Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

Cecilie Fougner

Secure and Efficient Preselection for Biometric Identification

Master's thesis in Communication Technology and Digital Security
Supervisor: Anamaria Costache
Co-supervisor: Pia Bauspieß
June 2023

NTNU
Norwegian University of Science and Technology

Cecilie Fougner

Secure and Efficient Preselection for Biometric Identification

Master's thesis in Communication Technology and Digital Security
Supervisor: Anamaria Costache
Co-supervisor: Pia Bauspieß
June 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

NTNU
Norwegian University of
Science and Technology

Title: Secure and Efficient Preselection for Biometric Identification

Student: Cecilie Fougner

Problem description:

Biometric identification describes the process of comparing an unidentified biometric probe against a reference database to check for a mated reference from the same identity. However, biometric data needs to be characterized as sensitive, as stated in European and international laws. Therefore, established security requirements exist in form of the ISO/IEC 24745 standard on biometric information protection that need to be fulfilled in order to protect the biometric information. One way of achieving biometric information protection is through the use of Fully Homomorphic Encryption (FHE), which this thesis will investigate.

The efficiency of FHE-protected identification systems is however a problem because FHE adds significant overhead to the computations. Therefore, this thesis describes an approach for biometric identification ensuring that both the security and the efficiency aspect are provided. To achieve this goal, this project will combine features from two previously established approaches: the first using stable hashes to efficiently group subjects together based on their similarity, and the second using Public-Key Encryption with Keyword Search (PEKS) with soft-biometric keywords to ensure that the privacy of the biometric information is protected.

The proposed approach will be implemented with the programming languages Python and C++ and use the OpenFHE library for support of the homomorphic encryption operations. Both the biometric and computational performance will be evaluated experimentally.

Approved on: 2023-02-17

Main supervisor: Associate Professor Anamaria Costache, NTNU

Co-supervisor: Pia Bauspieß, NTNU

Abstract

Biometric characteristics can be used to identify individuals due to their uniqueness and user-friendliness. Despite these benefits, there are privacy risks associated with biometric data that need to be managed. In addition, the computational workload of biometric identification systems needs to be improved to prevent the search in large-scale databases from becoming infeasible.

In this master's thesis, a biometric identification system is proposed where a preselection method is introduced to improve the computation workload. To this end, similar subjects are grouped in clusters by using the k-means clustering technique. To provide additional security, Fully Homomorphic Encryption is used on the reference templates to protect the sensitive information of the biometric data. Despite the high computation cost associated with this encryption scheme, the benefits outweigh the drawbacks because of the potential attacks with quantum computers and thus achieving long-term protection.

An experimental evaluation is performed for the proposed system, showing that the system is reduced to 2.8% of the workload effort of the baseline system performing an exhaustive search on the entire reference database. For evaluating the biometric performance, the proposed system achieves false positive identification rates around 0.001% and false negative identification rates from 1% to 0.001%. This indicates good accuracy. Therefore, the biometric identification system presented in this thesis can be considered efficient, accurate, and secure at the same time, improving upon the state-of-the-art.

Sammendrag

Biometriske kjennetegn kan brukes til å identifisere individer på grunn av deres særpreg og brukervennlighet. Til tross for disse fordelene så er det personvernrisiko knyttet med biometriske data som må håndteres. I tillegg må den beregningsmessige arbeidsmengden av biometriske identifiseringssytemer forbedres for å hindre at søk i store databaser skal bli umulige.

I denne masteroppgaven blir det presentert et biometrisk identifiseringssystem hvor en forhåndsvalgmetode er introdusert for å forbedre den beregnignsmessige arbeidsmengden. Til dette formålet grupperes like personer sammen i grupper ved å bruke k-means grupperingsalgoritme. For å introdusere mer sikkerhet er fullstendig homomorft kryptering brukt på referanse malene for å beskytte den sensitive informasjonen av den biometriske dataen. Til tross for de høye beregningskostadene som er knyttet til denne krypteringsmetoden, så vil fordelene overveie ulempene på grunn av de potensielle angrepene med kvantedatamaskiner og dermed oppnås langsidig beskyttelse.

En eksperimentell evaluering er gjennomført for det foreslåtte systemet, som viser at systemet er redusert ned til 2.8% av arbeidsmengden av grunnlinjesystemet som utfører fullstendig søk på hele referansedatabasen. Ved å evaluere den biometriske prestasjonen oppnår systemet falske positive identifiseringsrater rundt 0.001% og falske negative identifiseringsrater fra 1% til 0.001%. Dette indikerer god nøyaktighet. Dermed kan det biometriske identifiseringssystemet som presenteres i denne oppgaven bli betraktet som effektiv, nøyaktig og sikker på samme tid, og forbedre state-of-the-art.

Preface

This master's thesis concludes my degree in the 5-year MSc program in Communication Technology and Digital Security at the Norwegian University of Science and Technology (NTNU) in Trondheim. Throughout this process, I have had the opportunity to test a new research field and acquired knowledge that will be useful to take with me further.

I especially want to thank my Professor Anamaria Costache and my supervisor Pia Bauspieß for their support. They have guided me through the last two semesters with good meetings and reflections. With their safety, potential challenges and difficulties have been handled in a remarkable way.

Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

AS Authentication Server.

BFV Brakerski-Fan-Vercauteren.

BGV Brakerski-Gentry-Vercauteren.

BTP Biometric Template Protection.

C Client.

CKKS Cheon-Kim-Kim-Song.

CS Computation Server.

DET Detection Error Trade-off.

ECC Elliptic-Curve Cryptography.

FERET Face Recognition Technology.

FHE Fully Homomorphic Encryption.

FNIR False Negative Identification Rates.

FPIR False Positive Identification Rates.

FRGCv2 Face Recognition Grand Challenge version 2.

GapSVP Gap Shortest Vector Problem.

GDPR General Data Protection Regulation.

HE Homomorphic Encryption.

IBE Identity-Based Encryption.

IEC International Electrotechnical Commission.

IoM Index-of-Maximum.

ISO International Organization for Standardization.

LIoM Learning-Based Index-of-Maximum.

LSH Locality Sensitive Hashing.

LWE Learning with Errors.

NTNU Norwegian University of Science and Technology.

NTRU N-th Degree Truncated Polynomial Ring Units.

OpenFHE Open-Source Fully Homomorphic Encryption.

PEKS Public-Key Encryption with Keyword Search.

PHE Partially Homomorphic Encryption.

PKG Private Key Generator.

PQ Product Quantization.

R-LWE Ring-Learning with Errors.

RNS Residue Number System.

SHE Somewhat Homomorphic Encryption.

TFHE Fast Fully Homomorphic Encryption Over the Torus.

TNIR True Negative Identification Rate.

TPIR True Positive Identification Rate.

Chapter 1
Introduction

This Chapter introduces the topic of this thesis, describing the motivation, use cases, and challenges associated with biometric identification systems. To introduce the different tasks the thesis will focus on, research questions are also described. Lastly, this Chapter describes the remaining structure of this thesis.

1.1 Motivation and Challenges

The relevance of biometric data has emerged over the last century, with biometric characteristics such as facial features or fingerprint patterns used to identify and recognize persons [1]. This increasing usage can be associated with the fact that biometric data is both unique and user-friendly [2]. Every person has certain characteristics that are special and unique to themselves, and in addition, their biometric data can be applied for different use cases in the digital world [1]. For instance, using our facial image, we can unlock our phones, and using our fingerprint, we can receive a unique identifying document such as a passport. Unlocking a phone or going through passport control are examples of biometric verification, where a one-to-one comparison is computed against a known probe subject.

The more challenging case is biometric identification, where the identity of an unknown probe is to be determined through a search of a large biometric database. To this end, the system compares an unidentified probe against a reference database to find a mated reference [1]. The Indian National ID Programm Aadhaar [3] is one use case of biometric identification, where citizens are assigned unique identification numbers. As each citizen can only receive one identification number, the citizens are enrolled into the Aadhaar database when they receive their identification numbers [4]. For a new citizen, a biometric identification process is performed to check if the citizen is already enrolled in the database. For this purpose, an exhaustive search is performed, where the new citizen is compared to every citizen in the Aadhaar database [4].

Despite the user-friendliness of biometric data in identification systems, there are risks associated with its application. This is due to the fact that biometric data are considered sensitive and need to be protected accordingly [5]. If unauthorized persons process the biometric data, this can have major consequences [5]. Potential risks include impersonation attacks, disclosure of medical information that can be deduced from biometric data, and disclosure of sensitive personal information such as ethnicity [5]. To provide the appropriate security for the biometric data, different security standards are provided by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC), which need to be fulfilled when dealing with biometric data [6]. The following three main requirements for storing and processing biometric information securely are defined in ISO/IEC 24745 standard on biometric information protection [6]:

- *Unlinkability*: It should not be possible to link any protected references to their corresponding biometric sample or any other protected reference.
- *Renewability*: It should be possible that any protected reference of a biometric sample can be re-created without being linked to each other or without the need to re-enroll the corresponding biometric sample.
- *Irreversibility*: It should not be possible that knowledge of a protected biometric reference will result in knowledge of the original biometric sample.

Therefore, the main focus of this thesis will be ensuring that the security challenge is addressed in accordance with the ISO/IEC 24745 requirements introduced above. Several recent approaches to ensuring the appropriate level of security utilize Fully Homomorphic Encryption (FHE). Modern FHE schemes belong to the class of lattice-based cryptography and are assumed to be secure against attacks implemented on a quantum computer. If the correct parameter sets are chosen, this gives long-term protection. The use of biometric identification through exhaustive search on a database encrypted with FHE will be considered the baseline system in this thesis.

Although the security is the most important challenge when considering biometric data and biometric identification systems, a secondary problem for biometric systems is ensuring we maintain efficiency. For instance, using FHE comes with a significant computational overhead, especially regarding biometric identification searches over large-scale databases, which affects the efficiency. This is also reflected in the biometric identification of the Aadhaar database, which currently enrolls more than 1.3 billion Indian citizens [3]. With an exhaustive search on such a large number of references, a protected search would become infeasible [7].

To improve the efficiency of biometric identification, a preselection method can be introduced to reduce the search space [4]. For this thesis, two research approaches

of biometric identification with preselection on protected reference databases have been studied and analyzed. These two approaches are Stable Hashing [8] and the Public-Key Encryption with Keyword Search (PEKS)-based identification with soft-biometric keywords [7]. Both approaches aim to reduce the number of samples that are considered for the final comparison, while the reference database continues to be encrypted using FHE. While the stable hashing approach experiences security drawbacks, the PEKS-based identification has accuracy drawbacks, and this will be shown in this thesis.

The goal of this thesis is to combine the stable hashing [8] and PEKS-based [7] approach in a way that combines their benefits, but mitigates their respective drawbacks. With this, we can construct a proposed system that aims at being secure, accurate, and efficient.

1.2 Research Questions

The research questions formulated in the pre-project course TTM4502 [9] are maintained for this thesis.

RQ1. What are the main challenges in using cryptography for soft-biometrics-based preselection?

RQ2. What alternatives to soft-biometrics can be utilized for encrypted preselection?

RQ3. How will the performance be affected by combining PEKS with stable hashes?

RQ4. Will it be necessary to use a binning method on the database in advance of an identification search?

1.3 Outline

The remaining Chapters of this thesis are structured in the following way:

Chapter 2: Background and Preliminaries describes the necessary biometric terminology, equations, template protection, and preselection approaches used in the proposed system.

Chapter 3: Related Work presents similar and existing approaches for biometric identification and preselection approaches within the same research field as the thesis, but not further used.

Chapter 4: Methodology introduces the different work packages of this thesis.

Chapter 5: Proposed System describes the proposed biometric identification system for this thesis. This is separated into an enrollment and identification phase with associated figures and algorithms to illustrate these two transactions.

Chapter 6: Experimental Evaluation presents the implementation and evaluation of the proposed system. A description of the testing environment and the execution metrics are provided before the Chapter describes the results of the experiments in detail.

Chapter 7: Discussion includes an analysis and discussion of the proposed system and its results from the experimental evaluation. This Chapter also addresses the research questions and details how they have been solved throughout this thesis.

Chapter 8: Conclusion summarizes this thesis by highlighting the most significant features of the proposed system, its provided results, and if the system works as expected to achieve the thesis goal. Lastly, this Chapter introduces further work.

Chapter 2
Background and Preliminaries

The following Chapter describes different concepts and building blocks essential for understanding the proposed system for this thesis. Section 2.1 introduces biometric identification with related terminology. Section 2.2 introduces the baseline system for biometric identification and describes how the security of biometric templates can be improved. Lastly, Section 2.3 and Section 2.4 describe two different approaches to speed up the baseline system through preselection. These approaches are the basis of the work in this thesis and aim at efficient and secure preselection for biometric identification. Even though there are certain drawbacks related to the concepts introduced in Section 2.3 and Section 2.4, these drawbacks will be discussed in Chapter 5.

2.1 Biometric Identification

The ISO and the IEC provide several standards for unified biometric vocabulary and security requirements that need to be considered when working with biometric data. According to the ISO/IEC 2382 standard [10], biometric characteristics of an individual include their behavioral and biological characteristics such as face topography, facial skin texture, or finger topography, which often are used as entry fields in automated biometric recognition systems. These automated biometric recognition systems aim to compare biometric features against stored biometric references [10].

A biometric system can operate in two modes, either verification or identification [1]. This is illustrated in Figure 2.1, which also shows the enrollment process of references into a biometric enrollment database. When a biometric system operates in verification mode, the system validates a person's identity by performing a $1:1$ comparison between the person's biometric features and its corresponding reference template enrolled in the database [1]. The focus of this thesis is however on biometric systems operating in identification mode, where the system performs a $1:N$ com-

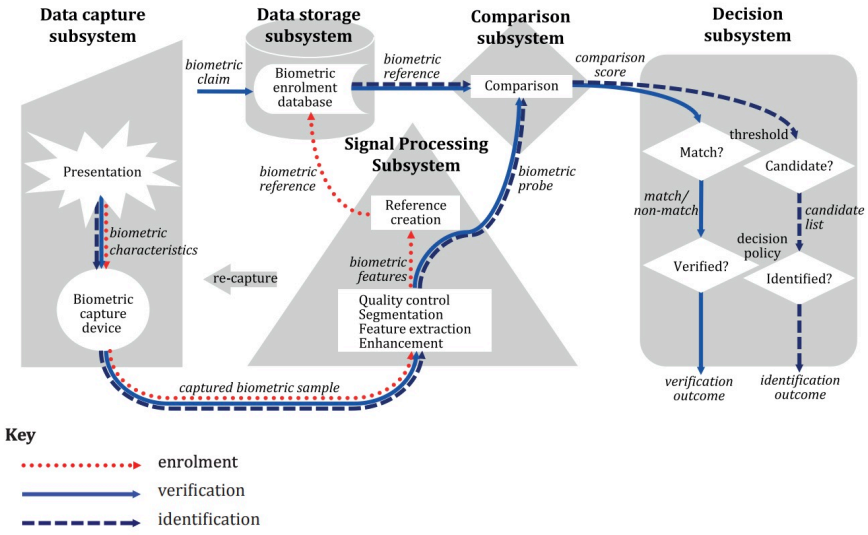


Figure 2.1: Figure from [13] showing the ISO standard for biometric system.

parison [1]. The idea behind biometric identification is to compare an unidentified biometric probe reference against a biometric enrollment database in order to find a mated reference [1]. Finding a mated reference means finding a reference where the biometric probe and one biometric reference from the enrollment database are from the same biometric subject [10].

As illustrated in Figure 2.1, during the enrollment, feature vectors are extracted from a biometric sample. For example, fixed-length vectors with floating point values are extracted from face images by using deep convolutional neural networks [11]. These feature vectors are stored in the enrollment database. During identification, a fresh probe sample (e.g., face image) is captured and the features are extracted in the same manner as during enrollment. These probe features are then compared to all reference feature vectors through the use of a single distance metric, e.g., the Euclidean distance [12]. The distance scores are ranked and the reference subject associated with the lowest distance score is returned as the identification outcome.

2.1.1 General Data Protection Regulation

In addition to the standards from the ISO and the IEC discussed in Chapter 1, additional regulations must be carefully followed when dealing with biometric data. One such regulation is the General Data Protection Regulation (GDPR) [5], which describes restrictions on protecting personal data and emphasizes that everyone has

the legal right to protect their personal data concerning themselves.

As defined by the GDPR [5], biometric data contain physical, physiological, or biometric characteristics that can be categorized as personal data used to confirm the unique identification of an individual, for instance, with a facial image. With this definition, the GDPR [5] further classifies biometric data as sensitive due to data processing risks. It is especially the soft-biometric characteristics such as ethnicity that are characterized as sensitive by the GDPR and hence need to be hidden from unauthorized persons [5]. This is because ethnicity can be deduced from the biometric characteristics. Every individual has the fundamental right that only authorized persons should be allowed to process their biometric data, e.g., for identification purposes. If an unauthorized person, e.g., an attacker, obtains access to the sensitive data of an individual, the attacker may use the individual's personal information for impersonation attacks [5]. Such attacks may cause a loss of confidence in the biometric recognition system's reliability [9]. As a result, the GDPR [5] emphasizes that the processing of biometric data should only be allowed for certain cases specified by the Regulation.

