h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACULTY OF COMPUTER SCIENCE

Hochschule Darmstadt, University of Applied Science
- Biometrics and Internet-Security Research Group-

BIOMETRIC INFORMATION PROTECTION IN I-VECTOR FEATURE SPACE

Abschlussarbeit zur Erlangung des akademischen Grades
Master of Science (M.Sc.)

vorgelegt von

SERGEY ISADSKIY

Referent:	Prof. Christoph Busch
Korreferent:	Andreas Nautsch
Ausgabedatum	01.04.2017
Abgabedatum	30.11.2017

ERKLÄRUNG

Ich versichere hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt habe.
Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht.
Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen.
Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner anderen Prüfungsbehörde eingereicht worden.

Darmstadt, 30. November 2017

Sergey Isadskiy

ABSTRACT

Nowadays, biometrics is revolutionizing the on-line payment system, creating an extra layer of security as part of a two-factor authentication process. Especially, voice biometric authentication is attractive to mobile banking and payment, since it provides consistency across multiple channels, works in real-time and is widely accepted by end-users. However, biometric data is highly sensitive. Hence, verification systems must protect biometric templates in order to prevent any disclosure of sensitive information about the user. In this thesis, a system architecture for protecting of biometric templates in terms of voice recognition is proposed. Using properties of homomorphic encryption, the system provides equal biometric performance as unprotected systems, thereby also fulfilling the requirements of the ISO/IEC 24745 standard on biometric information protection. Additionally, data of system vendors, such as expensively trained models, is handled in the same secure manner, assuring the needs of biometric system users and providers.

ZUSAMMENFASSUNG

Heutzutage werden Onlinebezahlsysteme durch die Verwendung von Biometrie revolutioniert, welche als zusätzliche Sicherheitsebene einer Zwei-Faktor-Authentifizierung dient. Vor allem die Verwendung der Stimme als biometrisches Merkmal ist für mobile Banking- und Bezahlunganwendungen attraktiv, da sie über verschiedenste Kanäle verwendet werden kann, echtzeitfähig ist und auf hohe Akzeptanz seitens der Anwender stößt. Da biometrische Daten hoch sensitive Informationen über den Anwender enthalten, muss ein biometrisches System diese Daten vor jedlichem unberechtigtem Zugriff schützen. In dieser Thesis wird eine Architektur zum Schutz biometrischer Templates im Rahmen der Sprechererkennung vorgestellt. Durch die Verwendung von Homomorphic-Encryption, bietet das System die gleiche Performanz wie ungeschützte Systeme und erfüllt dabei die Vorgaben des ISO/IEC 24745 Standard zum Schutz biometrischer Informationen. Auf die gleiche Weise werden die Daten der Anbieter Systeme, wie deren trainierte Modelle, verarbeitet und somit sichergestellt, dass sämtliche Anforderungen an das biometrische System sowohl der Anwender, als auch der Anbieter erfüllt werden.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

Secure authentication systems are crucial e.g., mobile banking and payment solutions. Therefore, biometrics provides security and user convenience at hand.

1.1 MOTIVATION

Data privacy and information security are two relevant terms used in context of data. Nowadays, with the concept of digital data, both are gaining in importance. Data security is *impersonal* and basically dealing with the security of the data (personal or non-personal) from accessibility of unauthorized users in general, data privacy is *personal* and used for protecting right respect to the processing of personal data. According to the definition in [1], personal data means any information relating to an identified or identifiable natural person, that makes it relevant for biometric systems in terms of collection, processing and use of data.

The term biometrics refers to "automated recognition of individuals based on their behavioural and biological characteristics" [2]. Biometric characteristics, such as fingerprints, face, voice, iris etc., represented by a biometric sample, are extracted then to biometric templates in order to be used for the biometric verification or identification process. For instance, in terms of security of electronic payments, biometric recognition is consumer-friendly, works in real-time and addresses the *Payment Services Directive (PSD2)* [3] requirement for more accurate validation, maximizing the privacy of personal data.

However, the above mentioned characteristics are irreplaceable in case of leaked information. This fact makes biometric systems vulnerable to any kind of attacks, that decrease the level of security of the biometric systems and disclose very sensitive information about the subjects.

In order to guarantee data privacy, biometric template protection schemes need to fulfil major requirements of biometric information protection [4]. However, the biometric templates cannot be encrypted with conventional encryption techniques, because the biometric samples captured from the same subject are not identical due to the number of variabilities influencing the sample. The most emerging topic in terms of security is the homomorphic encryption, where public

key cryptography is used. A similarity score is computed in the encrypted domain, so neither client nor server can learn any additional information about the other side template.

Placing focus on speaker recognition, the area of biometric template protection is an emerging topic. For the purpose of transferring data privacy methods from face and signature recognition to speaker recognition, recently proposed biometric information protection schemes are examined on i-vector speaker recognition approaches.
Thereby, the focus of this work is initially put on comparing of i-vectors by two-covariance comparator in the encrypted domain.

1.2 RESEARCH QUESTIONS

The purpose of this thesis is to transfer the concept of homomorphic encryption to the i-vectors model:

1. *Is it possible to transfer homomorphic encryption concepts to the i-vector/2Cov approach?*
   In order to answer this question, the biometric verification schemes based on homomorphic encryption, as well the homomorphic properties in general will be examined.

