alger, 2015.

[57] J. Richiardi and K. Kryszczuk. Biometric systems evaluation. In Henk C. A. van Tilborg and Sushil Jajodia, editors, *Encyclopedia of Cryptography and Security*, pages 117–124. Springer US, Boston, MA, 2011.

[58] J. T. R. Pinto. *Seamless Multimodal Biometrics for Continuous Personalised Wellbeing Monitoring*. PhD thesis, Universidade of Porto, December 2022.

[59] J. Wayman, A. Jain, D. Maltoni, and D. Maio, editors. *Biometric Systems: Technology, Design and Performance Evaluation*. Springer London, 2005.

[60] I. Benchennane. *Etude et mise au point d'un procédé biométrique multimodal pour la reconnaissance des individus*. PhD thesis, University of Sciences and Technology in Oran, 2015.

[61] Iso/iec 15408: Security techniques - evaluation criteria for it security. Technical report, 2009. Cité p. 27.

[62] Information technology - security techniques - security evaluation of biometrics. Final Committee Draft ISO/IEC FCD 19792, International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC), 2008.

[63] J. N. Pato and L. I. Millett, editors. *Biometric Recognition: Challenges and Opportunities*. The National Academies Press, Washington, DC, 2010.

[64] M. Faundez-Zanuy. Biometric recognition: why not massively adopted yet? *IEEE Aerospace and Electronic Systems Magazine*, 20(8):25–28, 2005.

[65] C. Roberts. Biometric attack vectors and defences. *Computers & Security*, 26(1):14–25, 2007.

[66] J. L. Wayman. *Technical Testing and Evaluation of Biometric Identification Devices*, pages 345–368. Springer US, Boston, MA, 1996.

[67] M. El-Abed, P. Lacharme, and C. Rosenberger. Security evabio: An analysis tool for the security evaluation of biometric authentication systems. In *5th IAPR International Conference on Biometrics, ICB 2012*, pages 460–465, New Delhi, India, 2012.

[68] A. Adler and S. Schuckers. Biometric vulnerabilities, overview. In S. Z. Li and A. Jain, editors, *Encyclopedia of Biometrics*. Springer, Boston, MA, 2009.

[69] C. Lee, J. Y. Choi, K. A. Toh, and S. Lee. Alignment-free cancelable fingerprint templates based on local minutiae information. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 37(4):980–992, 2007.

[70] V. Patel, N. K. Ratha, and R. Chellappa. Cancellable biometrics: A review. *IEEE Signal Processing Magazine*, 32:54–65, 2015.

[71] D. Cappelli, R.and Maio and D. Maltoni. Modelling plastic distortion in fingerprint images. In Sameer Singh, Nabeel Murshed, and Walter Kropatsch, editors, *Advances in Pattern Recognition — ICAPR 2001*, pages 371–378, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[72] T. Matsumoto, H. Matsumoto, K. Yamada, and S. Hoshino. Impact of artificial gummy fingers on fingerprint systems. In *Proceedings of SPIE, Optical Security and Counterfeit Deterrence Techniques IV*, volume 4677, pages 275–289, 2002.

[73] D. Baldissera, A. Franco, D. Maio, and D. Maltoni. Fake fingerprint detection by odor analysis. In *International Conference on Biometric Authentication (ICBA06)*, 2006.

[74] S. Pankanti and M. Yeung. Verification watermarks on fingerprint recognition and retrieval. In *Security and Watermarking of Multimedia Contents*, 1999.

[75] N. K. Ratha, J. H. Connell, and R. M. Bolle. An analysis of minutiae matching strength. In *Audio and Video-Based Biometric Person Authentication*, 2001.

[76] S. Windarta, S. Suryadi, K. Ramli, B. Pranggono, and T. S. Gunawan. Lightweight cryptographic hash functions: Design trends, comparative study, and future directions. *IEEE Access*, 10:82272–82294, 2022.

[77] Y. Wang. New way to construct cryptographic hash function, February 2014.

[78] S. Vrochidis, B. Huet, E. Chang, I. Kompatsiaris, L. Weng, I.-H. Jhuo, and W.-H. Cheng. Perceptual hashing for large-scale multimedia search. In S. Vrochidis, B. Huet, E. Chang, and I. Kompatsiaris, editors, *Big Data Analytics for Large-Scale Multimedia Search*. 2019.

[79] L. Struppek, D. Hintersdorf, D. Neider, and K. Kersting. Learning to break deep perceptual hashing: The use case NeuralHash. In *2022 ACM Conference on Fairness, Accountability, and Transparency*. ACM, June 2022.