Equivalent to biometric data, the GDPR [5] also defines health data as personal data. This data includes an individual's physical or mental health that may reveal information about the overall health status of the individual. From Amerifar *et al.* [14], it can be seen that through analyzing characteristics of the iris image of a person, i.e., the structure of the eye, abnormalities in the iris may be revealed. With these abnormalities, medical information could be revealed from the biometric template [14]. Because of the potential risks of leaking sensitive data such as health information, the GDPR [5] introduces conditions and limitations regarding processing data concerning health. Similar to biometric data, this should only be processed for specific cases defined by the Regulation.

2.2 Homomorphic Encryption

To be in compliance with the GDPR [5] and the security requirements from the ISO/IEC 24745 [6], biometric templates can be protected using Homomorphic Encryption (HE) [15]. For the purpose of this thesis, we consider the use of HE as the baseline system for biometric identification. Before describing HE and the baseline system, the following Section describes the most common architecture for biometric identification systems.

2.2.1 Biometric Architecture

Protected biometric identification systems often use an architecture based on the security assumptions presented in [16]. Through utilizing a two-server architecture

consisting of a Computation Server (CS) and an Authentication Server (AS), a secure biometric identification process can be achieved. In addition, the CS can communicate with a third entity, a Client (C), which provides the biometric templates. An insecure communication channel is assumed between the different entities, meaning an attacker can eavesdrop on the communication. Further details on the communication channel are not described since this is beyond the scope of this thesis.

The goal of this architecture is to divide different responsibilities between different servers and also to ensure that the privacy aspect is provided. While the C and the CS have access to the public key information of the system, only AS controls the secret key [16]. In addition, the CS will also have access to a protected reference database. A public-key infrastructure can be used for the key management between the different entities, but this is outside the scope of this thesis.

With separate servers, we assume that the AS only needs to be connected to a local network since it can communicate with the outside world through communication with the CS. On the other hand, we assume that the CS is connected to the network and thereby is accessible to an attacker over the Internet. These separations hence give additional protection to the protocol. With this architecture, most biometric systems assume the servers will behave according to the protocol, thereby assuming a semi-honest model [17].

2.2.2 Homomorphic Encryption

HE schemes are often applied to biometric data because of their advantage in ensuring privacy [18]. As explained by Drozdowski *et al.* [19], HE enables the possibility of more functionality in the encrypted domain. Thereby, it is possible to perform operations in the encrypted domain and still obtain the same results without performing decryption [19]. With biometric data, HE can be used to compare two encrypted references and obtain the same comparison score as if the comparison was performed with unprotected plaintext references.

The characteristic property associated of HE scheme is the homomorphic property. This property is defined in the following way [15]:

$$E(m_1) \odot E(m_2) = E(m_1 \odot m_2). \tag{2.1}$$

The operation $\odot$ indicates a specific operation used in the encrypted domain, such as multiplication or addition. If this property is true for the specified operation $\odot$, and for every message m_1 and m_2 in the set M consisting of all the possible messages, then the homomorphic property is present [19].

HE schemes can be classified into three main types, where each type depends on the supported operations and how many times each operation can be used for the encryption scheme [20]. Partially Homomorphic Encryption (PHE) supports only one single operation, either addition or multiplication, that can be used an unlimited number of times. Somewhat Homomorphic Encryption (SHE) supports using, for instance, multiplication and addition, but only for a limited number of times. Lastly, FHE supports an unlimited number of operations that can be used an unlimited number of times [20].

2.2.3 Fully Homomorphic Encryption

The first FHE scheme was presented by Gentry in 2009 [18]. This scheme was introduced as a solution to the drawbacks of the concept of privacy homomorphisms introduced by Rivest *et al.* [15] in 1978. The problem with privacy homomorphisms was the ability to ensure security with a large set of operations and limited applicability [15]. Despite many proposals for a viable construction for FHE, Gentry [18] first managed to construct a general FHE blueprint using ideal lattices. Such schemes add noise to ciphertexts, and using operations like multiplication makes the noise grow. This is what has limited the number of operations before Gentry [18], as the growing noise leads to incorrect decryption. The essential part of Gentry's FHE scheme was the *bootstrapping* operation. According to Gentry [18], a scheme is bootstrappable if it is possible to evaluate its own decryption circuit. If a ciphertext can be decrypted under a layer of encryption, it can also be refreshed [18]. Through bootstrapping, the idea is to refresh ciphertexts from the same plaintext, such that the new and fresh ciphertext obtains a shorter error vector [18]. Gentry [18] only managed to enable bootstrapping using a technique called *squashing*. The bootstrapping procedure was only possible if the decryption depth was reduced, and the squashing technique henced allowed for a reduction of the complexity of Gentry's decryption circuit [20].

Even though the construction of the first successful FHE scheme was promising, the complexity of Gentry's scheme [18] was relatively high because of the computational costs associated with the bootstrapping mechanism [20]. Acar *et al.* [20] describe that, due to this complexity, new optimizations of FHE schemes have been introduced based on Gentry's first FHE proposal [18].

Another variant of FHE optimization is related to Learning with Errors (LWE)-based FHE schemes [20]. As a solution to the complex bootstrapping technique by Gentry [18], Brakerski *et al.* [21] proposed the Brakerski-Gentry-Vercauteren (BGV) scheme as a FHE scheme without the bootstrapping procedure. This BGV scheme is based on the Ring-Learning with Errors (R-LWE) problem. LWE was first introduced by Regev [22] in 2009, showing that a quantum algorithm could be constructed by reducing worst-case lattice problems, e.g., the Gap Shortest Vector

Problem (GapSVP) [22]. While LWE uses the ring of integers modulo q, R-LWE uses a polynomial ring [21]. The hardness of R-LWE is reduced to the hardness of the LWE, and we can thus assume a post-quantum hardness. On the other hand, Fan and Vercauteren [23] proposed the Brakerski-Fan-Vercauteren (BFV) scheme in 2012. This BFV scheme is based on improving the LWE-based FHE scheme by Brakerski [24] into a R-LWE-based FHE scheme. The benefit of this BFV scheme [23] is using a simplified bootstrapping technique and faster computations. Another improvement for FHE is the Fast Fully Homomorphic Encryption Over the Torus (TFHE) approach proposed by Chillotti *et al.* [25], which also utilizes the LWE problem. Lastly, Cheon *et al.* [26] proposed the Cheon-Kim-Kim-Song (CKKS) scheme in 2017. In contrast to the aforementioned optimization of FHE, the CKKS scheme [26] introduced a variant of FHE using approximation arithmetic which enables the use of FHE over floating point values which are approximated to a certain level of precision. The security of this CKKS scheme is also based on the R-LWE problem [26].

2.2.4 Baseline System

FHE can be considered to construct a baseline system for protected biometric identification using an exhaustive search over the entire enrollment database, which will be described in the following [19]. For the enrollment phase, a reference template represented as a FHE plaintext r is encrypted to $c_r \leftarrow Enc(pk, r)$ and stored in the reference database together with an identifier ID (e.g., a personal identification number). For an unidentified probe feature vector, an associated protected ciphertext c_p is produced through FHE encryption with the following algorithm.

$$c_p \leftarrow Enc(pk, p). \tag{2.2}$$

The pk is the public key of the FHE scheme, while p is the plaintext of the probe template. The baseline system using FHE will then compare the protected ciphertext of the probe against every protected reference in a protected reference database in order to decide if there is a mated reference for the probe.

Several homomorphic operations are used for one FHE comparison between the probe and one database reference, as described in [12]. Firstly, a subtraction operation, $\Delta_{pr} = Sub(c_p, c_r)$, is performed between the probe ciphertext c_p and a reference ciphertext c_r stored in the database. Secondly, the square of the subtraction result, Δ_{pr}, is computed, i.e., multiplying the result by itself, $mult_{pr} = Mult(\Delta_{pr}, \Delta_{pr})$. Thirdly, a number of rotation operations, $rot_{pr} = Rot(mult_{pr}, 1)$, are performed where the multiplication result is shifted one position to the left. These are combined with iterative additions $add_{pr} = Add(mult_{pr}, rot_{pr})$ where the multiplication result is added with the rotation result. With d as the dimension of the probe and the database reference, $d - 1$ executions of the rotation and addition operations are

performed [12]. The computational workload of this baseline system is heavily based on the number of rotation operations, defined by Engelsma *et al.* [27] as the most expensive homomorphic operations.

2.2.5 Packing Data in FHE

Despite the benefits of using FHE, the computational overhead is high [28]. This is because the ciphertexts are expected to be larger than the biometric plaintext data to be encrypted, and thereby this reduces the efficiency of the system [28]. One solution to this problem is ciphertext packing, where multiple plaintexts are encrypted into one ciphertext [28]. As also described by Wu *et al.* [29], ciphertext packing allows for parallel computation, which improves the overall efficiency. One application of ciphertext packing is coefficient packing. This was used in [30] to rescue the workload of the baseline biometric identification system. As described by Bauspieß *et al.* [30], coefficient packing can improve the biometric identification comparisons and thereby decrease the relative computational complexity of FHE. Coefficient packing was also used in the biometric identification process in [7], which is one of the two preselection schemes that are the basis of this thesis. With coefficient packing, multiple plaintexts can be concatenated and encrypted into one ciphertext [30]. This means that with s available ciphertext slots and d as the dimension for every template, it is possible to concatenate $k = \lfloor \frac{s}{d} \rfloor$ plaintext into one [30]. As described by Bauspieß *et al.* [30], coefficient packing uses only one comparison computation for every k comparisons in the baseline system. With this approach, the coefficient packing technique utilized the concept of feature transformation, which aims to reduce the cost of each comparison [4]. At the same time, the approach is also an exhaustive search comparison because every combination needs to be compared and evaluated. This technique uses all available template slots, and the biometric identification can experience a significant computational workload reduction [30].

2.2.6 Other Biometric Template Protection Schemes

As described in Section 2.2, the benefit of HE schemes is the ability to perform biometric comparisons in the encrypted domain. This property is often associated with Biometric Template Protection (BTP) schemes [31]. Other categories of BTP schemes are biometric cryptosystems and cancelable biometrics [31]. As defined by Rathgeb and Uhl [31], biometric cryptosystems associate a digital key with the biometric data, whereas cancelable biometrics append transformations to the biometric data such that comparisons can be performed in a secure domain.

With biometric cryptosystems, the added security makes the process of forging, copying, and sharing biometric data significantly more advanced and difficult than previously password-based key approaches [31]. An important characteristic of this

BTP scheme is the helper data. This is defined in [31] as the public information of the biometric data that is needed to generate or retrieve the associated digital key. The helper data are an essential part of the key reconstruction, and therefore, they should not reveal significant information about their associated original biometric template [31]. Since the biometric cryptosystem approach involves digital keys, biometric comparisons are not performed directly on the biometric templates. Instead, with biometric comparisons, the digital keys are verified [31]. Either the key is returned if the same biometric template is being compared, or an error message is returned otherwise [31]. Despite the benefits of using this BTP scheme, Rathgeb and Uhl [31] explain the drawback of biometric cryptosystems often experiencing a significant decrease in recognition performance. In addition, the digital keys must be of sufficient size to prevent an attacker from guessing the corresponding keys from a biometric template [31].

The security property of the cancelable biometric scheme is that recovering the original biometric data should be computationally hard [31]. As explained in [31], the biometric data are being distorted by applying different transformations, but it is nevertheless essential that this is not reducing the corresponding biometric characteristics. In addition, different transformations are applied to various applications to make it harder for an attacker to link similar subjects [31]. According to Rathgeb and Uhl [31], the cancelable biometric schemes can be divided into two categories depending on their functionality. The first category, *non-invertible transforms*, applies non-invertible functions to prevent an attacker from being able to reconstruct the biometric data. The second category, *biometric salting*, applies invertible transformations on the biometric template with the transformation parameters as a hidden secrecy [31]. Both the biometric salting approach and the non-invertible transform approach often lead to a decrease in the biometric performance, i.e., the accuracy [31].

2.3 Stable Hashes

Recently, Osorio-Roig *et al.* [8] presented an approach to efficient face identification based on FHE using a stable hash generation for the purpose of preselection. This is one of the two works that are the basis of this thesis and will now be described in more detail in this Section. The process used so-called stable hash codes and a hash lookup table to group facial references based on similarity. The approach in [8] can be divided into an enrollment and identification Step, as illustrated in Figure 2.2. During the enrollment Step, N enrollment references are enrolled in the system. A corresponding stable hash code is produced for every enrollment reference through the stable hash generation procedure. This stable hash code is based on the input features from the face image of the given enrollment reference. In addition, a protected template of the enrollment reference is produced through FHE encryption. The associated

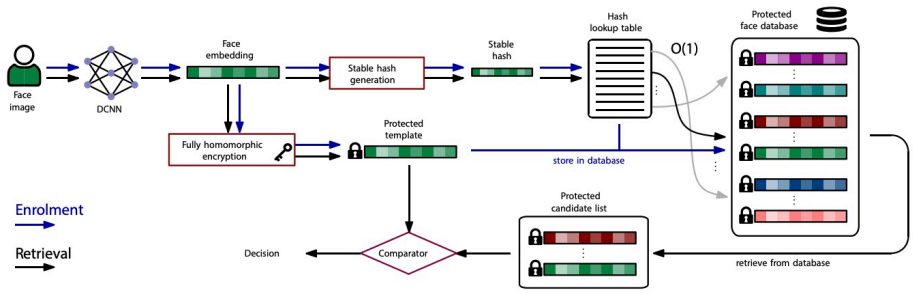


Figure 2.2: Figure from [8] showing the enrollment and identification (retrieval) process for the biometric identification system using stable hash generation.

stable hash code is then stored as a key index in a hash lookup table together with its corresponding protected reference template. During the enrollment of reference samples, two different options can be expected. The first option is creating a new stable hash code, resulting in a new entry in the hash lookup table. On the other hand, the second option is creating an already existing stable hash code, thereby resulting in a new collision where the protected reference is associated with other references from the same stable hash code [8].

The hash lookup table is further used during the identification Step. For a given unidentified probe sample, the corresponding stable hash code is produced through stable hash generation based on the probe feature vector. By utilizing the hash lookup table, the system will use the stable hash code of the probe as a key index to retrieve all the protected references from the enrollment Step that have the same stable hash code as the probe. The advantage of using this stable hash generation scheme is its efficiency that has computational complexity of $O(1)$, meaning that returning a candidate list of similar references can easily be carried out from an exact match by only processing the stable hash code from the hash lookup table [8]. All the references in the returning candidate list are retrieved from the same stable hash code. At the end of the identification Step, the reduced candidate list is compared with the probe template, as described in the baseline system described in Section 2.2.4, to derive if there is a mated reference in the candidate list for the given probe.

2.3.1 Product Quantization

The aforementioned stable hash generation used in [8] uses Product Quantization (PQ) in order to reduce the total representation space of vectors without affecting the dimensionality [32]. The process of generating the stable hash codes is illustrated in Figure 2.3. From the enrollment of a subject S, its feature representation can

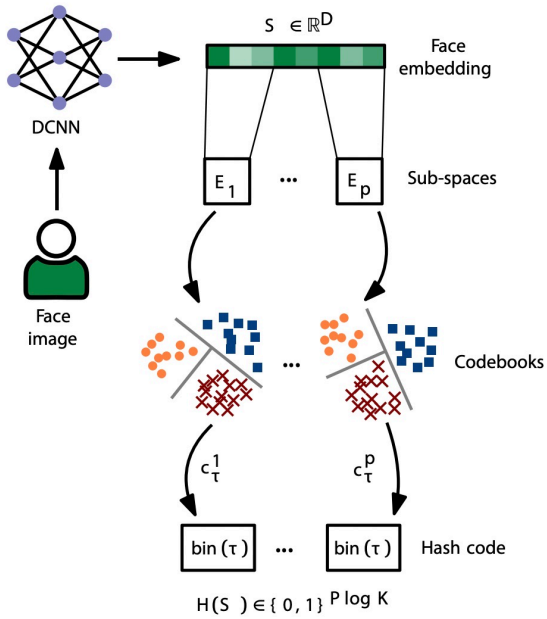


Figure 2.3: Figure from [8] showing the mapping of features to the nearest cluster.

be divided into P equal sub-vectors called sub-spaces, $S = \{E_1, ..., E_P\}$ [8]. Each sub-space E_j, for $1 \leq j \leq p$, has a corresponding codebook of size K, i.e., consisting of K clusters, $C = \{c_1, ..., c_K\}$, where each sub-space is mapped to its nearest cluster [8]. The index of the nearest cluster is then represented as a binary hash code $Q(E_j)$ consisting of $\log_2(K)$ bits [8]. Finally, by concatenating the binary hash code $Q(E_j)$ for each sub-space E_j, the resulting stable hash code *H(S)* is generated of the size $P \log_2(K)$ bits [8].