2. *How does homomorphic encryption influence the performance of 2Cov models and models using cosine-scoring technique regarding EER, C_{llr}^{min} and C_{llr}? Is the performance loss by using homomorphic encryption more than 5%?*
   The performance of the implemented systems will be measured evaluating various metrics.

3. *Is compliance of ISO/IEC IS 24745 in terms of irreversibility and unlinkability given?*
   The metrics of the ISO/IEC 24745 standard will be applied to check the security of biometric templates, according to the established requirements.

1.3 ORGANISATION OF WORK

This thesis is divided into four parts. The first part describes biometric systems in general and gives an overview on homomorphic encryption in biometrics. In the second part the schemes and techniques for the comparison of i-vectors in the encrypted domain are introduced. The third part evaluates the proposed methods in terms of various metrics and complexity. Finally, the last part provides a summary and discussion of the main results.

2

FUNDAMENTALS

2.1 BIOMETRIC SYSTEMS

The term biometrics is defined in [5] as "automated recognition of individuals based on their behavioural and biological characteristics". Thus, biometric system are implemented for that purpose, using for recognize either the biological (related to the structure of the body) or behavioural (related to the functions of the body) characteristics as an input.

2.1.1 *Components of the biometric System*

A biometric system consists of five subsystems. The functions of each subsystem are defined in [2] and shown in fig. 2.1.

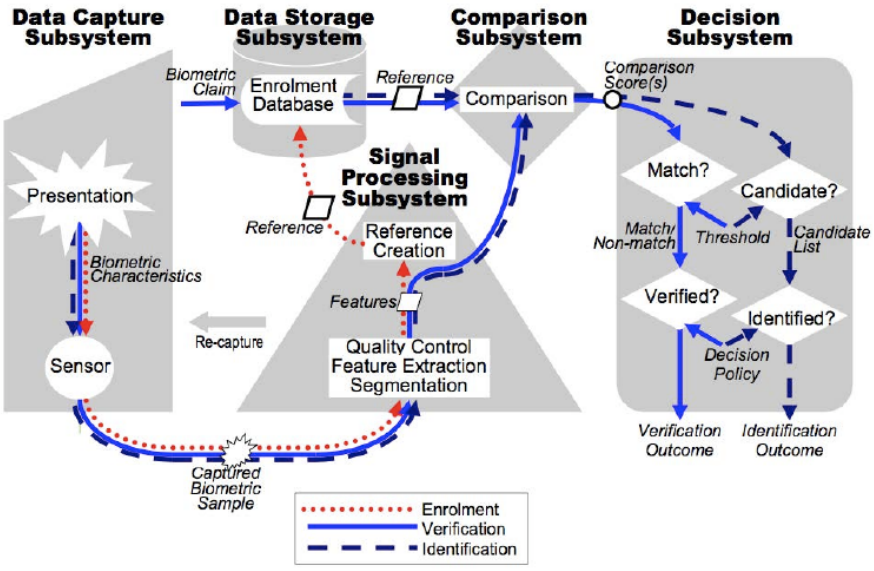

Figure 2.1: Components of the biometric system, see [2].

Data Capture Subsystem: A data capture subsystem collects captured biometric sample, converted from the biometric characteristics. A biometric capture device (e.g. sensor) collects characteristics as (digital) samples. The biometric sample is used as an input for a signal processing subsystem.

Signal Processing Subsystem: The signal processing subsystem can be divided into three modules:

1. *Segmentation*: The purpose of segmentation algorithm is to identify the parts of the biometric sample which contain regions of interest and the parts which contain background noise in order to erase the last ones.

2. *Quality Control*. The quality control processes predict the performance of the captured biometric samples. Samples of too low quality are expected to cause poor recognition performance, and are thus rejected, encouraging sample recaptures.

3. *Feature Extraction*: Biometric features are extracted from captured biometric samples. Depending on the systems mode, features are defined as reference or probe, respectively. References are stored in a data storage subsystem, and probes are used for comparison against loaded references.

Data Storage Subsystem: A data storage subsystem is responsible for storing of references of each enrolled user. For identification tasks the subsystem provides all references, associated to a biometric claim. In case of verification, a certain reference is retrieved for comparison.

Comparison Subsystem: A comparison subsystem receives a probe from a signal processing subsystem and a reference, stored in data storage subsystem. Comparison results are the numerical values, called comparison scores, indicating the degree of similarity or dissimilarity between reference and probe. In terms of this work, solely similarity scores are used for comparison.

Decision Subsystem: A decision subsystem uses a comparison score, given by a comparison subsystem, and, based on threshold and other decision policies, transform it into a binary yes or nor decision.

2.1.2 *Transactional Workflows in the biometric System*

The biometric system can operate in three different modes: enrolment, verification, identification, according to [2].

2.1.2.1 *Enrolment*

Enrolment is the process of creating a data record in the biometric enrolment database to serve for the comparison with a probe in verification and identification modes. Enrolment requires the data capture subsystem to collect captured biometric sample(s) and dependently extracted features. The usability of the generated biometric data is to ensure by testing verification or identification attempt.