[80] S. Jain, A.M. Cretu, and Y.A. de Montjoye. Adversarial detection avoidance attacks: Evaluating the robustness of perceptual hashing-based client-side scanning. In *31st USENIX Security Symposium (USENIX Security 22), 2022*, 2022.

[81] C. Zauner. Implementation and benchmarking of perceptual image hash functions. July 2010. University of Applied Sciences, Hagenberg.

[82] Q. Hao, L. Luo, S. T. K. Jan, and G. Wang. It's not what it looks like: Manipulating perceptual hashing based applications. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS '21)*, page 17, New York, NY, USA, November 2021. ACM.

[83] C. Shorten and T. M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(60):1–48, 2019.

[84] S. Xiang, H. Kim, and J. Huang. Histogram-based image hashing scheme robust against geometric deformations. In *The 9th ACM Multimedia and Security Workshop*, pages 121–128, 2007.

[85] P. Brasnett and M. Bober. Fast and robust image identification. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–5, 2008.

[86] A. Swaminathan, Y. Mao, and M. Wu. Robust and secure image hashing. *IEEE Transactions on Information Forensics and Security*, 1(2):215–230, 2006.

[87] S. Roy and Q. Sun. Robust hash for detecting and localizing image tampering. In *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, volume 6, pages VI–117–VI–120, 2007.

[88] X. Lv and Z. J. Wang. An extended image hashing concept: Content-based fingerprinting using fjlt. *EURASIP Journal on Information Security*, 2009:2:1–2:16, 2009.

[89] A. Hadmi. *Protection of Visual Data: Analysis of Perceptual Hash Functions*. PhD thesis, University of Montpellier II, 2012.

[90] M. Schneider and S. F. Chang. A robust content-based digital signature for image authentication. In *IEEE International Conference on Image Processing (ICIP'96)*, volume 3, pages 227–230, 1996.

[91] R. Venkatesan, S.-M. Koon, M. H. Jakubowski, and P. Moulin. Robust image hashing. In *IEEE International Conference on Image Processing*, pages 664–666, Vancouver, Canada, 2000.

[92] M. Srivastava, J. Siddiqui, and M. A. Ali. Robust image hashing based on statistical features for copy detection. In *The IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics Engineering*, pages 490–495, 2017.

[93] Z. Huang and S. Liu. Robustness and discrimination oriented hashing combining texture and invariant vector distance. In *The 26th ACM International Conference on Multimedia*, pages 1389–1397, 2018.

[94] C. Y. Lin and S. F. Chang. A robust image authentication method distinguishing JPEG compression from malicious manipulation. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(2):153–168, 2001.

[95] F. Liu and L. M. Cheng. Perceptual image hashing via wave atom transform. In Yun Qing Shi, Hyoung-Joong Kim, and Fernando Perez-Gonzalez, editors, *Digital Forensics and Watermarking*, pages 468–478, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[96] C. Qin, C.-H. Chang, and P.-L. Tsou. Robust image hashing using non-uniform sampling in discrete fourier domain. *Digital Signal Processing*, 23(2):578–585, 2013.

[97] V. Monga and B.L. Evans. Robust perceptual image hashing using feature points. In *2004 International Conference on Image Processing, 2004. ICIP '04.*, volume 1, pages 677–680 Vol. 1, 2004.

[98] V. Monga and B. L. Evans. Perceptual image hashing via feature points: Performance evaluation and trade-offs. *IEEE Transactions on Image Processing*, 15(11):3452–3465, 2006.

[99] C. P. Yan, C. M. Pun, and X. C. Yuan. Multi-scale image hashing using adaptive local feature extraction for robust tampering detection. *Signal Processing*, 121:1–16, 2016.

[100] C. Qin, X. Chen, J. Dong, and X. Zhang. Perceptual image hashing with selective sampling for salient structure features. *Displays*, 45:26–37, 2016.

[101] S. Kozat, R. Venkatesan, and M. K. Mihçak. Robust perceptual image hashing via matrix invariants. In *IEEE International Conference on Image Processing (ICIP'04)*, pages 3443–3446, 2004.

[102] V. Monga and M. K. Mihçak. Robust image hashing via non-negative matrix factorizations. In *IEEE International Conference on Acoustic, Speech and Signal Processing*, 2006.