2.3.2 K-Means Clustering Algorithm

The stable hash generation scheme using PQ in [8] used different clustering techniques to associate each sub-space with its nearest cluster. Even though Osorio-Roig *et al.* [8] evaluated four different clustering algorithms, this thesis will only focus on the k-means clustering algorithm further because of its accuracy and performance derived from the experimental evaluation in [8] and its computationally economical benefits explained in [33].

MacQueen [33] describes the k-means algorithm in the following way: Initially, the k-means algorithm will predefine k random center locations. In an iterative process, the algorithm will group the entire population (i.e., the set of reference

feature vectors) into k clusters based on the predefined centers. Each element is associated with its nearest cluster by comparing the minimal distance to all the centers. For each iteration, the centers will be updated to the mean of the elements in each cluster. With this procedure, the k-means clustering is a suitable technique for processing large sample sets, and some practical applications are, for instance, similarity grouping [33]. Based on this, the k-means approach is a helpful technique in biometric identification to group subjects based on their similarities.

The k-means algorithm uses a distance measure to compute a similarity score to associate a biometric sample with its nearest cluster [33]. One example of a suitable distance measure often used to compute similarity scores in different clustering techniques is the Euclidean distance [34]. The Euclidean distance between two points, x and y, is defined in the following way [34]:

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}. \tag{2.3}$$

Even though the Euclidean distance measure is suitable for computing a similarity score, the complexity and efficiency can be improved by not computing the squared root. The squared Euclidean distance can also be applied as a distance measure to the k-means algorithm, improving the clustering distribution. This is defined in the following way [34]:

$$d(x, y)^2 = \sum_{i=1}^{n}(x_i - y_i)^2. \tag{2.4}$$

2.4 Public-Key Encryption With Keyword Search

The following Sections describe the second approach to preselection on FHE protected data that is the basis to this thesis, which is based on PEKS. In 2004, Boneh *et al.* [35] introduced the mechanism of PEKS. PEKS is similar to standard public-key encryption schemes where a sender encrypts a message using the sender's public key, and the receiver decrypts the message using its private key. The only difference is the improved functionality of testing in the encrypted domain whether keyword-specific content is appended to the message without applying decryption [35]. To implement this functionality, the sender creates a ciphertext of each specified keyword with the receiver's public key, $PEKS(pk, keyword)$, and appends this to the original encrypted message, $E(pk, M)$ [35]. This process is illustrated in [35] with the following Equation for n specified keywords:

$$E(pk, M) \;||\; PEKS(pk, keyword_1) \;||\; ... \;||\; PEKS(pk, keyword_n). \tag{2.5}$$

For each specified keyword, there exists a corresponding counterpart called trapdoor. These trapdoors enable the functionality of comparing the ciphertext of a keyword

with its corresponding trapdoor to test whether they contain the same keyword [35]. If an external part performs this comparison functionality, for instance, a gateway, the external part will learn nothing about the keywords, thereby ensuring the mechanism's security [35].

2.4.1 PEKS Definition

A definition of a PEKS scheme is defined in detail in [36], where a PEKS scheme is a tuple of four algorithms $PEKS = (\mathbf{KeyGen}, \mathbf{PEKS}, \mathbf{Trapdoor}, \mathbf{Test})$:

- With the security parameter k as the input parameter, the public and secret key pair (pk, sk) is computed in the following way:

$$(pk, sk) \leftarrow \mathbf{KeyGen}(1^k). \tag{2.6}$$

- With the user's public key pk and a keyword $w \in \{0,1\}^*$ as the input parameters, the searchable ciphertext s_w is computed in the following way:

$$s_w \leftarrow \mathbf{PEKS}(pk, w). \tag{2.7}$$

- With the user's secret key sk and a keyword $w \in \{0,1\}^*$ as input parameters, the trapdoor t_w is computed in the following way:

$$t_w \leftarrow \mathbf{Trapdoor}(sk, w). \tag{2.8}$$

- With a trapdoor $t_w = \mathbf{Trapdoor}(sk, w')$ and a searchable ciphertext $s_w = \mathbf{PEKS}(pk, w)$ as input parameters, a bit b is produced in the following way, where $b = 1$ if $w = w'$, and $b = 0$ otherwise:

$$b \leftarrow \mathbf{Test}(t_w, s_w). \tag{2.9}$$

2.4.2 Identity-Based Encryption

Boneh *et al.* [35] describe that the defined PEKS scheme is closely related to Identity-Based Encryption (IBE) and describe that PEKS can be constructed from IBE and inherits its security properties. Similar to PEKS, the IBE is also based on a public-key encryption scheme. As explained by Boneh and Franklin [37], the purpose of IBE is to encrypt a message using some identifying information of the receiver, e.g., the email address. The receiver can then contact a third party, Private Key Generator (PKG), to authenticate themselves. From the PKG, the receiver will obtain its private key and is then able to decrypt the message [37].

A definition of an IBE scheme is defined in detail in [36], where an IBE scheme is a tuple of four algorithms $IBE = (\mathbf{Setup}, \mathbf{Extract}, \mathbf{Enc}, \mathbf{Dec})$:

- $(mpk, msk) \leftarrow \textbf{Setup}(1^k)$: With the security parameter k as input parameter, this algorithm generates the master public and master secret key pair (mpk, msk).
- $sk \leftarrow \textbf{Extract}(id, msk, mpk)$: With the user's identity $id \in \{0,1\}^*, mpk$, and msk as input parameters, this algorithm generates the user's secret key sk.
- $c \leftarrow \textbf{Enc}(m, id, mpk)$: With the message $m \in \{0,1\}^*$, identity id, and mpk as input parameters, this algorithm produces a ciphertext c.
- $m \leftarrow \textbf{Dec}(c, sk)$: With the ciphertext c, the receiver's secret key sk, and mpk, this algorithm produces the original message m from the ciphertext c.

2.4.3 Lattice-Based PEKS

Even though the original PEKS scheme introduced by Boneh *et al.* [35] led to significant efficiency with its ability to compare encrypted messages and keywords in the encrypted domain, this concept also came with drawbacks. As described by Behnia *et al.* [36], the combination of extensive use of pairing computation and the costly **Test**() algorithm, Equation (2.9), introduces considerable heavy cryptographic delays. In addition, [36] also explains the lack of post-quantum security, which often is required for applications handling sensitive user data. In order to provide a certain level of security, cryptographic key sizes will continuously be increased because of the powerful computations and breakthroughs in today's society [36]. Current PEKS schemes will therefore not be cryptographically secure in the long-term because they are based on cryptographic tools, e.g., Elliptic-Curve Cryptography (ECC), that may become infeasible with heavily increased key sizes [36].

In order to solve these aforementioned problems, Behnia *et al.* [36] proposed the first lattice-based PEKS scheme using N-th Degree Truncated Polynomial Ring Units (NTRU) in 2017. This NTRU-PEKS scheme utilizes the IBE scheme and requirements presented in [38]. By applying the progress in lattice-based cryptography with the use of R-LWE [39], the NTRU-PEKS scheme is considered to offer post-quantum security because of the security guarantees from R-LWE [36]. In addition, exploiting fast arithmetic operations over polynomial rings with R-LWE makes the NTRU-PEKS scheme more efficient than the previous PEKS scheme operating on pairwise computation [36].

2.4.4 PEKS With Soft-Biometric Keywords

In 2022, Bauspieß *et al.* [7] presented an approach introducing the use of biometrics to the already established approach of PEKS presented by Boneh *et al.* [35] and Behnia *et al.* [36]. The process from Bauspieß *et al.* [7] used PEKS with soft-biometric keywords. From Dantcheva *et al.* [40], soft-biometrics can be defined as biometric

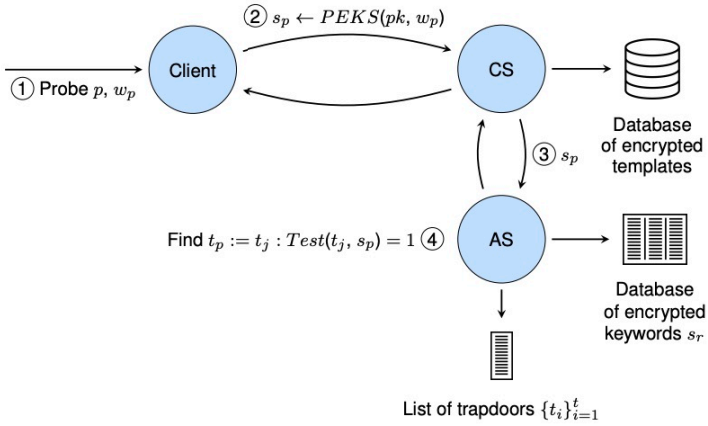


Figure 2.4: Figure from [7] showing the reverse PEKS search to retrieve the trapdoor associated with the probe template.

characteristics such as gender, age, or ethnicity, which can be associated with several persons. During an identification search, the idea is to reduce the search space to a smaller candidate list consisting of only the protected references from an enrollment database with the same soft-biometric keywords as the identifying probe template [7].

The PEKS concept using soft-biometrics used the two-server architecture presented in Section 2.2.1. With this model, there is a need for a mechanism called reverse PEKS search in order to produce the corresponding trapdoor to an identifying probe keyword vector [7], as illustrated in Figure 2.4. Since only the AS has access to the secret key sk, this is the only place where the **Trapdoor**() function, Equation (2.8), can be performed. The AS is initially given a list of trapdoors associated with every possible keyword vector. With these trapdoors, the AS will then use the **Test**() function, Equation (2.9), to find the associated trapdoor to the searchable ciphertext from the **PEKS**() function, Equation (2.7), sent from the CS [7].

In addition, Bauspieß *et al.* [7] also used the concept of a binning method to restrict the CS from keeping track of which protected references have the same soft-biometric keywords if the candidate list is small. With this mechanism, the reference templates are initially divided into equal bins. The distribution of the subjects and the bin size depend on the soft-biometric keyword distribution [7]. After the reverse PEKS search, the AS compares the associated trapdoor with each bin to find the associated bin holding the soft-biometric keywords for the identifying probe template [7]. This bin number is then transferred to the CS, which can perform the identification search only on the protected references in this bin [7]. With this

process, the CS cannot deduce whether the references in one bin all share the same soft-biometric characteristics, or have mixed soft-biometric characteristics. [7].

Chapter 3
Related Work

The following Chapter presents related material regarding workload reduction methods and indexing schemes. These topics are relevant to the concept for this thesis and thereby show a different usage of the relevant concepts.

3.1 Workload Reduction Methods

Different workload reduction approaches have been utilized to improve the efficiency of automated biometric recognition systems. In 2019, Drozdowski *et al.* [4] published an article where various workload reduction methods were discussed. These methods are separated into two main categories. The first category is the feature transformation approach, aiming to reduce the computational cost associated with each comparison [4]. According to Drozdowski *et al.* [4], this cost reduction can be achieved by creating more efficient representations of the biometric templates, for instance, by reducing the dimensionality. The other category is the preselection approach, which is the focus of this thesis. In contrast to reducing the cost of each comparison, this approach aims to reduce the number of biometric comparisons such that the search space is significantly reduced [4]. The approaches with the stable hash clustering and the PEKS, described in Section 2.3 and Section 2.4, respectively, are both examples of preselection approaches. Other related material approaches using preselection are described further in Section 3.1.1.

3.1.1 Preselection

Feature Fusion

One way of reducing the search space for a biometric identification search is by utilizing the concept of *multi-biometric systems*, which combines multiple references to reduce the workload [41]. Drozdowski *et al.* [41] illustrate this concept using feature fusion to reduce the number of template comparisons. The authors explain that a search tree can be constructed by fusing the enrollment references to create a

pre-filtering approach [41]. For identification of a probe template, the probe is at each tree level compared against the fused templates to find all the candidates most similar to the probe [41].

The selection of the feature vector pairs plays an important part in obtaining a significant workload reduction and improving the biometric performance. Similar references should be fused together to increase the discriminative power of the preselection procedure, i.e., increase the capability to distinguish references [41]. In addition to influencing the workload, selecting the feature vector pairs significantly impacts the comparison scores against a probe template [41]. Because of the discriminative power for the resulting fused template pairs, the probe template will achieve better comparison scores against mated fused templates, i.e., similar templates, than non-mated fused templates, i.e., other distinct templates [41]. This feature fusion method reduces the workload for biometric identification but suffers from significant storage space and the complexity of only being optimized for a given enrollment database [41]. If a new reference is appended, the search tree and the fusion of the enrollment references need to be reconstructed, which increases the overall complexity [41].

Cascading Database Filtering

On the contrary to reducing the biometric comparisons by combining several features from similar references, Drozdowski *et al.* [42] describe another approach to multi-biometric identification by utilizing a filtering technique with multiple biometric modalities, i.e., different biometric characteristics such as face image, fingerprint, and iris code. According to Drozdowski *et al.* [42], a cascading filtering system is produced in the following way: In the first phase, for a given probe template, the corresponding features of each specified biometric modality are extracted. Secondly, the first modality feature of the probe is compared against the enrollment database to find a candidate list with the best comparison scores. In the third phase, a new modality feature of the probe is compared against the resulting candidate list from the previous phase. This process is recursively followed until all the specified modality features of the probe have been compared against a candidate list of the prior phase. For the last phase, when all modality features have been used for filtering, a resulting candidate list is achieved and used for the final identification decision [42].

In evaluating this approach, the biometric performance and the workload reduction experience acceptable results for using two and three modalities [42]. Despite this, the defined approach is sensitive regarding the ordering of the modalities and which samples are used [42]. For instance, if a poor-quality sample is filtered out and used in the initial phase of the cascading filtering, the identification errors of the proposed system will most likely increase [42].

3.2 Indexing Schemes

In addition to being categorized as a preselection approach, the stable hash technique, described in Section 2.3, can also be defined as an indexing approach. The idea with an indexing approach is to produce an index for each biometric template in the database and then use a distance measure on these indexes to find a corresponding biometric reference for a given biometric probe.

Murakami *et al.* [43] describe a related indexing scheme for biometric identification. A key element to produce these indexes is some defined biometric features, called pivots. These pivots are either generated from artificial features or remaining features from the biometric subjects not considered for the enrollment database [43]. Another key element of this approach is that the indexing scheme is based on the permutation-based indexing technique introduced by Chavez *et al.* [44] in 2008. With some modifications, Murakami *et al.* [43] describe the idea of this permutation-based indexing in the following way: Firstly, with a distance measure, the system computes the distance between the defined pivots and the biometric templates. Secondly, a permutation of each template is defined where the pivots are sorted in ascending order according to the associated distance to the template. Thirdly, given an unidentified probe template, a corresponding permutation is constructed showing the ordering of the pivots from the lowest to the highest distance measured to the probe. Lastly, for each biometric template, an approximation score can be computed between the permutation of the probe template and the permutation of the given biometric template [43].

In order to make the indexes more secure, several changes are implemented to the permutation-based indexes. The proposed system in [43] further takes the inverse of the permutation. It uses a transformation to secure the indexes against leaking information about the original biometric template [43]. With this inverse permutation, the system computes the inner product between the probe index and each template index to measure an approximate distance [43]. By sorting the distances, the system computes an exact score and compares this against a pre-defined threshold to make a final decision for the biometric identification [43]. According to Murakami *et al.* [43], this indexing scheme outperforms existing indexing approaches. The size of the stored transformed indexes is relatively small, and the average computations are also significantly reduced [43]. Because of this, the proposed indexing scheme in [43] can be classified as both significantly efficient and secure.

Another related indexing scheme for biometric identification was recently proposed by Dong *et al.* [45]. Instead of the permutation-based indexing scheme in [43], Dong *et al.* [45] presented the concept of Learning-Based Index-of-Maximum (LIoM) hashing, which is based on projections and transformations. The LIoM hashing

is an improvement of the ordinary Index-of-Maximum (IoM) hashing technique from [46], which both utilized the concept of the Locality Sensitive Hashing (LSH) technique [45]. Based on a maximized probability that two items are similar, the purpose of LSH is to group similar items into the same "bucket" [47]. Since the ordinary IoM hashing was not optimized for utilizing the feature characteristics, the improved LIoM hashing replaced the use of random transformations with utilizing the feature values [45]. Due to this improvement, more compact hash codes could be produced for each feature vector, and hence, faster similarity matching using the Hamming distance could be achieved for two biometric features [45]. Despite the improvement, the LIoM hashing suffers from performance degradation due to the non-invertible transformation approach [45].

Chapter 4
Methodology

This Chapter describes the methodology of this thesis. The following Sections introduce the different phases the workload was divided into, while the following Chapters go into detail for each phase.

4.1 Literature Review

The initial work of the thesis consisted of acquiring knowledge about biometric identification and FHE through a literature review. This was necessary in order to study related information and concepts that were useful for the remaining workload of this thesis. The previous Chapters covered the presented result of the literature review, where the building blocks that comprise the different parts of the thesis field were presented in Chapter 2, and other related technologies were presented in Chapter 3. With this process, a literature search was performed through the relevant scientific dissemination channels for the research fields of biometrics and cryptography.

The remaining workload of the thesis consisted of transferring the information and knowledge that were acquired to achieve the goal of the thesis and answer the research questions presented in Section 1.2. This workload was divided into four phases: vulnerability analysis, construction and design, implementation and evaluation, and comparison and discussion. These phases are individually described in the following Sections.