2.1.2.2 *Verification*

Verification is the process of confirming a biometric claim in the comparison subsystem through (1:1) biometric comparison of a reference, stored in the biometric enrolment database of the data storage subsystem, with a probe of the claimed identity, generated in the signal processing subsystem. A comparison score, resulting from the comparison, is used in the decision subsystem to accept or reject the biometric claim, according to the pre-defined threshold.

2.1.2.3 *Identification*

Identification is the process of searching against a biometric enrolment database to find and return the biometric reference identifier(s) associated to an individual. For this purpose, a (1:n) comparison between a probe, provided by the data capture and signal processing subsystems, and all references from the database of the data storage subsystem is performed. The comparison scores for all references are calculated, generating a biometric candidate list. The scores of all reference identifiers (candidates) should exceed a pre-defined threshold. The decision subsystem uses the score to decide, if the individual is identified or not.

2.1.3 *Biometric Performance*

Performance of biometric systems can be measured in a variety of ways. One of those ways is analyzing with error estimation, since biometric systems encounters different types of failures.

2.1.3.1 *Biometric system error rates*

Failure-to-Capture(FTC): The proportion of failures of the biometric capture process to produce a captured biometric sample that is acceptable for use [5], calculated as:

$$FTC = \frac{N_{tca} + N_{nsq}}{N_{tot}}, \tag{2.1}$$

where N_{tca} is a number of terminated capture attempts, N_{nsq} is the number of images created with insufficient sample quality and N_{tot} is the total number of capture attempts. FTC occurs in data capture subsystem.

Failure-to-Extract(FTX): proportion of failures of the feature extraction process to generate a biometric template from the captured biometric sample, calculated as:

$$FTX = \frac{N_{ngt}}{N_{sub}}, \tag{2.2}$$

where N_{ngt} is the number of templates, that were failed to generated and N_{sub} is the total number of biometric samples being submitted from data capture subsystem. FTX occurs in signal processing subsystem.

Failure-to-Enroll(FTE): The proportion of failures of the enrolment process to create and store an enrolment data record for a biometric data subject [5], calculated as:

$$FTE = \frac{N_{nec}}{N}, \tag{2.3}$$

where N_{nec} is the number of subjects, that were failed to enrolled and N is the total number of subjects. FTE occurs in data storage subsystem.

Failure-to-Acquire(FTA): proportion of a specified set of probe acquisitions that failed to create a biometric probe[5], calculated as:

$$FTA = FTC + FTX * (1 - FTC). \tag{2.4}$$

2.1.3.2 *Algorithmic verification error rates*

False-Match-Rate(FMR): proportion of the completed biometric non-mated (impostor) comparison trials that result in a false match [5], calculated as:

$$FMR(t) = \int_t^1 \Phi(s|H_A)ds, \tag{2.5}$$

where t is a decision threshold, H_A defines the statement, where reference and probe come from different users, Φ is a probability density function, and s is a similarity score.

False Non-Match Rate(FNMR): proportion of the completed biometric mated (genuine) comparison trials that result in a false non-match [5], calculated as:

$$FNMR(t) = \int_t^1 \Phi(s|H_0)ds, \tag{2.6}$$

where H_0 defines the statement, where reference and probe come from the same user.
FMR100 is a special case of FNMR when FMR is 1%.

Equal-Error-Rate (EER): the point where $FNMR(t) = FMR(t)$.

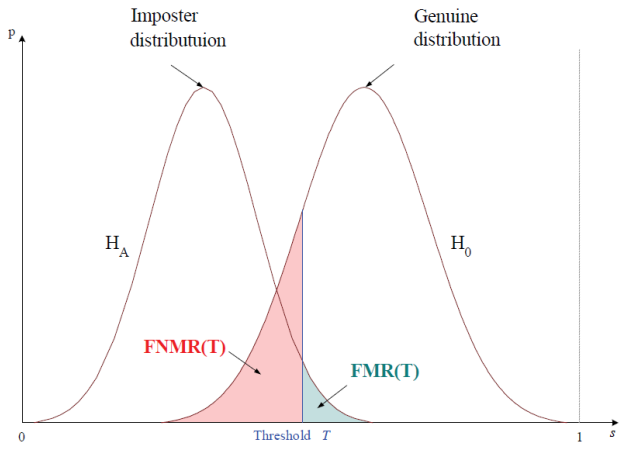

Figure 2.2: FMR and FNMR.

2.1.3.3 *Overall system performance*

False Rejection Rate (FRR): a proportion of verification transactions with truthful claims of identity that are incorrectly denied [2], calculated as:

$$FRR = FTA + FNMR * (1 - FTA). \tag{2.7}$$

False Rejection Rate (FAR): a proportion of verification transactions with wrongful claims of identity that are incorrectly confirmed [2], calculated as:

$$FAR = FMR * (1 - FTA). \tag{2.8}$$

2.1.3.4 *Comparison scores*

Likelihood Ratio (LR): ratio of the probability of the evidence in genuine or impostor [6], given as:

$$LR = \frac{p(E|H_0)}{p(E|H_A)}, \tag{2.9}$$

where E is an observation, H_0 is the probabilities of genuine hypothesis and H_A is the probabilities of impostor hypothesis.