[103] X. Lv and Z. J. Wang. Fast johnson-lindenstrauss transform for robust and secure image hashing. In *International Workshop on Multimedia Signal Processing (MMSP)*, pages 725–729, 2008.

[104] Z. Tang, S. Wang, X. Zhang, W. Wei, and Y. Zhao. Lexicographical framework for image hashing with implementation based on DCT and NMF. *Multimedia Tools and Applications*, 52(2-3):325–345, 2011.

[105] R. Sun and W. Zeng. Secure and robust image hashing via compressive sensing. *Multimedia Tools and Applications*, 70(3):1651–1665, 2014.

[106] H. Liu, D. Xiao, Y. Xiao, and Y. Zhang. Robust image hashing with tampering recovery capability via low-rank and sparse representation. *Multimedia Tools and Applications*, 75(13):7681–7696, 2016.

[107] X. Lv and A. Wang. Compressed binary image hashes based on semi-supervised spectral embedding. *IEEE Transactions on Information Forensics and Security*, 8(11):1838–1849, 2013.

[108] L. Du, Z. Chen, and Y. Ke. Image hashing for tamper detection with multi-view embedding and perceptual saliency. *Advances in Multimedia*, 2018:1–11, 2018.

[109] J. Chen, X. Kang, Y. Liu, and Z. J. Wang. Median filtering forensics based on convolutional neural networks. 22(11):1849–1853, 2015.

[110] Y. Qian, J. Dong, W. Wang, and T. Tan. Deep learning for steganalysis via convolutional neural networks. In *Media Watermarking, Security, and Forensics*, volume 9409, 2015.

[111] B. Bayar and M. C. Stamm. A deep learning approach to universal image manipulation detection using a new convolutional layer. In *The 4th ACM Workshop on Information Hiding and Multimedia Security*, pages 5–10, 2016.

[112] H. Y. Choi, H. U. Jang, D. Kim, J. Son, S. M. Mun, S. Choi, and H. K. Lee. Detecting composite image manipulation based on deep neural networks. In

International Conference on Systems, Signals and Image Processing, pages 1–5, 2017.

[113] L. Weng, I. H. Jhuo, W. H. Cheng, S. Vrochidis, B. Huet, E. Chang, and I. Kompatsiaris. Perceptual hashing for large-scale multimedia search. In *Big Data Analytics for Large-Scale Multimedia Search*, chapter 9, pages 239–265. Wiley-Blackwell, Hoboken, NJ, USA, 2019.

[114] H. Farid. An overview of perceptual hashing. *Journal of Online Trust and Safety*, 2021.

[115] M. Roy, D. M. Thounaojam, and S. Pal. A perceptual hash based blind-watermarking scheme for image authentication. *Expert Systems with Applications*, 227:120237, 2023.

[116] M. Miller, M. Rodriguez, and I. Cox. Audio fingerprinting: nearest neighbor search in high dimensional binary spaces. In *The IEEE Workshop on Multimedia Signal Processing (MMSP)*, pages 182–185, 2002.

[117] C. Grüne. The complexity classes of hamming distance recoverable robust problems. In José A. Soto and Andreas Wiese, editors, *LATIN 2024: Theoretical Informatics*, pages 321–335, Cham, 2024. Springer Nature Switzerland.

[118] S. F. C. Haviana and D. Kurniadi. Average hashing for perceptual image similarity in mobile phone application. *Journal of Telematics and Informatics*, 4(1):12–18, 2016.

[119] V. Popkov. Possible application of perceptual image hashing. Master's thesis, Tallinn University of Technology, 2015.

[120] M. Fei, Z. Ju, X. Zhen, and J. Li. Real-time visual tracking based on improved perceptual hashing. *Multimedia Tools and Applications*, 76:4617–4634, 2017.

[121] A. Drmic, M. Silic, G. Delac, K. Vladimir, and A. S. Kurdija. Evaluating robustness of perceptual image hashing algorithms. In *40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 995–1000, Opatija, Croatia, 2017.

[122] R. Fitas, B. Rocha, V. Costa, and A. Sousa. Design and comparison of image hashing methods: A case study on cork stopper unique identification. *Journal of Imaging*, 7(3):48, 2021.

[123] D. Z. Wang and J. y. Liang. Research and design of theme image crawler based on difference hash algorithm. *IOP Conference Series: Materials Science and Engineering*, 563(042080), 2019.

[124] N. Karunanayake, J. Rajasegaran, A. Gunathillake, S. Seneviratne, and G. Jourjon. A multi-modal neural embeddings approach for detecting mobile counterfeit apps: A case study on google play store. *IEEE Transactions on Mobile Computing*, 21(1):16–30, 2022.