4.2 Phase 1: Vulnerability Analysis

The initial phase consisted of observing vulnerabilities in the two existing systems in the focus of this thesis, stable hashing and PEKS-based preselection. It can be considered that almost every biometric system will present a trade-off between advantages and drawbacks that are specific to the presented approach. Therefore, it

was necessary to understand where the bottleneck was in order to be able to solve and understand the problem. The discoveries and the details of this vulnerability analysis are explained in Section 5.1.

4.3 Phase 2: Construction and Design

By studying the vulnerabilities in the previous phase, we constructed a picture of what needed to be improved. However, besides the potential drawbacks, the existing systems also had some potential benefits that we wanted to maintain to utilize and build on. Therefore, by utilizing a combination of the two existing systems and combining their benefits, we proposed and constructed a new system that desired to improve and remove the associated previous vulnerabilities. This was the focus of the construction and design phase, which is further explained in Section 5.2.

4.4 Phase 3: Implementation and Evaluation

With a new system, it was necessary to test and evaluate it to check if the previous vulnerabilities were removed and if the system fulfilled the desired goal of this thesis. The proposed system was implemented and evaluated with regard to computational efficiency, biometric performance, and security. To evaluate the biometric performance, we measured the accuracy of the proposed system. The details of the experiment and the evaluation outcome are further presented in Chapter 6.

4.5 Phase 4: Comparison and Discussion

In addition to performing an experimental evaluation of the proposed system, we also needed to provide arguments and discussion as to why the system worked and the reason behind its building blocks. For this purpose, it was also necessary to compare the proposed system with a baseline system. In addition to the baseline system, the proposed system was compared against the existing systems to see if there was an improvement based on the observed vulnerabilities presented in the initial phase. While the baseline system is presented in Section 2.2.4, the details of the comparison phase are further described and discussed in Chapter 7. The research questions are addressed in Chapter 7 as well.

Chapter 5
Proposed System

This Chapter introduces the proposed system for this thesis, describing a preselection method for biometric identification. Section 5.1 presents the motivation for the proposed system by describing observations and vulnerabilities in existing systems that we want to improve and eliminate with our proposed system. These vulnerabilities will be mitigated in the proposed system, which is described in Section 5.2.

5.1 Observations and Vulnerabilities From Existing Systems

The following Section describes the observed vulnerabilities from the two approaches, stable hashing [8] and PEKS-based identififcation [7], that this thesis is based on. For the proposed system for this thesis, these vulnerabilities have been addressed and attempted to solve through a combination of the benefits of both works. The following Section will show that the stable hashing approach [8] ensures the efficiency aspect, while the PEKS-based identification approach [7] ensures the privacy aspects. However, each of the two approaches is lacking the benefit of the other.

5.1.1 Stable Hash Vulnerabilities

Even though the biometric identification process using the stable hash generation approach from [8] is efficient, it needs to be more secure to ensure the appropriate level of privacy when dealing with biometric information. This Section will analyze and show why there are security concerns. This insecurity is associated explicitly with missing security properties and a deterministic approach.

Security Properties

Stable hash codes can be perceived at first sight as having relatively similar behavior to cryptographic hash functions. Both produce a fixed-length output that is deterministic, i.e., stable for each input. A stable hash code key is stored in a hash lookup table to retrieve the corresponding protected reference templates from

the same stable hash code [8]. On the other hand, given an input message x, a cryptographic hash function will retrieve the corresponding message digest $H(x)$ [48]. Despite this similar usage, the two approaches are relatively different regarding their security properties. As described by Sobti and Geetha [48], an essential property of a cryptographic hash function is the strong avalanche effect describing that similar input messages should result in significantly different output message digests [49]. This property is not present for the stable hash codes as similar input keys in the hash lookup table should result in similar output reference templates [8]. This is because of the clustering technique with the k-means approach, which groups similar objects. From an efficiency standpoint, it is therefore beneficial for the stable hash codes that a slight variation in input features will not cause a considerable variation in the resulting cluster center.

However, since the stable hash codes do not satisfy the same security properties as cryptographic hash functions, they may exhibit privacy concerns because they are not considered cryptographically secure. For instance, if the returning references from the candidate list are relatively small, there are potential risks of violating the unlinkability requirement, described in Section 1.1, since it may be possible to keep track of similar references. With this behavior, there may also be concerns regarding the possibility of revealing the original template from the stable hash code, which may indicate the actual person. The pre-project also elaborated on these issues with corresponding references [9].

Deterministic Approach

In order to store the stable hash codes more securely in a hash lookup table, Osorio-Roig *et al.* [8] proposed to use a cryptographic hash layer, e.g., SHA256, around the existing stable hash code. This was to prevent an attacker from being able to reconstruct the original facial image if any information from the stable hash codes is leaked [8]. Despite the attempt to produce more secure hash codes, the proposed scheme continues to experience vulnerabilities, as the scheme is still deterministic. A vulnerability with the deterministic scheme is that the same output is generated each time the same input value is used [50]. This behavior can cause problems regarding privacy and security if the input space is small [50], as is the case with the stable hash codes. The stable hash codes from [8] and the proposed use of the cryptographic hash function SHA256 behave according to the deterministic approach.

In order to ensure a privacy-preserving procedure for biometric identification, it will be necessary to eliminate the deterministic approach and instead add a non-deterministic scheme. A non-deterministic behavior produces a new outcome when the same input is provided [50]. This can be categorized as randomized encryption and is necessary in order to give the appropriate level of security, in particular with

regard to the unlinkability requirement of ISO/IEC 24745 [6].

5.1.2 Soft-Biometrics Vulnerabilities

Even though soft-biometrics can be used as a suitable preselection method to reduce the total search space for an identification process, they also introduce drawbacks. Firstly, there are problems with noise due to preselection errors. One reason for this is the identification errors arising from age estimation. As described by Dantcheva *et al.* [40], age estimation is challenging for humans and machines because of unknown features such as health and genetics. These unknown features may vary from individual to individual, making it harder to estimate the age [40]. Secondly, automated soft-biometric estimators can introduce unwanted bias to the biometric system. Friedman and Nissenbaum [51] define the term *algorithmic bias* as algorithmic system errors causing unfair outcomes for one or several individuals. Terhörst *et al.* [52] further point out that bias associated with sensitive attributes, such as soft-biometrics, can lead to different recognition performances and unfair impact for certain smaller population subgroups. This associated bias is due to the unequally distributed classes used for training purposes of recognition models [52]. Lastly, soft-biometrics can be a sensitive preselection method when the reference database is unbalanced. As discussed in the pre-project [9], the identification model experiences better performance when the probe template is associated with a few similar references in the enrollment database. This is also elaborated upon by Rathgeb *et al.* [53], who explain that a reason behind an unfair biometric system is often associated with biased training due to unbalanced data sets with regard to demographics.

5.2 Proposed System

This Section describes the proposed system for this thesis. In order to achieve an efficient and privacy-preserving preselection approach for biometric identification, the following proposed system focuses on replacing soft-biometrics with stable hash codes. This means that instead of using soft-biometric keywords as the PEKS keywords, the stable hash codes, i.e., cluster centers, are used as the keywords in the PEKS approach. The proposed system aims to remove the aforementioned vulnerabilities described in Section 5.1 and thereby be in compliance with the security requirements from the ISO/IEC 24745 standard on biometric information protection [6].

5.2.1 Architecture

For the architecture of the proposed system, we use a two-server model, as explained in Section 2.2.1, consisting of a computation server CS and an authentication server AS, where the CS communicates with a client C which provides the biometric data, such as the biometric features. While the C and the CS have access to the public

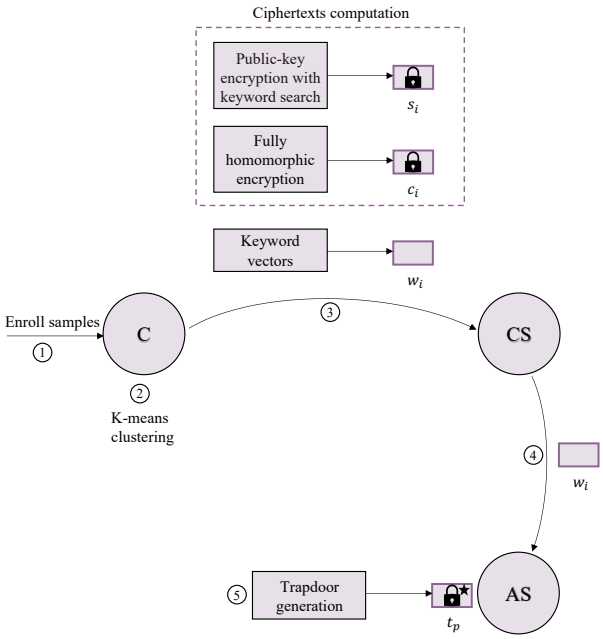


Figure 5.1: Figure showing the enrollment phase of the proposed system given N enroll samples.

key information, the AS only has access to the secret key. In addition, the CS has access to a protected reference database. We assume that a semi-honest model will be applied as the involved entities are expected to follow the protocol correctly [17].

5.2.2 Enrollment Phase

The enrollment phase is the initial phase before the identification process occurs and consists of different setup procedures. Figure 5.1 illustrates the enrollment phase, which can be divided into five Steps. In Step (1), C receives the enroll samples. These enrollment references are used in the k-means clustering technique in Step (2). The C will then, in Step (3), compute the keyword vector w_i associated with the k clusters. In addition, the C produces the searchable ciphertext s_i and the protected ciphertext c_i for each enrollment reference. The keyword vectors, w_i, and the ciphertexts, s_i and c_i, are sent to the CS. In Step (4), the k different keyword vectors w_i will be provided to the AS. Lastly, in Step (5), the AS will produce the trapdoors associated with the k different keyword vectors, i.e., based on the k clusters from the k-means clustering algorithm.

To further show the technical details behind the enrollment phase, Algorithm 5.1

Algorithm 5.1 Enrollment Phase

Input: N enrollment references: **enroll samples**
Output: k clusters: **cl**, k keyword vectors: **w**, N protected ciphertexts: **c**, N protected ciphertexts: **s**, k trapdoors: **t**

1: $cl \leftarrow KMeans(k).fit(enroll\ samples)$
2: **for** i `in` $\{0, .., k-1\}$ **do**
3: $\quad w_i \leftarrow RandomVector(i)$;
4: **end for**
5: **for** i `in` $\{0, ..., N-1\}$ **do**
6: $\quad c_i \leftarrow Enc(pk_{FHE}, i)$;
7: $\quad s_i \leftarrow PEKS(pk_{PEKS}, w_i)$;
8: **end for**
9: **for** i `in` $\{0, ..., k-1\}$ **do**
10: $\quad t_i \leftarrow Trapdoor(sk, w_i)$;
11: **end for**

shows the pseudocode of the technical functionality, where the input and output of the enrollment phase are indicated. In this phase, we enroll N references, referred to as *enroll samples*, into an enrollment database. In addition, we specify k as the number of clusters used for the preselection method. With the *enroll samples* dataset, the k-means clustering algorithm will be applied to train the dataset to define k clusters, called cl, as shown in Line 1. Each of the k clusters will consist of similar enrollment references, and the cluster indexes will be defined as an integer in the range 0 to $k-1$. To use the k cluster indexes as the PEKS keywords in the PEKS scheme, k fixed, random vectors $w_1, ..., w_k$, are created that are associated with each of the k clusters, as shown in Line 3. In Line 6 and Line 7, the enrollment phase produces protected and searchable ciphertext for each of the N enrollment references in the *enroll samples*. Using the plaintext template of an enrollment reference, a corresponding protected ciphertext can be produced using Equation (2.2) of the FHE scheme, as shown in Line 6. On the other hand, using the associated keyword of the enrollment reference, i.e., the associated cluster index from the k-means algorithm, a corresponding searchable ciphertext can be produced using the **PEKS** algorithm, Equation (2.7), from the PEKS scheme, as shown in Line 7. Lastly, using these k PEKS keyword vectors w_i, the enrollment phase produces k trapdoors based on the k clusters. For this purpose, the k trapdoors are computed using the **Trapdoor** algorithm, Equation (2.8), of the PEKS scheme, as shown in Line 10.

5.2.3 Identification Phase

At the time of the identification phase, the objective of the system is to decide whether a mated reference exists in the enrollment database for a given unidentified

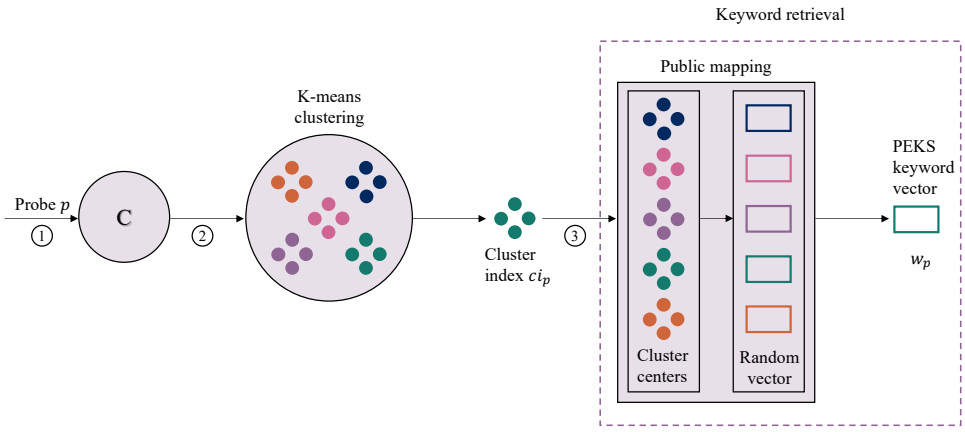


Figure 5.2: Figure showing the keyword vector retrieval process of the proposed system for a given probe sample p.

probe sample p. This process can be divided into two main Steps for our proposed system: i) keyword vector retrieval and ii) PEKS with FHE comparison. In the following Section, Steps i) and ii) are described with two overview Figures, Figure 5.2 and Figure 5.3, before a more comprehensive technical description of the entire identification phase is given in Algorithm 5.2.

Keyword Vector Retrieval

The first Step of the identification phase is the keyword vector retrieval, illustrated in Figure 5.2. This process is divided into three Steps. In Step (1), the C receives the given probe sample p. Then, in Step (2), the probe's corresponding cluster index, ci_p, is computed using the k-means clustering approach. Lastly, in Step (3), we utilize a public mapping between cluster centers and keyword vectors to obtain the corresponding PEKS keyword vector w_p of the probe p.

PEKS and FHE Comparison

The second Step of the identification phase is performing the PEKS and the FHE comparison to obtain a final identification decision of the given probe sample p, illustrated in Figure 5.3. This process is divided into eight Steps. In Step (1), the C receives the probe sample p to be identified along with its corresponding keyword vector w_p, which was the outcome of the keyword vector retrieval process. Using the probe p, the C will, in Step (2), use FHE to compute a protected ciphertext c_p of the probe p. At the same time, the C will also use the keyword vector w_p to compute the searchable ciphertext s_p of the probe p. With these ciphertexts, the

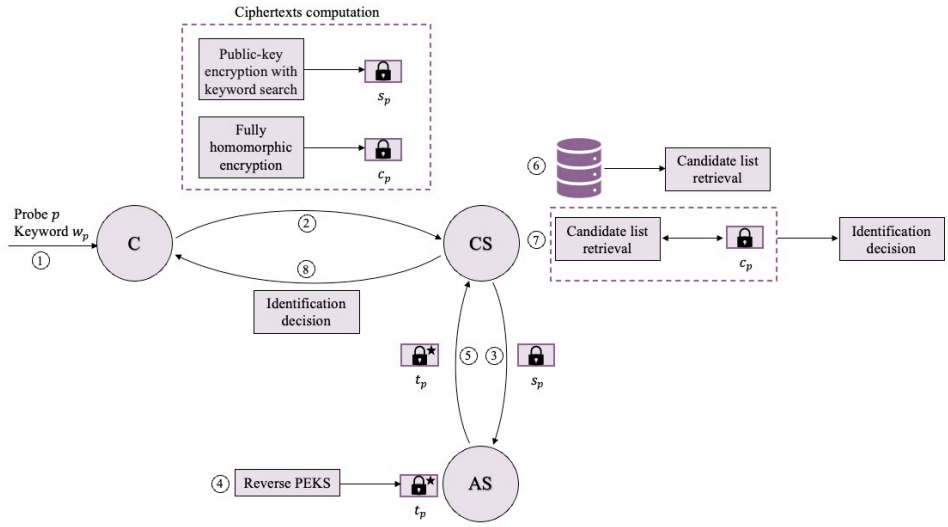


Figure 5.3: Figure showing the identification process with PEKS search and FHE comparison for a given probe sample p and its corresponding keyword vector w_p.

C will then forward the protected ciphertext c_p and the searchable ciphertext s_p to the CS. After obtaining the s_p, the CS will, in Step (3), forward the s_p to the AS. In Step 4, the AS will perform the reverse PEKS search, described in Section 2.4.4. This process is necessary to obtain the corresponding trapdoor t_p of the probe p.