Example [7]: Given an unknown speech sample X (crime evidence) and a speech sample Y (exemplar) from a subject (suspect), the observed evidence E can be defined based on certain feature parameters (e.g. difference of average voice frequency between X and Y). Thus, it is possible to calculate the probability distribution of this feature parameter for speech samples of the same subject (H_0) and for speech samples of different subjects (H_A). Knowing the hypothesises, the conditional probabilities $p(E|H_0)$ and $p(E|H_A)$ are calculated.
The calculation is based on Bayes' theorem:

$$\frac{p(H_0|E)}{p(H_A|E)} = \frac{p(E|H_0)}{p(E|H_A)} \cdot \frac{P(H_0)}{P(H_A)}. \tag{2.10}$$

Thus, the posterior probabilities $\frac{p(H_0|E)}{p(H_A|E)}$ can be only estimated if the prior probabilities of the hypotheses $P(H_0)$ and $P(H_A)$ are given.

Log Likelihood Ratio (LLR): logarithmic compressed LRs, typically using the natural or the base 10 log.

Detection Cost Function (DCF): application dependent metric, representing a linear combination of FFR and FAR, weighted by their costs [8]. DCF is calculated as:

$$DCF(\eta) = C_{FR} * \mathrm{FRR}(\eta) * \pi + C_{FA} * \mathrm{FAR}(\eta) * (1 - \pi), \tag{2.11}$$

where C_{FR} and C_{FA} are *application-dependent* cost parameters, representing the cost weights for FRR and FAR, respectively, π is a target prior probability, and η is the decision threshold.

DCFmin: DCF for the optimal threshold η_{opt}, obtained by:

$$\eta_{opt} = \arg\min_{\eta} \mathrm{DCF}(\eta). \tag{2.12}$$

Cost of LLR: an *application-independent* performance metric, proposed in [9], which is used to measure the goodness of LLR scores and calculated from genuine and imposter LLR scores by integrating over all cost functions DCF in eq. 2.11:

$$C_{llr} = \int_0^1 \mathrm{DCF} = \frac{1}{2}\left(\frac{1}{N_{H_0}} \sum_{i\ for\ H_0=true}^{N_{H_0}} \log_2\left(1 + \frac{1}{LR_i}\right)\right) + \frac{1}{2}\left(\left(\left(\frac{1}{N_{H_A}} \sum_{i\ for\ H_A=true}^{N_{H_A}} \log_2(1 + LR_j)\right)\right)\right) \tag{2.13}$$

where N_{H_0} and N_{H_A} are the number of mated vs. non-mated trials, LR_i and LR_j are the LRs derived from SU and DU comparisons, respectively. For the case the biometric system produces good quality LRs, all the mated trials should produce LRs greater than 1, and all the not-mated trials should produce LRs less than 1.

Minimum Cost of LLR: a discrimination loss. C_{llr} can be split into a discrimination loss C_{llr}^{min} and a calibration loss C_{llr}^{cal} after the system has been calibrated. For the case that the system is optimal calibrated, C_{llr}^{min} can be used to show the overall performance of the system.

2.2 PRIVACY IN BIOMETRICS

Data privacy and information security are two relevant terms used in context to biometric data. Data security is *impersonal* and basically dealing with the security of the data (personal or non-personal) from

accessibility of unauthorized users in general. Contrastively, data privacy is *personal* and used for protecting right respect to the processing of personal data. According to the definition in Regulation (EU) 2016/679 [10], personal data means any information relating to an identified or identifiable natural person, and since biometric characteristics are unique to individuals, the data privacy issue becomes relevant for biometric systems in terms of collection, processing and use of data.

2.2.1 *General Data Protection Regulation*

Biometric data is categorised by the General Data Protection Regulation (GDPR) [1], that becomes enforceable in 2018, replacing Regulation (EU) 2016/679 [10], even as "sensitive personal data", that follows to additional protections and restrictions.

The GDPR defines biometric data as *"personal data resulting from specific technical processing relating to the physical, physiological or behavioural characteristics of a natural person, which allow or confirm the unique identification of that natural person, such as facial images or dactyloscopic data2* ([1], Article 4(14)). The processing of biometric data shall be prohibited ([1], Article 9(1)), except for the cases, mentioned in [1], Article 9(2):

- if the data subject has given explicit consent for the processing of his or her biometric data ([1], Article 9(2)(a)),
- the processing is necessary for reasons of substantial public interest ([1], Article 9(2)(f)),
- the processing is requested the frame of court proceedings ([1], Article 9(2)(g)).

If biometric data is processed on large scale and processing is likely to result in a high risk to the rights and freedoms of data subjects ([1], Article 35(1)), data protection impact assessments (DPIAs) are necessary, where data controller needs to *"evaluate, in particular, the origin, nature, particularity and severity"* of the risk and inform supervisory authority, if this risk cannot be mitigate by appropriate technical measures. When implementing biometric technologies, these measures can prevent spoofing or alteration of the biometric data (privacy by design ([1], Article 25(1))). The data controller shall additional ensure that, by default, only personal data which is necessary for each specific purpose of the processing is processed ([1], Article 25(2)).

2.2.2 *Payment Services Directive*

PSD2 [3] is the second Payment Services Directive, which will come into force in 2018, designed in order to improve the existing European rules for electronic payments, thus providing a better protection of consumers and promoting an innovation in the payment space.