[125] F. Vega, J. Medina, D. Mendoza, V. Saquicela, and M. Espinoza. A robust video identification framework using perceptual image hashing. In *2017 XLIII Latin American Computer Conference (CLEI)*, pages 1–10, Cordoba, Argentina, 2017.

[126] W. C. Huang, F. Di Troia, and M. Stamp. Robust hashing for image-based malware classification. In *The 15th International Joint Conference on e-Business and Telecommunications (ICETE 2018) - Volume 1: DCNET, ICE-B, OPTICS, SIGMAP and WINSYS*, pages 451–459, 2018.

[127] E. Aboufadel and S. Schlicker. Wavelets introduction. In *Encyclopedia of Physical Science and Technology*. 3rd edition, 2003.

[128] D. Maio, D. Maltoni, R. Cappelli, J.L. Wayman, and A.K. Jain. Fvc2000: fingerprint verification competition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):402–412, 2002.

[129] L. Du, Z. He, Y. Wang, X. Wang, and A. T. S. Ho. An image hashing algorithm for authentication with multi-attack reference generation and adaptive thresholding. *Algorithms*, 13:227, 2020.

[130] L. Du, Z. Chen, and A. T. S. Ho. Binary multi-view perceptual hashing for image authentication. *Multimedia Tools and Applications*, 80(15):22927–22949, 06 2021.

[131] M. Grandini, E. Bagli, and G. Visani. Metrics for multi-class classification: An overview. August 2020. White Paper.

[132] Mitre Corporation. Common weakness enumeration, 2021. Last Accessed: 2022-07-08.

[133] A. Adler. Sample images can be independently restored from face recognition templates. In *Canadian Conference on Electrical and Computer Engineering. Toward a Caring and Humane Technology (CCECE 2003)*, pages 1163–1166, 2003.

[134] M. Hosseinzadeh, B. Sinopoli, and E. Garone. Feasibility and detection of replay attack in networked constrained cyber-physical systems. In *57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 712–717, 2019.

[135] M. Lebcir, S. Awang, and A. Benziane. Reversible watermarking technique for fingerprint authentication based on DCT. *IOP Conf Ser: Mater Sci Eng*, 769:012070, 2020.

[136] M. L. Benseddik, K. Zebbiche, M. S. Azzaz, and S. Sadoudi. Interpolation-based reversible data hiding in the transform domain for fingerprint images. *Multimedia Tools and Applications*, 81:20329–20356, 2022.

[137] V. Monga and B. L. Evans. Perceptual image hashing via feature points: Performance evaluation and tradeoffs. *IEEE Transactions on Image Processing*, 15(11):3452–3465, 2006.

[138] J. Oostveen, T. Kalker, and J. Haitsma. Feature extraction and a database strategy for video fingerprinting. In Shi-Kuo Chang, Zen Chen, and Suh-Yin Lee, editors, *Recent Advances in Visual Information Systems*, pages 117–128, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[139] P. Yip and K. R. Rao. On the shift property of DCTs and dsts. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(3):404–406, 1987.

[140] B. Ahmaderaghi, F. Kurugollu, J. M. D. Rincon, and A. Bouridane. Blind image watermark detection algorithm based on discrete shearlet transform using statistical decision theory. *IEEE Transactions on Computational Imaging*, 4(1):46–59, 2018.

[141] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain. Fvc2000: fingerprint verification competition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):402–412, 2002.

[142] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[143] J. Ren, X. Jiang, and J. Yuan. *Face and Facial Expressions Recognition and Analysis*, pages 3–29. Springer International Publishing, Cham, 2016.

[144] S. Q. Abbas, F. Ahmed, N. Živić, and O. Ur-Rehman. Perceptual image hashing using svd based noise resistant local binary pattern. In *2016 8th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pages 401–407, 2016.

[145] R. Davarzani, S. Mozaffari, and K. Yaghmaie. Perceptual image hashing using center-symmetric local binary patterns. *Multimedia Tools and Applications*, 75:4639–4667, 2016.

[146] S. Q. Abbas, F. Ahmed, and Y.-P. P. Chen. Perceptual image hashing using transform domain noise resistant local binary pattern. *Multimedia Tools and Applications*, 80:9849–9875, 2021.

[147] NIST/SEMATECH. e-handbook of statistical methods, 2012. Last Accessed: 2022-07-08.