After finding the probe trapdoor t_p, the AS will then, in Step (5), forward the probe trapdoor t_p back to the CS. With access to the probe trapdoor t_p, the CS will, in Step (6), obtain a candidate list consisting of the enrollment references with the same PEKS keyword vector as the probe p, i.e., have the same cluster index. In Step (7), the CS will then compare the protected ciphertext c_p of the probe p against the protected references in the candidate list in order to make a final identification decision if there is a mated reference for the probe p. Lastly, in Step (8), the CS will then forward the final identification decision back to the C and completes the identification phase.

Technical Description

To further show the technical details behind the identification phase, Algorithm 5.2 shows the pseudocode of the technical functionality, where the input and output of the entire identification phase are indicated. Lines 1–2 summarizes the functionality behind the keyword vector retrieval process from Figure 5.2. In Line 1, the associated

Algorithm 5.2 Identification Phase

Input: Probe sample: **p**
Output: Identification decision: **ID**

1: $ci_p \leftarrow \min_{dist} Euc\_dist(cl, p)$
2: $w_p \leftarrow RandomVector(ci_p)$
3: $c_p \leftarrow Enc(pk_{FHE}, p)$
4: $s_p \leftarrow PEKS(pk_{PEKS}, w_p)$
5: **for** i `in` $\{0, .., k-1\}$ **do**
6: `Find` $t_p = t_i$`:` $t_p \leftarrow Test(t_i, s_p) = 1$;
7: **end for**
8: **for** i `in` $\{0, .., N-1\}$ **do**
9: `Find candidatelist:` $Test(t_p, s_i) = 1$;
10: **end for**
11: **for** i `in` $\{0, .., M-1\}$ **do** ▷ *M is the length of* `candidatelist`
12: $dist_i \leftarrow Euc\_dist(c_p, c_i)$;
13: **if** $dist_i > \delta$ **then**
14: $threshold\_list[i] = (i, dist_i)$;
15: **end if**
16: **end for**
17: $ID \leftarrow \min_{dist} threshold\_list$;

cluster index ci_p of a given probe sample p is computed by measuring the squared Euclidean distance, Equation (2.4), between the probe sample and all the defined cluster centers, cl, from the k-means algorithm of the enrollment Step (Line 1, Algorithm 5.1). The associated cluster index of the probe ci_p is the cluster with the minimal distance. To define the corresponding PEKS keyword vector of the probe, Line 2 extracts out one of the k random, fixed vectors from the enrollment Step (Line 3, Algorithm 5.1) associated with the same cluster index as the probe p.

The following Lines 3–17 summarize the functionality of the PEKS and FHE comparison from Figure 5.3. Using the probe sample template p and the public key pk_{FHE}, Line 3 computes the protected ciphertext c_p of the probe using Equation (2.2) from the FHE scheme. On the other hand, using the PEKS keyword vector w_p of the probe and the public key pk_{PEKS}, Line 4 computes the searchable ciphertext s_p of the probe using the **PEKS** algorithm, Equation (2.7), from the PEKS scheme. Based on the searchable ciphertext s_p of the probe, Line 6 runs the reverse PEKS scheme to find the corresponding probe trapdoor index t_p. For this probe trapdoor computation, the **Test** algorithm, Equation (2.9), of the PEKS scheme is used. In this process, the searchable ciphertext s_p is compared against each of the precomputed k trapdoors, t_i, from the enrollment Step (Line 10, Algorithm 5.1). The probe trapdoor t_p will be the given trapdoor where the **Test** algorithm outputs true. Having the probe

trapdoor index t_p, Line 9 also utilized the **Test** algorithm to obtain a candidate list. The candidate list is obtained by comparing the probe trapdoor t_p with all the searchable ciphertexts, s_i, from the enrollment Step (Line 7, Algorithm 5.1). In this way, the **Test** algorithm can find those subject references with the same keyword vector as the probe, i.e., are assigned to the same cluster as the given probe. The candidate list will then consist of the M subject references where the **Test** algorithm outputs true.

For each of the M subject references in the candidate list, Line 12 computes the Euclidean distance between the protected ciphertext c_p of the probe and the protected ciphertext c_i of the subject reference in the candidate list. The M protected ciphertexts c_i are extracted from the protected enrollment database that was computed during the enrollment Step (Line 6, Algorithm 5.1). Lastly, to find the resulting subject most similar to the probe, Line 13 evaluate the Euclidean distances from Line 12 against a threshold value, δ. If the Euclidean distance is below δ, the candidate subject and the corresponding Euclidean distance to the probe are added to a threshold list, as indicated in Line 14. The threshold value, δ, is used as an upper bound to characterize two samples as mated or non-mated references based on their comparison score, i.e., their Euclidean distance. After iterating through all the candidate subjects, Line 17 extracts the subject id, referred to as ID, with the minimal Euclidean distance from the threshold list. If this ID from the threshold list is identical to the subject id of the probe sample, a mated reference is found, and the identification is successful. Otherwise, the proposed system has incorrectly identified the probe, i.e., the identification process could not find a mated reference of the probe with the given candidate list.

5.2.4 Binning Method and Coefficient Packing

The proposed system can be further improved by utilizing two approaches. The first approach is a binning method, which will improve the candidate list retrieval in Line 9, Algorithm 5.2. The first time a probe reference with a new keyword vector is used as input in the proposed system, the CS will perform the ordinary candidate list search by searching through all the N enrollment references having the same keyword vector as the probe. The protected references of these candidate subjects will then be stored in a separate and fixed bin. The next time the CS receives a probe reference with a known keyword vector that has been used before, it will remember the candidate references and use the same bin. This will improve the preselection process and can be considered an engineering decision. As discussed in Section 2.4.4, Bauspieß *et al.* [7] also used a binning method on the enrollment references before the identification search. Instead of using this complex binning approach to sort the references into equal bins according to the keyword vector distribution, the proposed binning method utilizes the clusters from the k-means clustering technique to group

similar subjects. Each cluster is then treated as a separate bin. With k possible clusters, k fixed bins will be iteratively constructed as a new keyword vector is used as the input keyword for an unidentified probe reference.

The second approach for improving the computational efficiency of the proposed system is utilizing coefficient packing, as described in Section 2.2.5. This method is used in the implementation and will improve the comparison time between the probe reference and each candidate subject. When the protected reference templates are fixed in the bins, coefficient packing can be used to concatenate the templates to reduce the number of comparisons. In addition, when producing the protected ciphertext of the probe, the probe reference can be concatenated with itself several times in order to fit the entire template slots. This is necessary in order to compare the concatenated candidate templates with the concatenated probe reference. This procedure has been established and evaluated in [30].

Chapter 6
Experimental Evaluation

This Chapter describes the experimental evaluation of the proposed system. Section 6.1 and Section 6.2 give an introduction to the databases and different metrics that were used in the experiments. Section 6.3 describes the structure of the implementation code. Lastly, Section 6.4 shows the results of the experiments.

6.1 Databases

For the experimental evaluation of the proposed system presented in Section 5.2, two databases have been used. These are the Face Recognition Technology (FERET) database [54] and the Face Recognition Grand Challenge version 2 (FRGCv2) database [55].

6.1.1 FERET Database

The FERET database [54] that was used in the experimental evaluation consisted of 1413 facial images from 529 subjects. For each subject, multiple samples were taken covering their frontal views, quarter profile, and different backgrounds [54]. The number of samples per subject varied from two to four samples.

6.1.2 FRGCv2 Database

The FRGCv2 database [55] was larger than the FERET database, consisting of 3165 facial images from 533 subjects. The different images contained samples of facial images with different expressions under various lighting conditions [55]. Overall, the number of samples per subject was also higher than in the FERET database and varied from two to nine samples. This gave more samples to associate with each subject for the proposed system evaluation.

6.2 Metrics

For later use in the experimental evaluation, this Section describes relevant metrics in accordance with the ISO/IEC 19795-1 [13] standard on biometric performance testing:

- *Penetration Rate*: Indicates the average number of candidates in the preselection list, i.e., in the candidate list.
- *Preselection Error Rate*: Indicates the percentage of the database where the preselection candidates do not include the corresponding subject identifier.
- *False Positive Identification Rates (FPIR)*: Indicates the percentage of the database where subject references not enrolled in the database nevertheless get returned a reference identifier after identification. This implies a false positive result where the system proposes that the given subject has a mated reference in the database, despite this not being the case.
- *False Negative Identification Rates (FNIR)*: Indicates the percentage of the database where subject references are enrolled but are not receiving a reference identifier after the identification process. This implies a false negative result where the system cannot propose a mated reference for the corresponding subject, despite this not being the case.
- *True Positive Identification Rate (TPIR)*: Indicates the percentage of the database where subject references enrolled in the database receive the correct reference identifier after the identification process. This will illustrate the accuracy of the identification process, i.e., how many subject references are correctly identified.
- *True Negative Identification Rate (TNIR)*: Indicates the percentage of the database where subject references not enrolled in the database do not receive a reference identifier after the identification process.
- *Detection Error Trade-off (DET)*: Indicates the connection between FPIR on the horizontal axis and FNIR on the vertical axis.

6.3 Structure of the Code

For the experimental evaluation, three code bases have been used to implement the proposed scheme in Section 5.2. The code written for this thesis can be found in the following repository: https://github.com/ceciliefo/Master_thesis. This Section presents each code base and which additional libraries were used.

6.3.1 Stable Hash Generation Code

The code for the stable hash generation was based on the original code used in [8]. For executing the code, Visual Studio Code[1] was used. The implementation was performed in Python and used the `KMeans` function from the machine learning package Scikit-Learn [56]. This `KMeans` function was used for the k-means algorithm to produce the stable hash codes, i.e., the cluster centers of all the references in the given databases described in Section 6.1. The stable hash generation code was used in Line 1 in Algorithm 5.1 for the enrollment phase and in Line 1 in Algorithm 5.2 for the identification phase.

6.3.2 PEKS Code

The code for performing the PEKS with the stable hash codes was based on the original code used in [7] and the available C++ code for PEKS on GitHub[2]. For executing the code, the CodeLite IDE[3] was used. The PEKS code was used in Line 3, Line 7 and Line 10 in Algorithm 5.1 for the enrollment phase, and in Line 2 and Lines 4–10 in Algorithm 5.2 for the identification phase.

6.3.3 FHE Comparison Code

Like the PEKS code, the last code regarding the FHE comparison and the final identification decision also used the CodeLite IDE and was written in C++. For the FHE comparisons, the Open-Source Fully Homomorphic Encryption (OpenFHE) Library [57] has been used. The FHE code was used in Line 6 in Algorithm 5.1 for the enrollment phase and in Line 3 and Lines 11–17 in Algorithm 5.2 for the identification phase.

6.4 Results

The following Section presents the execution parameters and the experimental evaluation results. The execution times for the proposed system were measured with the following specifications:

- Operating System: macOS Monterey.
- Version: 12.4.
- CPU: Apple M2 @ 3.50 GHz.
- RAM: 8 GB.

[1]Visual Studio Code: https://code.visualstudio.com/
[2]GitHub repository for PEKS scheme: https://github.com/Rbehnia/NTRUPEKS
[3]CodeLite IDE: https://codelite.org/

Regarding the biometric data, floating point feature representations extracted through the open-source face recognition ArcFace [11] were used. The performance evaluation was tested in a closed-set scenario, meaning every probe reference had a corresponding mated reference in the enrollment database.

6.4.1 Execution Parameters

For the experimental evaluation, the following parameters have been used:

- Sub-Spaces, P: Defines the number of equal sub-spaces that one feature representation set is divided into [8]. This is necessary for the stable hash generation scheme. For the experiments, $P = 1$ has been used.
- K-Means Clusters, k: Defines the number of clusters created through the k-means clustering technique. For the experiments, $k = 64$ has been used.
- Feature Space Dimensionality, D: Defines the number of dimensions for representing the biometric data for one feature representation, i.e., for one biometric subject sample. For the experiments, $D = 512$ has been used.
- FHE scheme, $fheScheme$: Defines the following FHE scheme for the experimental evaluation. For the experiments, $fheScheme = CKKS$ has been used. The CKKS scheme was used for the FHE computation as the biometric data was represented with floating point values. In addition, the CKKS scheme was implemented with a variant of Residue Number System (RNS) to improve the efficiency of the FHE operations [57].
- CryptoContext: With OpenFHE, a CryptoContext environment needs to be created before the FHE computations. The CryptContext manages all the OpenFHE objects, for instance, the key pair generation [57]. The CryptoContext was implemented with the CKKS schemes and was specified with the following parameters for the experiments: $multDepth$, $scaleFactorBits$, $batchSize$, and $securityLevel$.
- Multiplicative Depth, $multDepth$: Defines the maximum multiplication depth. This is not the same as the maximum number of multiplications that are supported for the entire scheme [57]. The multiplication depth is only defined for one given multiplication [57]. As explained in [57], a shorter multiplicative depth is often preferable for better performance. Because of this, the experiments used $multDepth = 1$. The computation of the squared Euclidean distance only required one multiplication, so $multDepth = 1$ is the lowest possible choice.
- Scaling Factor Bits, $scaleFactorBits$: Defines the scaling factor in bit-length from encoding real numbers into integers with the CKKS computation [57]. The

scaling factor will affect the accuracy of the computations. With a lower scaling factor, the execution time will enhance, but only until a certain point. The aim is to achieve a valid accuracy for a low execution time without increasing the approximation errors [57]. For the experiments, $scaleFactorBits = 30$ has been used.

- Plaintexts Slots in the Ciphertext, *batchSize*: Defines the number of plaintexts that are combined into one ciphertext [57]. For the experiments, $batchSize = 4096$ has been used. That means that with each feature representation with the dimensionality of 512, the available number of plaintext slots in one ciphertext is $\frac{4096}{512} = 8$.
- Security Level, *securityLevel*: Defines the security level for the FHE scheme. The most commonly used security values are 128-bit, 192-bit, and 256-bit. For the experiments, *securityLevel* = 128-bit has been used.

6.4.2 Computational Cost of the Baseline System

Before evaluating the proposed system, we examined the computational cost of the baseline system, performing an exhaustive search over the entire protected reference database, as presented in Section 2.2.4. For measuring the computational cost, we used the concept of coefficient packing and the CKKS parameters as presented in Section 6.4.1. With the batch size of the ciphertext set to 4096, we can concatenate eight ciphertexts of dimension 512 into one "full" ciphertext. This is because $\frac{4096}{512} = 8$. Since we can concatenate eight reference ciphertexts, we only needed 67 ciphertexts to represent the FRGCv2 enrollment database with 533 subjects. This is because $\frac{533}{8} = 66.63$. As described in Section 2.2.2, we performed one subtraction, one multiplication, 511 rotations, and 511 addition operations for the homomorphic encryption comparison. This was performed for each of the 67 packed ciphertext vectors. The 511 rotations and additions are due to the feature dimensionality of $D = 512$ and $D - 1$ rotations and additions are required for the computation of the squared Euclidean distance [30]. By measuring this computational cost, it took $354,845$ms to perform the biometric identification process of the baseline system.

6.4.3 Keyword Vector Distribution

Preselection is now introduced on top of the baseline system, as described in the proposed system in Chapter 5. Figure 6.1 illustrates the distribution of subjects in each cluster for the FRGCv2 database. The number of subjects varied from a minimum of three to a maximum of 15 subjects, where the average number of subjects in a cluster was $\frac{533}{64} = 8.33$. For the FERET database, the distribution of subjects to clusters is not shown in this thesis, but the number of subjects in each cluster varied from a minimum of two to a maximum of 17 subjects.

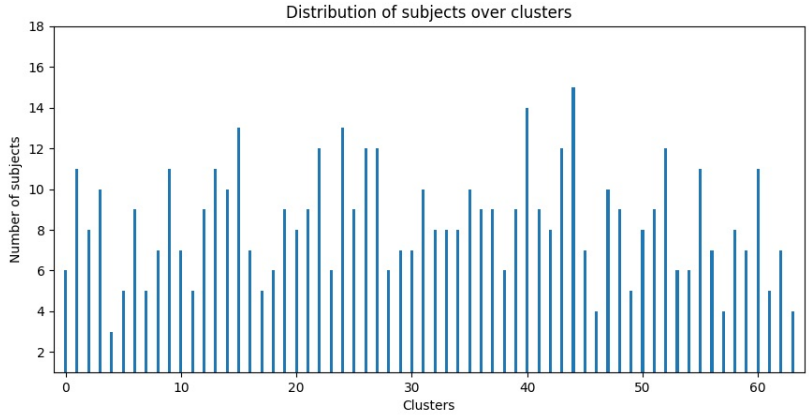


Figure 6.1: Figure showing the keyword vector distribution for the number of subjects in each cluster for the FRGCv2 database [55].

Table 6.1: A Table showing the accuracy of the k-means clustering with $k = 64$ clusters for each database.

Database	Enroll Samples	Search Samples	False Negative	True Positive	Accuracy
FERET [54]	529	884	19	865	0.9785
FRGCv2 [55]	533	2632	207	2425	0.9214

6.4.4 Accuracy of the Stable Hash Generation

The results for the k-means clustering accuracy of the stable hash generation are illustrated in Table 6.1 for the databases FERET and FRGCv2.