PSD2 requires Third Party Providers (TPPs) to apply *strong customer authentication* ([3], Article 4(30)) [1], that must be based on at least two or more independent elements:

1. *knowledge*: something only the user knows (a password, a PIN, etc.),
2. *possession*: something only the user holds (a card, a token, etc.),
3. *inherence*: something the user is (biometrics).

The independence of the elements means that if one of them is compromised, the reliability of the others is still guaranteed ([11], Chapter 2, Article 9(1)). In addition, they cannot be disclosed (Chapter 2, Article 6) or replicated ([11], Chapter 2, Article 7), and in terms of biometrics they must have low false positives ([11], Chapter 2, Article 8). All in all, the elements must be designed in such a way to protect the confidentiality of the authentication data ([3], Article 4(30)).
PSD2 requires the strong customer authentication, if e.g. a payer accesses its payment account online or initiates an electronic payment transaction [3], Article 97(1).

The banks will probably chose the knowledge element as a first factor of authentication, so inherence could be the second one, since solutions based on possession element are more vulnerable to the fraud attacks and can not always satisfy the requirements, described in ([11], Chapter 2, Article 7).

TPPs will be able to access payment accounts and submit payments only if it has been agreed by the customers. Thus, a consent is a connection point between GDRP and PSD2.

2.2.3 *Privacy demands biometric template protection*

Biometric characteristics are irreplaceable in case of leaked information. This fact makes biometric systems vulnerable to any kind of attacks, that decrease the level of security of the biometric systems and disclose very sensitive information about the users of biometric

1 PSD2 empowers the European Banking Authority (EBA) to develop regulatory technical standards (RTS) and guidelines, including RTS on SCA and secure communication [11], that must be released on different times.

system.

In order to guarantee data privacy, biometric template protection schemes need to fulfil major requirements of biometric information protection [4]:

1. *irreversibility*: a potential attacker should not be unable to reconstruct the original biometric sample from the biometric template, since the reconstructed biometric sample can be used for the unauthorized system access or identity thief.

2. *unlinkability*: the stored biometric references shall not be linkable across applications or databases. A potential attacker should be unable to retrieve biometric features or gain any secret information, combining one or more biometric templates (*leakage amplification*). In addition, if the attacker in possession of two protected templates, it should be difficult for him to identify if they belong to the same user or not (*cross-matching*).

3. *confidentiality*: biometric references shall be kept confidential in order to protect biometric references against access by an unauthorized user resulting in a privacy risk.

2.3 HOMOMORPHIC ENCRYPTION

The main motivation to use *homomorphic encryption (HE)* is that it allows computations to be performed on ciphertexts in encrypted domain, thus generating encrypted results which when decrypted, can match the result of the operations carried out on the plaintexts. In terms of biometrics, HE approaches can be applied to transform the score computation into the encrypted domain, keeping the private biometric data safe and secure.

The other notable feature of HE is the providing of *semantic security*, so that an attacker cannot guess with better probability than $1/2$ whether the given ciphertext stems from which plaintext. In case of RSA [12], the first proposed public-key approach with homomorphic property, semantic security can be achieved only by padding of a message with random bits. This step, however, destroys the homomorphism of RSA.

In search of the encryption scheme for this work, the choice is fallen on the Paillier encryption scheme, since the cryptosystem based on it, remains semantic secure, fulfilling the homomorphic properties, described below.

2.3.1 *Definition of Homomorphic Encryption*

In abstract algebra, homomorphism is a structure-preserving map between two algebraic structures of the same type, such as groups. A group G consists of a set of elements $\{a, b, c...\}$ and an operation $\circ$, such that any two elements of the group can form a third element: $a \circ b = c$, where $a, b, c \in G$. The set an operation must satisfy four conditions, the group axioms:

1. Closure: For all a,b in G the result of the operation $a \circ b$ is also in G.
2. Associativity: For all a, b and c in G, $(a \circ b) \circ c = a \circ (b \circ c)$.
3. Identity: There exists an element e in G, such that for every element a in G, the eq. $e \circ a = a \circ e = a$ holds.
4. Invertibility: For each a in G, there exists an element b in G, such that $a \circ b = b \circ a = e$, where e is the identity element.

Given two groups $(G, \diamond)$ and $(H, \circ)$, a group homomorphism from $(G, \diamond)$ to $(H, \circ)$ can be defined as a function $f : G \to H$ such that for all g and g' in G the following eq. holds:

$$f(g \diamond g') = f(g) \circ f(g') \tag{2.14}$$

Group homomorphism is demonstrated in fig. 2.3.

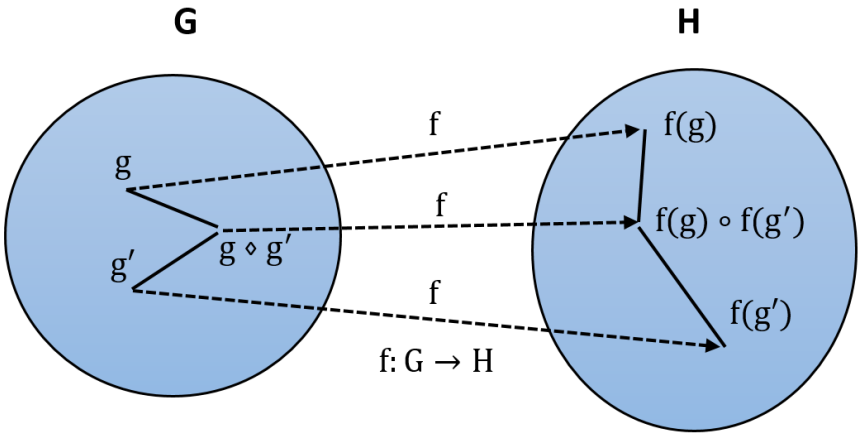

Figure 2.3: Group Homomorphism.

Let (P, C, K, enc, dec) be an encryption scheme, where P, C and K defines the plaintext, ciphertext and key space, respectively, *enc* and *dec* are the corresponding encryption and decryption algorithms. If the plaintexts build a group $(P, \diamond)$ and the cipertexts build a group $(C, \circ)$, the encryption algorithm *enc* undertakes the mapping between P and C:

$$enc_k(a) \circ enc_k(b) = enc_k(a \diamond b),\ homomorph\ \forall a, b \in P,\ \forall k \in K, \tag{2.15}$$

where k is a public key with $enc_k : P \rightarrow C$.

Given a cryptosystem C with plaintext m_n, ciphertext c_n and an encryption function enc, such that $enc(m_n) = c_n$ and some operation $\circ$:

C is *additively homomorphic* if:

$$\exists \circ : enc(m_1) \circ enc(m_2) = enc(m_1 + m_2) \tag{2.16}$$

C is *multiplicatively homomorphic* if:

$$\exists \circ : enc(m_1) \circ enc(m_2) = enc(m_1 \cdot m_2) \tag{2.17}$$

If a cryptosystem shows both additive and multiplicative homomorphism, it is called *fully homomorphic*. Otherwise, it supports only additive or multiplicative mechanism and is called *partly homomorphic*. Since fully homomorphic is quite limited regarding its computation capabilities and not practical in a real-life applications [13], the work will focus on *partially homomorphic* schemes, that are widely used in biometric approaches, as described below.

2.3.2 *Paillier Homomorphic Encryption Scheme*

A number of public-key asymmetric cryptosystems are classed under *particular homomorphic* schemes. Besides well established schemes based on a different problem, like RSA [12] or ElGamal [14], another important approach became significant over the years. In 1999, Pascal Paillier proposed a probabilistic asymmetric algorithm [15]. The algorithm is described in the next subsection.

2.3.2.1 *Key generation*

1. Choose two large prime numbers p and q randomly and independently of each other, so that:

$$gcd(pq, (p-1)(q-1)) = 1, \tag{2.18}$$

   where gcd represents the largest positive integer that divides pq and $(p-1)(q-1)$ without a remainder.

2. Estimate RSA modulus n and Carmichael's function λ:

$$\begin{aligned} n &= p \cdot q \\ \lambda &= \frac{(p-1)(q-1)}{gcd(p-1, q-1)} \end{aligned} \tag{2.19}$$

3. Select random generator g where $g \in \mathbb{Z}^*_{n^2}$

4. Calculate the modular multiplicative inverse to ensure that n divides the order of g :

$$\mu = (L(g^{\lambda} \bmod n^2))^{-1} \bmod n, \tag{2.20}$$

where function L is defined as $L(x) = \frac{x-1}{n}$.

The public key for encryption is $pk(n, g)$, and the secret key for decryption is $sk(\lambda, \mu)$.

2.3.2.2 *Encryption*

1. Let m be a message to encrypt where $m \in \mathbb{Z}_n$
2. Select random r where $r \in \mathbb{Z}_n^*$
3. Compute ciphertext:

$$c = g^m \cdot r^n \bmod n^2 \tag{2.21}$$

2.3.2.3 *Decryption*

1. Let c be a chipertext to decrypt where $c \in \mathbb{Z}_{n^2}^*$
2. Compute the plaintext:

$$m = L\left(c^{\lambda} \bmod n^2\right) \cdot \mu \bmod n \tag{2.22}$$

2.3.2.4 *Background*

Paillier cryptosystem is based on the fact that certain discrete logarithms can be estimated easily, e.g. by using binomial theorem:

$$\begin{aligned}(1+n)^x = \sum_{k=0}^{x} \binom{x}{k} n^k = \binom{x}{0} n^0 + \binom{x}{1} n^1 + \binom{x}{2} n^2 + \ldots \\ + \binom{x}{x-1} n^{x-1} + \binom{x}{x} n^x \\ = 1 + n \cdot x \left(\bmod n^2\right).\end{aligned} \tag{2.23}$$

If $y = (1+n)^x \bmod n^2$, then $x = \frac{y-1}{n} (\bmod n)$.

$L\left((1+n)^x \bmod n^2\right) = \mathbf{x} (\bmod n)$, where $L(u) = \frac{u-1}{n}$ for $x \in \mathbb{Z}_n$.