For each database, the reference samples were divided into two parts: *enroll samples* consisting of the first sample of every subject in the database and *search samples* containing the remaining samples of every subject. While the *enroll samples* was used to train the k-means model during the enrollment phase to define k cluster centers, as explained in Section 5.2.2, the *search samples* was used to evaluate the accuracy of the stable hash generation procedures, i.e., the clustering technique. With the dataset from the *search samples*, the stable hash generation code computed the corresponding cluster index for each reference sample in the *search samples*. These cluster indexes were obtained by computing the Euclidean distance between the search samples and the cluster centers. For each search sample, the distances to

the cluster centers were stored in a separate list, referred to as the *distance list*. By appending an index value to every distance value in the *distance list*, we could keep track of which distance was associated with which cluster center. After sorting this *distance list*, the minimal distance value could be extracted. The search sample's corresponding cluster index was the original index associated with the minimal distance value from the *distance list*. In order to evaluate the accuracy of the cluster technique, the code compared if the subject samples in *search samples* were mapped to the same cluster index as the corresponding subject in *enroll samples*.

Based on this, we can use the following Equation to compute the accuracy:

$$Accuracy = \frac{Search\ samples - False\ negative}{Search\ sample}. \tag{6.1}$$

For the FERET database, 19 subject samples from the search samples were not associated with the same cluster as their corresponding subject identifier in enroll samples. Hence, the measured accuracy of clustering with the FERET database is $\frac{884\ -\ 19}{884} = 0.9785$. For the FRGCv2 database, this accuracy is lower because the number of false negatives and search samples is higher than for FERET: $\frac{2632\ -\ 207}{2632} = 0.9214$. With more search references to test the clustering mechanism, as in the case of the FRGCv2 database, there is a higher probability that some subjects are classified to a different cluster than the enrollment references are associated with, thereby indicating lower accuracy.

From the defined accuracy for FERET and FRGCv2 databases, we can measure the number of preselection error rates in the following way:

$$Preselection\ error\ rate = 1 - Accuracy = \frac{False\ negative}{Search\ sample}. \tag{6.2}$$

For the FRGCv2 database, the preselection error rate is $\frac{207}{2632} = 0.0786$, meaning that 7.86% of the database were not included in the preselection candidate list. On the other hand, there were only $\frac{19}{884} = 0.0215$ preselection errors for the FERET database.

6.4.5 Execution Time for the Stable Hash Generation

In order to measure the efficiency of the clustering technique, we measured the time for the stable hash generation, as shown in Table 6.2. After k clusters were created from the enrollment references, in 1182.00ms, we measured the time to compute the corresponding stable hash code, i.e., the cluster index of a given probe sample. After running 1000 executions, the median time for computing the cluster index of a random probe sample was 0.28ms.

Table 6.2: A Table showing the execution times in milliseconds for the stable hash generation process.

Functionality	Execution Times (ms)
Enrollment features stable hash generation	1182.00
Probe stable hash generation	0.28

6.4.6 Execution Times for the PEKS Search

For the remaining execution times, only the clusters from the FRGCv2 database have been used. This is because the cluster sizes between FERET and FRGCv2 databases are relatively similar and will therefore not affect the execution times for the PEKS and FHE comparison.

For measuring the execution of the PEKS search, different functionalities were measured, as illustrated in Table 6.3. With $k = 64$ clusters from the k-means clustering technique, we initially needed to compute 64 trapdoors, one per keyword cluster. After running 1000 executions of the trapdoor generation function (Line 10, Algorithm 5.1), the median time for one execution was 2.74ms. This means the initial phase of performing 64 trapdoors takes approximately $2.74 \times 64 = 175$ms. In addition to the trapdoor generation, the initial phase of the PEKS search also consisted of constructing one searchable ciphertext for each enrollment reference in the database. While it took 0.27ms to compute one searchable ciphertext, the total execution time of computing searchable ciphertexts for 533 subjects for FRGCv2 is approximate $0.27 \times 533 = 144$ms. Lastly, the median execution time for performing the functionality of the reverse PEKS search took 7.69ms in order to compare the searchable ciphertext of the probe with $k = 64$ cluster trapdoors. In addition to these different functionalities of the PEKS scheme, it was also interesting to measure the execution time for finding the candidates with the same keyword as the given probe. Using the FRGCv2 database and the worst-case cluster with 15 subjects, the candidate list retrieval took 62.30ms. The CS uses 62.30ms the first time a new probe keyword vector is used in the identification system to find the associated candidates. This execution time will not be performed for every identification because of the binning method described in Section 5.2.4. The CS will remember the associated bin for a known keyword vector and will therefore save this candidate list retrieval time.

6.4.7 Execution Times for the FHE Comparison

With the FHE comparison, we want to measure the execution time for the worst-case candidate list. This means we only measure the execution time for one of the candidate lists from the PEKS search having the most candidates in one cluster. The execution

Table 6.3: A Table showing the execution times in milliseconds for different functionalities of the PEKS search.

PEKS Functionalities	Execution Times (ms)
Trapdoor generation	2.74
Searchable ciphertext encryption	0.27
Reverse PEKS search	7.69
Candidate list retrieval	62.30

Table 6.4: A Table showing the execution times in milliseconds for different functionalities of the FHE comparison and identification decision for the worst-case scenario with 15 candidates.

FHE Functionalities	Execution Times (ms)
Encrypted database setup (without coefficient packing)	3037.00
Encrypted database setup (with coefficient packing)	383.00
Probe encryption	2.00
Identification decision	9996.00

times using the FRGCv2 database and the worst-case scenario with 15 candidates are illustrated in Table 6.4. By utilizing the coefficient packing technique with the given parameters for the dimensionality and the batch size, we can concatenate eight ciphertexts. For a combined database of FERET and FRGCv2, the total size is 1062 subjects. Without using the coefficient packing technique, the setup of the protected database, i.e., creating protected ciphertexts using the FHE scheme for each of the 1062 subjects, took 3037ms. On the other hand, by utilizing the advantages of the coefficient packing technique, we can concatenate eight protected ciphertexts and hence only need 133 ciphertexts. This is because $\frac{1062}{8} = 132.75$. With only 133 ciphertexts, we could set up the protected database after 383ms. This is approximately eight times faster than the setup time for the protected database without using the coefficient packing technique. In addition to the database setup, it took 2ms to produce a protected ciphertext for one probe reference.

By performing the FHE comparison for 1000 executions, the median time for performing the final identification decision, i.e., comparing the protected ciphertext of a probe against the protected ciphertexts of the candidates in the candidate list, took 9996ms, using coefficient packing on both the probe and the candidate references.

Table 6.5: A Table showing the execution times in milliseconds for identifying a probe sample p compared with the execution time for the baseline system.

Functionality	**Execution Times (ms)**
Probe stable hash generation	0.28
Probe encryption	2.27
Reverse PEKS search	7.69
FHE comparisons	9996.00
Total	10,006.24
Baseline (exhaustive search)	354,845.00

6.4.8 Total Execution Time for the Identification Process

Based on the aforementioned execution times of the proposed system's different functionalities and building blocks, Table 6.5 shows only the necessary execution times for identifying a probe reference, without the setup times. The different functionalities and execution times are based on the keyword vector retrieval process, Figure 5.2, and the PEKS and FHE comparison, Figure 5.3. The probe encryption execution time of 2.27ms was the combined time of the probe encryption using FHE and the searchable ciphertext encryption of the probe using PEKS. From Table 6.5, the total execution time for running the proposed system for biometric identification of a given probe sample, i.e., finding a mated reference, was $10,006.24$ms.

6.4.9 Comparison Scores

To evaluate the biometric performance, i.e., accuracy, we computed comparison scores for mated and non-mated sample references in the database. A mated comparison score between two samples means that the samples are from the same subject, while a non-mated comparison score means that the samples are from two different subjects [13]. To compute the comparison score, the Euclidean distance function, Equation (2.3), has been used between every database sample. Table 6.6 shows the number of observations, maximum and minimum, and average comparison scores for mated and non-mated pairs of the FERET and FRGCv2 databases.

Table 6.6 can be visualized into a histogram plot showing the distribution of mated and non-mated comparison scores, as illustrated in Figure 6.2 for the FERET database and Figure 6.3 for the FRGCv2 database. The histogram plots in Figure 6.2 and Figure 6.3 were normalized because the number of observations of non-mated

Table 6.6: A Table showing different statistics of the comparison score values for each database.

	FERET [54]		**FRGCv2 [55]**	
Statistics	**Mated**	**Non-Mated**	**Mated**	**Non-Mated**
Observations	2656	1,992,500	17,766	9,996,294
Minimum	0.06	1.01	0.12	0.68
Maximum	0.86	1.65	1.02	1.66
Mean	0.47	1.40	0.63	1.40
Standard deviation	0.13	0.06	0.14	0.07

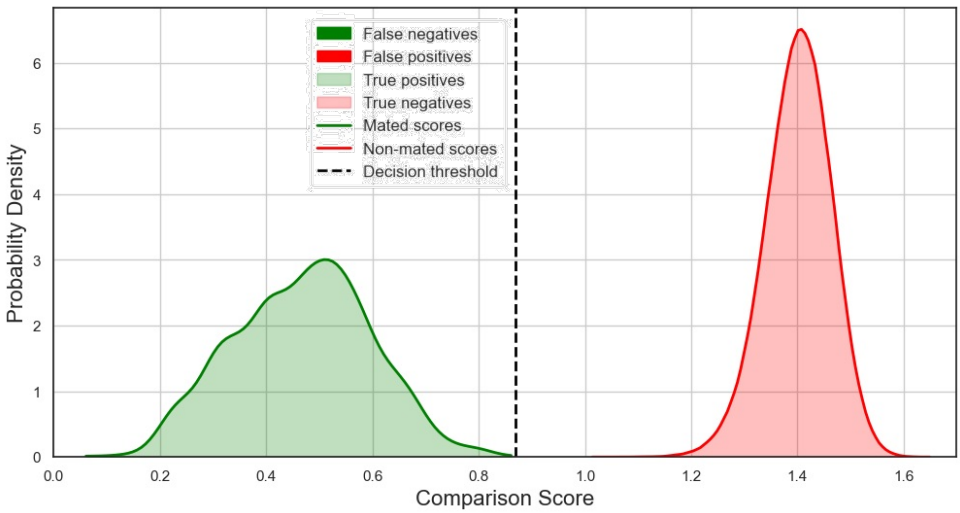


Figure 6.2: Figure showing the comparison scores for mated and non-mated reference samples with a threshold value set at 0.87 for the FERET database [54].

comparisons was significantly higher than the number of mated comparisons [13], as illustrated in Table 6.6. When using the Euclidean distance to compute the comparison scores, this can be characterized as a "dissimilarity" score type since we are interested in the distance between two samples, i.e., their dissimilarity [13]. This means mated comparison scores are closer to 0, while non-mated comparison scores are higher, as shown in Figure 6.2 and Figure 6.3.

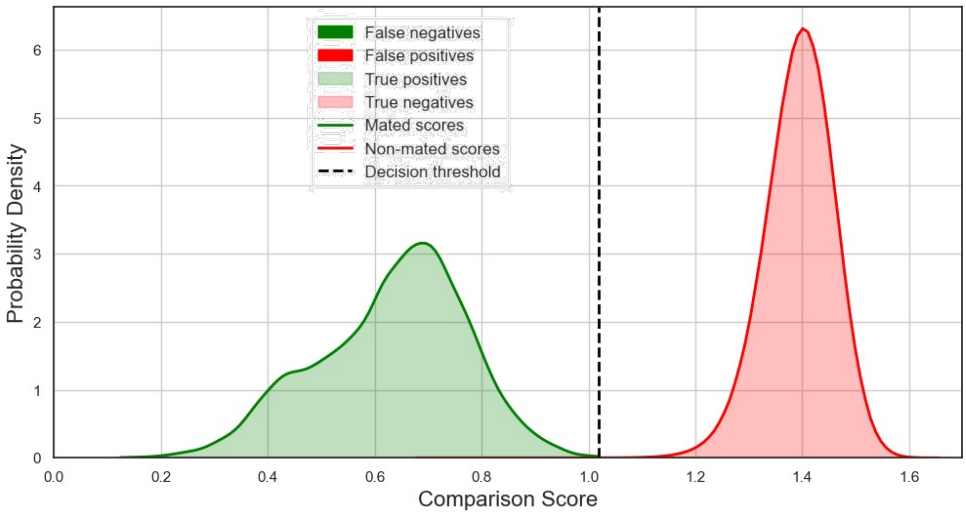


Figure 6.3: Figure showing the comparison scores for mated and non-mated reference samples with a threshold value set at 1.02 for the FRGCv2 database [55].

Table 6.7: A Table showing the errors of false positive and false negative for a specified threshold value of each database.

FERET [54] **Threhold = 0.87**			**FRGCv2 [55]** **Threshold = 1.02**		
Predicted/Actual	**Mated**	**Non-Mated**	**Predicted/Actual**	**Mated**	**Non-Mated**
Mated	100%	0%	Mated	100%	0.00297%
Non-mated	0%	100%	Non-mated	0%	99.997%

Using the histogram plots of the comparison scores, we could identify a threshold value to distinguish two samples as either a mated or a non-mated pair only based on their comparison score value. This threshold value will be the δ parameter from Line 13, Algorithm 5.2. Figure 6.2 and Figure 6.3 show the lowest threshold value for FERET and FRGCv2 databases resulting in a minimal amount of false positives and false negatives. Table 6.7 illustrates the number of prediction errors and correctness for the given threshold value of FERET and FRGCv2 databases. The FERET database was assigned a threshold value of 0.87, while the FRGCv2 database was assigned a threshold value of 1.02. A mated comparison score above the threshold value is considered a non-mated comparison score, increasing the FNIR. On the other hand, a non-mated comparison score below the threshold value is considered a mated

comparison score and hence increases the FPIR. True positive and true negative comparison scores indicate that the mated and non-mated comparison scores do not appear as a different comparison score type. From Table 6.7, using the threshold of 1.02 for the FRGCv2 database, the FPIR was 0.00297%, while the FNIR was 0%. On the other hand, the TPIR was 100%, while the TNIR was 99.997%. For the FERET database using the threshold of 0.87, the FNIR and FPIR was 0%, while the TPIR and TNIR was 100%.

6.4.10 DET Curve

The comparison score distribution in Figure 6.2 and Figure 6.3 only indicated the prediction errors, i.e., the FPIR and FNIR, for one threshold value. To indicate the prediction errors for the entire database, we could use a DET curve to evaluate the biometric performance [13]. Figure 6.4 shows the DET curve for biometric identification using the FRGCv2 and FERET database with the baseline system with no preselection approach, as described in Section 2.2.4. We only used the baseline system because the biometric performance of the biometric identification on the database was not expected to be significantly improved by using the proposed system with a preselection approach. The proposed system was used to improve the workload effort and the security and privacy aspects. As illustrated in Figure 6.4, the DET plot for the FERET database is empty. This will be discussed and explained in Chapter 7.

From Figure 6.4, we could compare the FPIR and FNIR in the databases. The DET curve has been constructed by utilizing every comparison score of the database as a threshold value and then measuring the FPIR and FNIR to illustrate the accuracy. The FPIR is the most critical aspect of the system, as it expresses the percentage of false positives or zero-effort imposters. Therefore, a lower FPIR rate implies a better biometric performance. For instance, if we consider an airport control example, the airport database consists of subjects permitted to enter the country. For a subject not enrolled in this database, we do not want the identification process to identify the subject as another person enrolled in the airport database, thereby giving illegitimate permission to enter the country. Figure 6.4 shows that the FPIR for the FRGCv2 database was very low, at 0.001%, indicating good system accuracy. For false positive values around 0.001%, the FNIR was also low, from 1% to 0.001%.

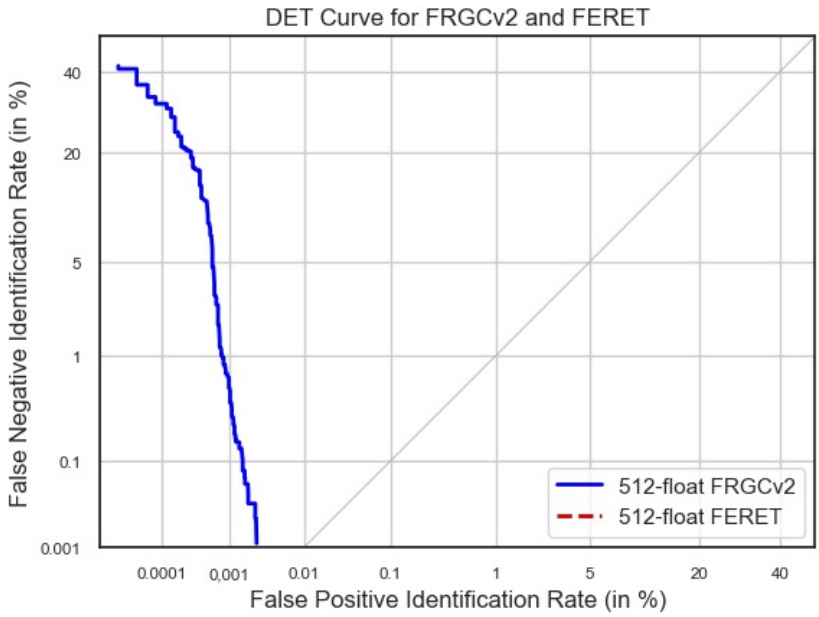


Figure 6.4: Figure showing the DET curve for the baseline system of the biometric identification using the FRGCv2 database [55] and the FERET database [54].

Chapter 7
Discussion

This Chapter presents arguments behind the proposed system and its building blocks, in combination with answering the initial research questions. Lastly, this Chapter discusses the results of the experimental evaluation.

7.1 Argumentation on the Proposed System

In Section 1.2, RQ1 and RQ2 were described concerning the challenges of soft-biometrics and if any alternatives could replace its usage. These issues have been addressed by looking at the described vulnerabilities in Section 5.1.2 and the proposed system presented in Section 5.2. The benefit of the proposed system is that by replacing soft-biometrics with stable hash codes, the challenges of bias and unfair identification outcomes can be decreased. In addition, by using the stable hash codes in combination with PEKS, the system will exploit the efficiency aspect of the stable hashing approach and solve the problem with the deterministic approach. By utilizing a non-deterministic approach with PEKS and FHE, the security of the proposed system has been increased. This is because both PEKS and FHE satisfy the requirements for a non-deterministic behavior due to their use of a randomized factor added to their scheme, which enables randomized encryption. For instance, using the PEKS algorithm, Equation (2.7), a new searchable ciphertext is produced each time, even though the same keyword is provided as input. The non-deterministic quality of PEKS and FHE also ensures unlinkability and renewability, and therefore fulfills ISO/IEC 24745 [6]. In addition, PEKS and FHE utilize the R-LWE problem, and thereby, we achieve irreversibility with post-quantum security.

Another benefit of the proposed system is that it can be argued that the system uses a generic framework to enable privacy-preserving biometric identification. This means that any efficient but insecure clustering approach can be combined with the security of PEKS and FHE to secure the framework.

7.1.1 Trade-Offs in Biometric Clustering

With any clustering technique, it is likely that the problem of clusters leaking information about the references occurs. This could be a problem if the clusters are small and relatively different in their size. There could also be a problem with roughly expecting that a percentage of the population will be expected in the given clusters, primarily because of their sizes. For instance, if a small percentage of the population is of Asian ethnicity, there is a high probability that one of the smallest clusters will consist of references of Asian ethnicity.

By using the k-means clustering technique, this process also inherits the bias of the underlying feature extraction algorithm [11]. The ideal approach is that every cluster has the same distribution as the database distribution. This means that if the database consists of 20% Asian ethnicity and 80% Caucasian ethnicity, then this is expected to be reflected in the clustering distribution with the k-means algorithm. A balanced database is rarely the case when considering biometric systems; therefore, some clusters will be larger than others.

7.1.2 Binning Method Argumentation

Regarding RQ4 (Section 1.2), there was uncertainty concerning if a binning method was necessary for the proposed method of this thesis. Because our proposed system replaced soft-biometrics with stable hash codes, the defined binning method for soft-biometric purposes [7] was no longer necessary. However, as described in Section 5.2.4, a different binning method could be applied. Even though the proposed system has removed the use of soft-biometric keywords, a binning method was still necessary for the proposed system because we wanted to ensure that the CS did not violate the unlinkability requirement, i.e., the same reason as presented by Bauspieß *et al.* [7]. The goal of the binning method was to make it harder for the CS to recognize a pattern for similar subjects. Without a binning method, the CS would most likely recognize a clustering pattern after, e.g., 1000 executions of the proposed system. Since this was not desirable behavior, we wanted to hide which subjects were similar from the CS and use a binning method.

Even though each of the k clusters was viewed as k separate bins, we could not entirely hide the unlinkability from the CS. After many iterations, the CS would be able to know which subjects were grouped together, even though we viewed the clusters as separate bins. Because of this, we could initially give the CS the cluster distribution since it would recognize the clustering pattern in the end. Despite this behavior, the CS would not learn in which way the subjects are similar. The CS would learn that the same subjects are recursively grouped together because of their similarities. Still, it would not know if the subjects are from the same soft-biometrics, e.g., the same gender, ethnicity, or age group. This was because we used encryption

on the references, i.e., only the protected ciphertext of each reference was stored in the different bins.

With our proposed clustering technique and binning method, the clusters will no longer leak information because of the added encryption. By using plaintext references in the different clusters, there are potential problems that the clusters will leak some soft-biometrics away, as explained above in Section 7.1.1. For instance, one plaintext cluster could only contain female subjects, while another only contains male subjects. However, when we use encryption, we use randomized ciphertexts and hence receive randomized clusters from the given clustering technique. Once we use encryption, we know that some particular subjects are similar, but we cannot distinguish their similar characteristics. In addition, the relative sizes will also keep information private because of the use of randomized ciphertexts. A possible solution to smaller clusters, even for added encryption, would be to add the clusters and the bins with random data, i.e., random feature vectors. In this way, the clusters and the bins will be roughly the same size. With this argumentation, we would not observe any statistics on the population that would be reflected in the clusters and their sizes. Therefore, by using encryption and adding random data in the clusters, the problems discussed in Section 7.1.1 can be prevented. Adding more random data to some clusters could add extra cost to the identification process. Still, it can be argued that this will overall increase security, and the added extra cost will most likely not be significant.

By looking at security attacks, it can be argued that it would not decrease security if an attacker learns that, e.g., eight subjects in a bin are grouped based on the similarity of their feature vectors. This is because the attacker will not know in which way the subjects are similar; hence, the approach can still be argued that it is privacy-preserving. Here we have considered eight subjects in one bin because this is the average expected number of subjects, since $\frac{533}{64} \approx 8$. On the other hand, eight subjects might be too small of known similar references. With this point of view, combining, e.g., two clusters would be necessary to increase the number of similar references. An attacker would then know that 16 references are similar, which is not a problem since the similar soft-biometric characteristics will be kept secret. If we combine two bins into one, the execution times would double, but this will not be considered a problem because the increased computational workload would likely not be too large, as illustrated in Section 6.4. Based on this, we can make our proposed system and binning method more accurate with fewer references in each bin and secure simultaneously. Alternatively, we could scale the efficiency down with more references in each bin and achieve better preselection errors. A final point to note regarding this argumentation is that the second alternative, combining two clusters into one, will not be necessary if the clusters are relatively large. From Section 6.4.3, the largest cluster size of FERET was 17, while FRGCv2 had 15 subjects. This

means that if the smallest clusters are padded with random data, as discussed above, all clusters will have the same size as the largest cluster, and there is no need to combine two clusters to increase the number of similar subjects.

With the use of clustering techniques, it can be argued that any clustering based on similarity will most likely reveal some privacy-sensitive information in the form of the soft-biometric characteristics of the references. Therefore, we have presented a generic solution that can resolve these problems.

7.2 Experimental Evaluation Argumentation

This Section discusses some of the results from the experimental evaluation in Section 6.4, mainly the biometric performance, the comparison scores, the DET curve, and the efficiency of the proposed system.

7.2.1 Biometric Performance

To answer RQ3, we must discuss and look at the results presented in Section 6.4. For the proposed system, a candidate list consisting of enrollment references with the same PEKS keyword would be obtained, i.e., having the same cluster as an unidentified probe reference. This candidate list would be a bin of one of the $k = 64$ clusters, illustrated in Figure 6.1 for the FRGCv2 keyword vector distribution. After padding the smallest clusters with random feature vectors, the penetration rate, i.e., the average percentage of retrieved references relative to the size of the complete reference database, using the FRGCv2 database, would be 2.8%. This is because a maximum of 15 subjects would be similar to the probe and hence would be compared using FHE to decide if there was a mated reference. For the FERET database, the penetration rate would be 3.2% because there would be a maximum of 17 subjects in one cluster. Both of these penetration rates would be significantly reduced compared to the baseline system. For the baseline system, the penetration rate would be 100% as the preselected subset is equal to the number of subjects in the database because there was no use of preselection, i.e., no additional approach to reducing the biometric search space. By utilizing larger values of the number of clusters, i.e., $k = 128$ or $k = 256$, the number of subjects in each cluster would be significantly reduced, and hence also the penetration rate. A larger value than $k = 256$ would not be useful since this corresponds to the fact that some clusters only consist of one subject. When using a clustering technique to group subjects based on their similarities, having at least two subjects in each cluster is necessary to achieve a workload reduction through preselection. This is to prevent the distribution of subjects to clusters not only based on a random procedure but also on some similar characteristics between the objects in the clusters.

Comparing the biometric performance of the proposed system with the baseline system, the biometric performance is not significantly affected, as illustrated by the DET curve in Figure 6.4. The proposed system was not introduced to decrease the prediction errors but, on the contrary, to reduce the penetration rate and the workload effort, thereby improving the efficiency of the biometric identification process. As described above, using the stable hashing approach in the preselection approach significantly reduces the penetration rate. In addition, the combination of stable hash codes, i.e., cluster centers, with PEKS removed critical vulnerabilities. These vulnerabilities include the security issues and deterministic approach of the original stable hashing approach [8] and the bias and estimation problems in the PEKS-based identification approach [7]. In addition, the performance of the proposed system was still privacy-preserving because of the security impact of using PEKS and FHE.

Since the proposed system for this thesis used the same stable hash generation as the original work in [8], and both used the FERET database, we can compare the preselection error rates. Compared to the original work in [8], the preselection error rate of the proposed system was significantly higher. From Table V in [8], the FERET database's preselection error rate was 0%. This was from utilizing the k-means clustering technique and $k = 64$ clusters and $P = 1$ sub-spaces. In contrast, the preselection error rate of the proposed system was 2.15% for the FERET database, as presented in Section 6.4.3. There are several reasons for this difference. One reason is that the original work [8] used a different and larger FERET database than in the experiment of this thesis. Another reason is that the training process was different in [8] than in the proposed system. The original paper [8] used the same samples for the enrollment and the training samples, which could have affected the preselection error rate to be 0%. Because of these arguments, it was not possible to reproduce the same accuracies and preselection error rates as the original stable hash paper.

7.2.2 Comparison Scores

By studying the histogram plots of the comparison scores for the FERET and the FRGCv2 databases, Figure 6.2 and Figure 6.3, and looking at Table 6.6 of the comparison score statistics, we can see that the separation between mated and non-mated comparison scores is higher and better for the FERET database than the FRGCv2 database. This is because the minimum comparison score for non-mated pairs was higher than the maximum for mated pairs for the FERET database. Since the FRGCv2 database had a lower separation between comparison scores, some mated pairs obtained a significantly high comparison score, and some non-mated pairs obtained a significantly low one. One reason why a non-mated pair, i.e., samples from two separate subjects, obtained a significantly low comparison score could be that the two samples are from two subjects with relatively similar biometric

characteristics and appearance. For instance, the subjects are siblings with similar face structures, hair, and eye colors. On the other hand, one reason why a mated pair, i.e., two samples of the same subjects, received a poor and high comparison score could be that there was a poor quality of the samples. This could make it harder to measure the Euclidean distance of the sample vectors. With the FRGCv2 database, different samples were taken with different lighting conditions, which might impact the quality and affect the computation of the comparison scores.

7.2.3 DET Curve

In Section 6.4.10, Figure 6.4 only showed the DET curve for the FRGCv2 database, while the DET curve for the FERET database was empty. This was due to the perfect separation between mated and non-mated comparison scores for the FERET database, as presented above. When there is no overlap between mated and non-mated comparison scores, the DET plot will be empty. If we choose a threshold value below 0.87 in Figure 6.2, we can see that the FNIR will increase because some mated comparison scores will be above the threshold and classified as non-mated. Nevertheless, the FPIR will never increase. It will always be zero as long as we choose a comparison score threshold below the minimum value of non-mated comparison scores. Because of this separation of the FERET database, the accuracy of the biometric identification will be good.

On the other hand, since there was an overlap between mated and non-mated comparison scores for the FRGCv2 database, a DET curve could be illustrated. The DET curve for the FRGCv2 database was also remarkably good because of the low false positive scores, as indicated with FPIR around 0.001% and lower.

7.2.4 Efficiency and Workload Reduction

From Table 6.5, it took $10,006.24$ms to identify a probe reference using this proposed preselection approach for biometric identification. Out of the different functionalities, the FHE comparisons were the most time-consuming process, comprising $\frac{9996.00}{10,006.24} = 99.90\%$ of the entire execution time of the proposed biometric identification process. This was again due to the computational cost associated with FHE and the number of rotation operations that were used, as described in Section 2.2.3 and Section 2.2.4. On the other hand, the preselection approach was the least time-consuming task of the entire execution time. This comprised the time for producing the stable hash generation of the probe, creating the searchable ciphertext of the probe, and running the reverse PEKS search, all in the time of $0.28 + 0.27 + 7.69 = 8.24$ms. The preselection approach thereby only took $\frac{8.24}{10,006.24} = 0.082\%$ time of the entire execution time of the proposed system for the biometric identification process.

By comparing the execution time of the proposed system with the execution time of the baseline system, we have reduced the workload down to $\frac{10{,}006.24}{354{,}845.00} = 2.8\%$ of the baseline system. This is a good reduction of execution time and hence shows remarkable efficiency.

Since the proposed system used similar building blocks like PEKS and FHE comparison as the original work in [7], with PEKS with soft-biometric preselection, we could compare the execution times. Compared to the original work [7], the identification time of the proposed system was lower. From Table II in [7], the identification time was 19.52s, including the preselection approach. This was from using float templates and a dimensionality size of 512, i.e., identical to our experiment. In contrast, the identification time of the proposed system was 10.006s, as presented in Table 6.5. There are multiple reasons for this improvement. Firstly, the proposed system used the updated library OpenFHE instead of the PALISADE library used in [7]. OpenFHE is the successor of PALISADE and was released at the end of 2022 [57]. Secondly, our experiment has used a lower value for $scaleFactorBits$, thereby achieving faster execution times. The original work in [7] used $scaleFactorBits = 50$, while our experiment used $scaleFactorBits = 30$, as presented in Section 6.4.1. The $scaleFacorBits$ parameter of the proposed system was chosen to improve the efficiency but not affect the accuracy. In addition, the proposed system has reduced the number of subjects in each candidate list. The experiment in [7] has 88 subjects in each bin, while the proposed system only has 15 or 17 subjects, depending on the database. This will thereby speed up the execution time. Lastly, as specified in Section 6.4, our experiment was run on a new M2 CPU with a clock frequency of 3.50 GHz. In contrast, the experiment in [7] was run on an Ubuntu 20.04 operating system with an Intel i7-10750H CPU using a clock frequency of 2.60 GHz. The use of an M2 CPU could therefore have improved the execution times [58]. These arguments are also reflected in the case that our experiment was reduced down to 2.8% of the baseline system instead of the 8.4% from [7].

Compared to the original work in [7], there is also a distance regarding the number of trapdoors needed for the identification process. The original work [7] needed 155 trapdoors, while our experiment only needed 64. This indicates that the setup time for computing the necessary trapdoors is reduced for our experiment, in addition to reducing the execution time for the reverse PEKS search.

7.3 Ethics

Ethical concerns have been elaborated upon in the pre-project [9], and they will be briefly reiterated. Since the GDPR [5] characterizes biometric data as sensitive, as described in Section 2.1.1, some ethical considerations have been taken care of regarding the experimental evaluation of this thesis. This was to ensure that

the subjects received sufficient protection to which they were entitled. Firstly, the biometric data of the database subject for FERET and FRGCv2 has only been stored locally and not in any cloud environments. Secondly, the biometric data have only been used for the implemented experiment and not used in other cases. Thirdly, our experiment has used subject samples where the subjects have signed consent for using their facial images and biometric features for testing purposes. This was the case for both FERET and FRGCv2 databases. Lastly, the subject samples of the two databases were taken in a controlled environment.

Chapter 8
Conclusion

Constructing a secure and efficient biometric identification system is a challenging task. On the one hand, the security must be maintained by protecting the sensitive biometric information. However, on the other hand, the efficiency must be maintained such that the identification search does not become infeasible. Throughout this thesis, a proposed system has been constructed that respects both considerations. By combining stable hash codes, i.e., cluster centers, with the two approaches, Public-Key Encryption with Keyword Search and Fully Homomorphic Encryption, we have ensured that efficiency, security, and accuracy are provided. Thereby accomplishing the initial goal of this thesis. Since the proposed system uses a generic framework to enable privacy-preserving, the proposed system is also applicable to other indexing approaches.

The proposed biometric identification system was tested on two databases, the FERET database with 529 subjects and 1413 samples and the FRGCv2 database with 533 subjects and 3165 samples. Using the stable hash generation and grouping subjects in different clusters through the k-means clustering technique, a preselection method was introduced to ensure the efficiency. For this preselection approach, an accuracy and efficiency evaluation were tested for both databases. The FERET database received an accuracy of 97.85%, while the FRGCv2 database obtained 92.14%. For this evaluation, 64 clusters were used. While the FERET database had a maximum of 17 subjects in one cluster, the FRGCv2 had 15 subjects. Regarding the efficiency evaluation, it took 0.28ms for one probe sample to perform the stable hash generation, i.e., computing the corresponding cluster index. On the other hand, it took 10,006.24ms to complete the entire identification search for one probe reference. Compared to the workload of the initial baseline system, the identification time was reduced down to 2.8%.