2.3.2.5 *Paillier properties*

According to [15], given two ciphertexts $enc(m_1) = g^{m_1} \cdot r_1^n (\bmod n)$ and $enc(m_2) = g^{m_2} \cdot r_2^n (\bmod n)$, homomorphic addition and multiplication can be defined:

- The product of two chipertexts will decrypt to the sum of their corresponding plaintexts:

$$dec\left(enc\left(m_1\right) \cdot enc\left(m_2\right) \ mod\, n^2\right) = m_1 + m_2 \, mod\, n. \tag{2.24}$$

- The product of a ciphertext with a plaintext raising g will decrypt to the sum of the corresponding plaintexts:

$$dec\left(enc\left(m_1\right) g^{m_2} \, mod\, n^2\right) = m_1 + m_2 \, mod\, n. \tag{2.25}$$

- An encrypted plaintext raised to the power of another plaintext (constant) will decrypt to the product of the two plaintexts:

$$dec\left(enc\left(m\right)^k \ mod\, n^2\right) = k \cdot m \, mod\, n. \tag{2.26}$$

2.3.2.6 *Security*

Ciphertext Indistinguishability (IND) is an important concept in terms of encryption, and describes an unability of an attacker to distinguish pairs of ciphertexts based on the message they encrypt. The Paillier encryption provides this property against *chosen-plaintext attacks (IND-CPA)*. In particular, when an attacker generates two messages, which are randomly encrypted by a challenger, guessing which encrypted messages stems from which chosen-plaintext is at least as difficult as solving the *decisional composite residuosity assumption (DCRA)*. DCRA describes that given a composite n and an integer z, it is hard to compute whether z is a n-residue modulo n^2 or not, i.e., whether there exists y such that:

$$z = y^n \, mod\, n^2. \tag{2.27}$$

Bellare et al. [16] demonstrated that no HE scheme can be secure against *adaptive chosen ciphertext attacks (IND-CCA2)* (algorithm, where an adversary can call an encryption or decryption oracle before and after receiving ciphertext) because of its malleable property. In case of Paillier cryptosystems, where only the public-key and an encryption of m_1 and m_2 are given, the malleability would mean, that it is possible for an adversary to compute a valid encryption of their sum $m_1 + m_2$. In [17], the schema is introduced, where the combined hashing of message m with random r prevents the adversary to change m in a meaningful way, so the scheme remains secure against *IND-CCA2*.

2.3.3 *Overview on biometric template protection*

The aim of biometric template protection schemes is to prevent the use of inverse biometrics, that allow the attacker to reconstruct a synthetic biometric sample from the information of the stored biometric templates, using it later for recognition attempts.
A distinction is usually made between approaches in *cancelable biometrics*, *cryptobiometrics* and *biometrics in the encrypted domain*.

2.3.3.1 *Cancelable biometrics*

"Cancelable biometrics consist of intentional, repeatable distortions of biometric signals based on transforms which provide a comparison of biometric templates in the transformed domain" [18].

The first group of cancelable biometrics approaches provides *non-reversible transformation* of the biometric data, using noninvertible funtions. The impostor fails to obtain the original feature set even in possession of transforms. Two drawbacks to be named here are challenging aspects with alignment during template comparison and reduce of information during the transformation, causing more complexity in terms of discriminability (similarity structure) and leading to a corresponding accuracy decrease. Ratha et al. [19] use the technique of non-reversible transformation to generate cancelable fingerprint templates.

In the second group of cancelable biometrics, *biometric salting*, a function, defined by a subject-specific token, is used to create a distorted biometric template. Because the transformation function is invertible, the token has to be kept secret and presented by user at authentication. The token introduces a new source of entropy in the biometric system and provides better low false accept rates [20], but at the same time the template becomes insecure, if an impostor compromises the token. An example of salting approach is presented in [21].

2.3.3.2 *Cryptobiometrics*

In case of cryptobiometrics public available *helper data* is generated. Helper data does not reveal any important information, but is used during authentication in order to obtain a cryptographic key, leading to a successful match.

In a *key binding* cryptobiometric cryptosystem, helper data is generated after the fusion of biometric data with a cryptographic key in the binding process. A key retrieval algorithm delivers a key at authentication. An error correcting code, used additionally in combination with helper data, proves the error tolerance in the case where reference and probe templates differ from each other. If the number of errors is in the tolerance range, a codeword with the similar number of error can be received and subsequent decoded to estimate the exact codeword and obtain the key. That improves the tolerance to intra-user variations of biometric samples, but can reduce the matching accuracy, if error correctness algorithm does not cover all possible variations of the template [20]. Fuzzy commitment scheme [22] is an example of key binding technique.

In a *key generation*, helper data is obtained only from the biometric template, and a cryptographic key is generated directly from a biometric sample and helper data at authentication. Chang et al. [23] and Vielhauer et al. [24] introduced quantization schemes. The intervals for each real-valued feature are calculated and stored as helper data. At verification, the characteristics of a biometric sample are mapped into previously defined intervals in order to obtain a key.
Dodis et al. [25] proposed secure sketch and fuzzy extractor technique. While secure sketch extracts random string from a biometric sample at enrolment and assists in reconstruction of the template, a primitive fuzzy extractor is used to generate a key from the biometric data at verification.
The main limitation in the key generation process gets quickly visible. The key generation suffers from high false rejection rate because of the variability of the biometric traits.