Regarding the security protection, several considerations were elaborated. Firstly, only the protected ciphertexts from the Fully Homomorphic Encryption approach were stored in the clusters. If an attacker received the knowledge of which subjects

were grouped in which clusters, the attacker would only know which references were grouped together but would not know on which basis. This means the attacker would not know if the subjects were grouped based on the same gender, ethnicity, or age group. Secondly, a solution to add random data to the smallest clusters was also presented such that all clusters were of the same size and thereby not revealing any information only based on their cluster sizes. Lastly, it has been shown that long-term protection can be achieved through Public-Key Encryption with Keyword Search and Fully Homomorphic Encryption. This is due to the advances in lattice-based cryptography, and using the security guarantees of the Ring-Learning with Errors problem.

Lastly, the biometric performance of the proposed system was evaluated by using comparison scores and detection error trade-off curves. By evaluating comparison scores for mated and non-mated pairs, the FERET database received a better separation than the FRGCv2 database. Due to this remarkable separation of mated and non-mated comparison scores, the detection error trade-off curve was empty for the FERET database. On the other, for the FRGCv2 database, the connection between false positives and false negatives could be illustrated through the detection error trade-off curve. The FRGCv2 database showed low values of false positives around 0.001%, while the false negatives varied from 1% to 0.001%.

8.1 Further Work

For further work, additional research tasks can be considered for the proposed system. Firstly, due to the high cost of FHE computations and comparisons, it can be interesting to look into other approaches for template protection than FHE. This is to reduce the comparison cost but still ensure that the same level of security is ensured. Secondly, instead of testing on two separate databases, a larger database can be tested with the proposed system to see the impact on the accuracy, efficiency, and comparison scores.

References

[1] A. K. Jain, A. Ross, and S. Prabhakar, «An introduction to biometric recognition», *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 4–20, 2004.

[2] R. Kessler, O. Henniger, and C. Busch, «Fingerprints, forever young?», in *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, pp. 8647–8654.

[3] Unique Identification Authority of India, *Aadhaar Dashboard*. [Online]. Available: https://www.uidai.gov.in/aadhaar_dashboard/ (last visited: May 10, 2023).

[4] P. Drozdowski, C. Rathgeb, and C. Busch, «Computational workload in biometric identification systems: An overview», *IET Biometrics*, vol. 8, no. 6, pp. 351–368, 2019.

[5] European Parliament, *EU Regulation 2016/679 of the European Parliament and of the Council (General Data Protection Regulation)*, 2016.

[6] ISO/IEC JTC1 SC27 Security Techniques, *ISO/IEC 24745:2022. Information Technology - Security Techniques - Biometric Information Protection*, International Organization for Standardization, 2022.

[7] P. Bauspieß, J. Kolberg, P. Drozdowski, C. Rathgeb, and C. Busch, «Privacy-preserving preselection for protected biometric identification using public-key encryption with keyword search», *IEEE Transactions on Industrial Informatics*, 2022.

[8] D. Osorio-Roig, C. Rathgeb, P. Drozdowski, and C. Busch, «Stable hash generation for efficient privacy-preserving face identification», *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 4, no. 3, pp. 333–348, 2022.

[9] C. Fougner, «Secure and efficient preselection for biometric identification», Department of Information Security, Communication Technology, NTNU – Norwegian University of Science, and Technology, Project report in TTM4502, Nov. 2022.

[10] ISO/IEC JTC1 SC37 Biometrics, *ISO/IEC 2382-37:2017 Information Technology - Vocabulary - Part 37: Biometrics*, International Organization for Standardization, 2017.

[11] J. Deng, J. Guo, and S. Zafeiriou, «ArcFace: Additive angular margin loss for deep face recognition», in *Conferance on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019.

[12] V. N. Boddeti, «Secure face matching using fully homomorphic encryption», in *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, 2018, pp. 1–10.

[13] ISO/IEC JTC1 SC37 Biometrics, *ISO/IEC 19795-1:2021. Information Technology – Biometric Performance Testing and Reporting – Part 1: Principles and Framework*, International Organization for Standardization, Jun. 2021.

[14] S. Amerifar, A. T. Targhi, and M. M. Dehshibi, «Iris the picture of health: Towards medical diagnosis of diseases based on iris pattern», in *2015 Tenth International Conference on Digital Information Management (ICDIM)*, IEEE, 2015, pp. 120–123.

[15] R. L. Rivest, L. Adleman, and M. L. Dertouzos, «On data banks and privacy homomorphisms», *Foundations of Secure Computation*, vol. 4, no. 11, pp. 169–180, 1978.

[16] M. Yasuda, T. Shimoyama, J. Kogure, K. Yokoyama, and T. Koshiba, «Packed homomorphic encryption based on ideal lattices and its application to biometrics», in *International Conference on Availability, Reliability, and Security*, Springer, 2013, pp. 55–74.

[17] J. Brickell and V. Shmatikov, «Privacy-preserving graph algorithms in the semi-honest model», *International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT) 2005*, vol. 3788, pp. 236–252, 2005.

[18] C. Gentry, «Fully homomorphic encryption using ideal lattices», in *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, 2009, pp. 169–178.

[19] P. Drozdowski, N. Buchmann, C. Rathgeb, M. Margraf, and C. Busch, «On the application of homomorphic encryption to face identification», in *2019 International Conference of the Biometrics Special Interest Group (BIOSIG)*, 2019, pp. 1–5.

[20] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, «A survey on homomorphic encryption schemes: Theory and implementation», *ACM Computing Surveys*, vol. 51, no. 4, Jul. 2018.

[21] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, «(Leveled) fully homomorphic encryption without bootstrapping», *ACM Transactions on Computation Theory*, vol. 6, no. 3, Jul. 2014.

[22] O. Regev, «On lattices, learning with errors, random linear codes, and cryptography», *Journal of the ACM*, vol. 56, no. 6, Sep. 2009.

[23] J. Fan and F. Vercauteren, *Somewhat practical fully homomorphic encryption*, Cryptology ePrint Archive, Paper 2012/144, 2012.

[24] Z. Brakerski, «Fully homomorphic encryption without modulus switching from classical GapSVP», *Annual International Cryptology Conference (CRYPTO) 2012*, pp. 868–886, 2012.

[25] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, «TFHE: fast fully homomorphic encryption over the torus», *Journal of Cryptology*, vol. 33, no. 1, pp. 34–91, 2019.

[26] J. H. Cheon, A. Kim, M. Kim, and Y. Song, «Homomorphic encryption for arithmetic of approximate numbers», in *International Conferance on the Theory and Application of Cryptology and Information Security*, Springer, 2016, pp. 409–437.

[27] J. J. Engelsma, A. K. Jain, and V. N. Boddeti, «HERS: homomorphically encrypted representation search», *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 4, no. 3, pp. 349–360, 2022.

[28] Z. Brakerski, C. Gentry, and S. Halevi, «Packed ciphertexts in LWE-based homomorphic encryption», *Public-Key Cryptography – PKC 2013*, pp. 1–13, 2013.

[29] W. Wu, J. Liu, H. Wang, J. Hao, and M. Xian, «Secure and efficient outsourced k-means clustering using fully homomorphic encryption with ciphertext packing technique», *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 10, pp. 3424–3437, 2021.

[30] P. Bauspieß, J. Olafsson, J. Kolberg, P. Drozdowski, C. Rathgeb, and C. Busch, «Improved homomorphically encrypted biometric identification using coefficient packing», in *2022 International Workshop on Biometrics and Forensics (IWBF)*, 2022, pp. 1–6.

[31] C. Rathgeb and A. Uhl, «A survey on biometric cryptosystems and cancelable biometrics», *EURASIP Journal on Information Security*, vol. 3, Sep. 2011.

[32] H. Jégou, M. Douze, and C. Schmid, «Product quantization for nearest neighbor search», *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 117–128, 2011.

[33] J. MacQueen, «Some methods for classification and analysis of multivariate observations», *Proceedings of the Fifth Berkely Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 281–297, 1967.

[34] P.-N. Tan, M. Steinbach, and V. Kumar, «Data: Measures of similarity and dissimilarity», in *Introduction to Data Mining*, Pearson, 2014, pp. 64–83.

[35] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, «Public key encryption with keyword search», in *Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT) 2004*, Springer, 2004, pp. 506–522.

[36] R. Behnia, A. A. Yavuz, and M. O. Ozmen, «High-speed high-security public key encryption with keyword search», in *IFIP Annual Conference on Data and Applications Security and Privacy*, Springer, 2017, pp. 365–385.

[37] D. Boneh and M. Franklin, «Identity-based encryption from the Weil pairing», *Annual International Cryptology Conference (CRYPTO) 2001*, vol. 2139, pp. 213–229, 2001.

[38] L. Ducas, V. Lyubashevsky, and T. Prest, «Efficient identity-based encryption over NTRU lattices», *International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT) 2014*, vol. 8874, pp. 22–41, 2014.

[39] V. Lyubashevsky, C. Peikert, and O. Regev, «On ideal lattices and learning with errors over rings», in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, 2010, pp. 1–23.

[40] A. Dantcheva, P. Elia, and A. Ross, «What else does your biometric data reveal? A survey on soft biometrics», *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 3, pp. 441–467, 2015.

[41] P. Drozdowski, F. Stockhardt, C. Rathgeb, D. Osorio-Roig, and C. Busch, «Feature fusion methods for indexing and retrieval of biometric data: Application to face recognition with privacy protection», *IEEE Access*, vol. 9, pp. 139 361–139 378, 2021.

[42] P. Drozdowski, C. Rathgeb, B.-A. Mokroß, and C. Busch, «Multi-biometric identification with cascading database filtering», *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 2, no. 3, pp. 210–222, 2020.

[43] T. Murakami, R. Fujita, T. Ohki, Y. Kaga, M. Fujio, and K. Takahashi, «Cancelable permutation-based indexing for secure and efficient biometric identification», *IEEE Access*, vol. 7, pp. 45 563–45 582, 2019.

[44] E. Chavez, K. Figueroa, and G. Navarro, «Effective proximity retrieval by ordering permutations», *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 9, pp. 1647–1658, 2008.

[45] X. Dong, S. Kim, Z. Jin, J. Y. Hwang, S. Cho, and A. B. J. Teoh, «Open-set face identification with index-of-max hashing by learning», *Pattern Recognition*, vol. 103, p. 107 277, 2020.

[46] Z. Jin, J. Y. Hwang, Y.-L. Lai, S. Kim, and A. B. J. Teoh, «Ranking-based locality sensitive hashing-enabled cancelable biometrics: Index-of-max hashing», *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 2, pp. 393–407, 2018.

[47] M. S. Charikar, «Similarity estimation techniques from rounding algorithms», in *Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing*, ser. STOC '02, Montreal, Quebec, Canada: Association for Computing Machinery, 2002, pp. 380–388.

[48] R. Sobti and G. Geetha, «Cryptographic hash functions: A review», *International Journal of Computer Science Issues*, vol. 9, no. 2, pp. 461–479, 2012.

[49] C. E. Shannon, «A mathematical theory of cryptography», *Bell System Technical Memo MM 45-110-02*, 1945.

[50] M. Bellare, A. Boldyreva, and A. O'Neill, «Deterministic and efficiently searchable encryption», *Annual International Cryptology Conference (CRYPTO) 2007*, vol. 4622, pp. 535–552, 2007.

[51] B. Friedman and H. Nissenbaum, «Bias in computer systems», *ACM Transactions on Information Systems*, vol. 14, no. 3, pp. 330–347, Jul. 1996.

[52] P. Terhörst, J. N. Kolf, M. Huber, F. Kirchbuchner, N. Damer, A. M. Moreno, J. Fierrez, and A. Kuijper, «A comprehensive study on face recognition biases beyond demographics», *IEEE Transactions on Technology and Society*, vol. 3, no. 1, pp. 16–30, 2021.

[53] C. Rathgeb, P. Drozdowski, D. C. Frings, N. Damer, and C. Busch, «Demographic fairness in biometric systems: What do the experts say?», *IEEE Technology and Society Magazine*, vol. 41, no. 4, pp. 71–82, 2022.

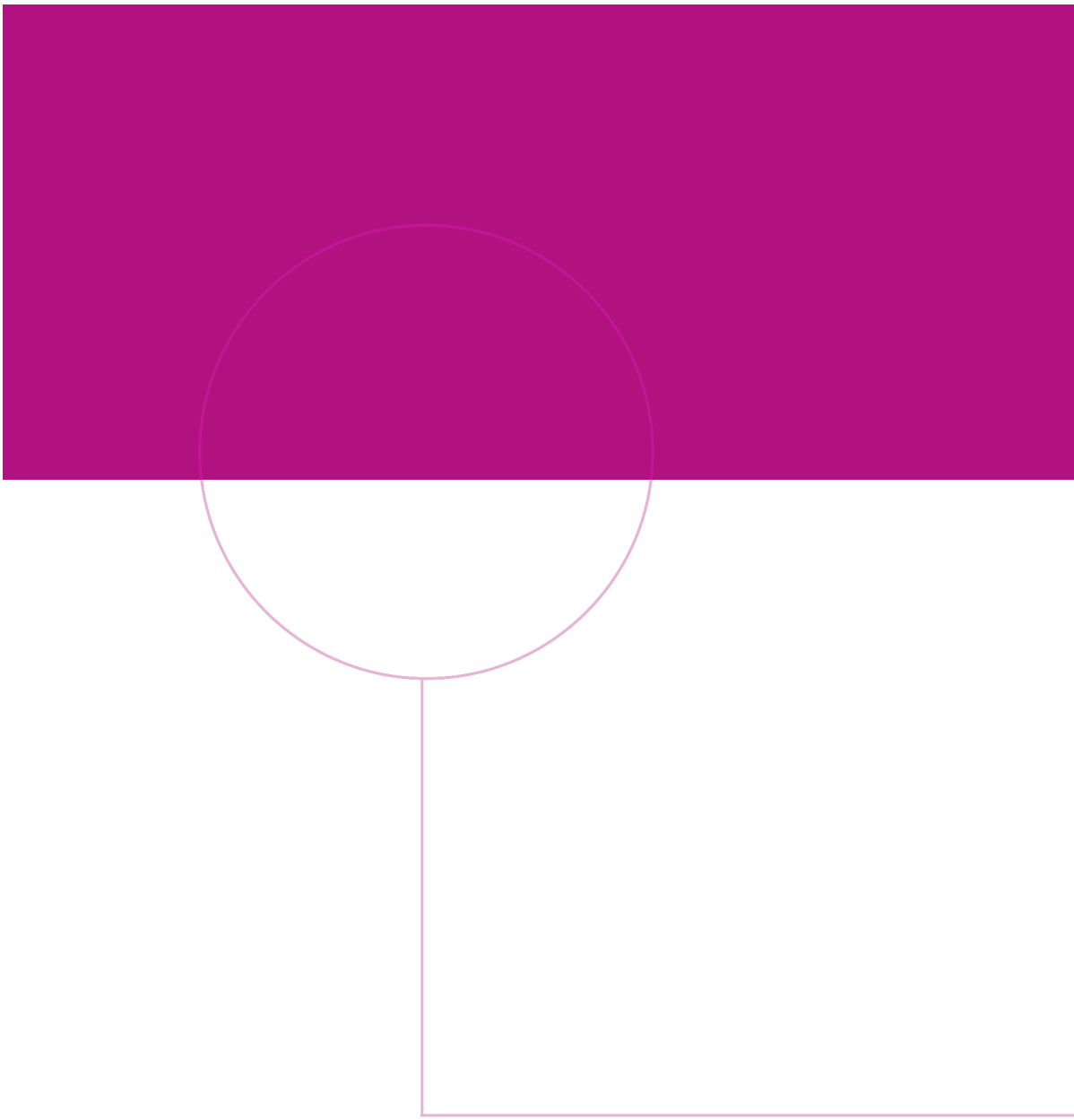
[54] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, «The FERET evaluation methodology for face-recognition algorithms», *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.

[55] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, «Overview of the face recognition grand challenge», in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005, 947–954 vol. 1.

[56] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, «Scikit-learn: Machine learning in Python», *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[57] A. A. Badawi, J. Bates, F. Bergamaschi, D. B. Cousins, S. Erabelli, N. Genise, S. Halevi, H. Hunt, A. Kim, Y. Lee, Z. Liu, D. Micciancio, I. Quah, Y. Polyakov, S. R.V., K. Rohloff, J. Saylor, D. Suponitsky, M. Triplett, V. Vaikuntanathan, and V. Zucca, *OpenFHE: open-source fully homomorphic encryption library*, Cryptology ePrint Archive, Paper 2022/915, 2022.

[58] *Intel Core i7 10750H vs Apple M2: performance comparison.* [Online]. Available: https://nanoreview.net/en/cpu-compare/intel-core-i7-10750h-vs-apple-m2 (last visited: Jun. 11, 2023).



NTNU

Norwegian University of
Science and Technology