2.3.3.3 *Biometrics in the encrypted domain*

This scheme of biometric template protection provides encryption of the reference templates and comparison in the encrypted domain. In encrypted domain, it becomes possible to avoid such limitations of the above mentioned template protection schemes, as information loss in the case of cancelable biometrics and some challenging aspects in binding and generation processes in the cryptobiometrics approaches. For the purpose of this work the schemes based on *Garbled Circuits (GC)* [26] or *Oblivious Transfers* [27] are not be observed, and the focus is set on HE. HE allows computations to be performed on ciphertexts, thus generating encrypted results which when decrypted, matches the result of the operations carried out on the plaintext. Also no *Auxiliary Data*, such as a subject-specific token, is required.

In this work, *anonymous biometric authentication protocols* are emphasized on: solely the fact that the user is one of various trusted and allowed individual are of practical importance, his or her identity is not relevant for identification purpose. It also means, that the probe template is compared to *all* templates from the database, and access is succeed, if at least one of them matches. *Semi Homomorphic Encryption (SHE)* schemes, which only allow a limited number of operations on encrypted data, are successfully implemented in many biometric applications, based on *anonymous biometric authentication protocols* [28–35].

In [28], an algorithm based on HE and tested on a dataset of iris patterns is presented. Client C sends the private key pk, the encrypted probe and the encrypted inversed probe to the server S. S provides a secure computation of Hamming distances between C's probes and every reference from a biometric database DB. Comparison between

the calculated encrypted distance and a plaintext threshold cannot be measured as a summation or multiplication of two numbers, so S needs to first use bit extraction to get the encrypted bit representation of all the distances. S needs C's assistance in extracting the bits and decryption the numbers. In order to avoid C to get any information from encrypted distances, S adds random numbers to the distances. C decrypts the randomized distances, extracts the individual bit, sends it encrypted back to S. S is responsible for the threshold comparison, C assists S only in the secure multiplication, that is used to minimize the adversarial intention of C. Finally, a collision-resistant hash function HASH is used to prevent both parties to gain any significant information about each other. Output is a single bit for access or rejection.
This approach uses also so-called k-anonymous Quantization in order to reduce the scope of the similarity search from the entire DB to k candidates.

Adversarial behaviors can be generally classified into two types: *semi-honest (honest-but-curious)* and *malicious* [36]. Security is guaranteed in a *honest-but-curious* model, as the adversaries follow the protocol but try to learn additional information. In [28] it is S, that follows the protocol, but can try to find out private data of the biometric subject. In a *malicious* model the adversary can change private inputs or even terminate the protocol. As described above, the protocol is secure against malicious behavior of the adversary client C', as it can manipulate data in order to get access to the system.

Another variant of Paillier HE approach, tested for iris identification system, is proposed in [29]. It includes comparison server M beside the database DB and authentication server AS, as shown in fig. 2.4. AS receives an encrypted biometric template from client and requests randomly all k stored templates t_i from DB and computes k times $q \bigoplus t_i$, where $\bigoplus$ is encrypted xor operation. M takes over the task of determining the hamming distances between reference and probe templates (since this operation cannot be accomplished in the encrypted domain using partially HE techniques). The previous permutation of the data (string from *xor* operation) is required in order to reduce potential *honest-but-curious* behavior of M. Since M is in possession of private key, that could be used to eavesdrop the data being transmitted from the client, encrypted previously with the public key. Finally, M decrypts permuted strings and calculates Hamming distance, sending the result back to AS, that decides, using predefined threshold, whether access is granted or not.
Another distinctive feature of this approach is that it operates on chunks. *Paillier Chunkwise* outperforms a bit-wise HE, since it still fulfills the requirements of Paillier scheme, but being faster by sev-

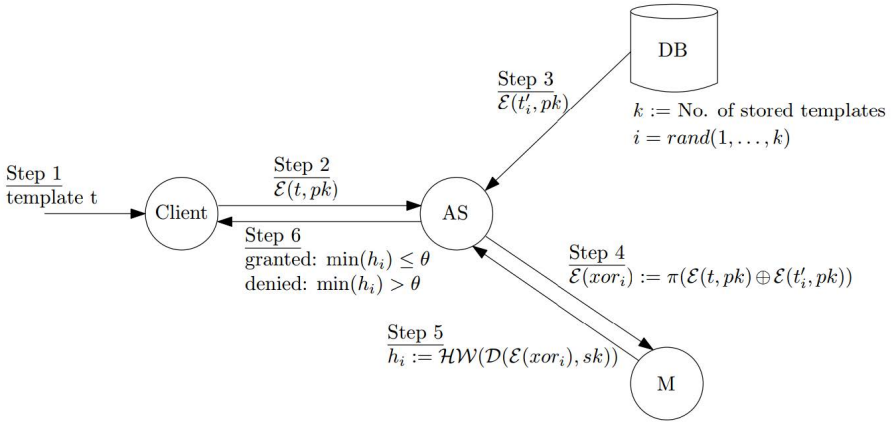

Figure 2.4: Architecture of iris identification system, cf.[29].

eral orders of magnitude compared to the bite-wise Paillier scheme.

For fingerprint recognition, a fingerprint verification system based on the FingerCode fixed-length representation of fingerprints and HE [30] should be mentioned. At first, the feature vector, extracted and encrypted by the client *C*, is sent to the server *S*. Then the distances (in that case - the square of the Euclidean distance) between the target vector and the vectors located in the database are calculated by server *S* in the ciphertext domain using HE. The encrypted distances remains unknown to *S*, still it aims to avoid on the server side to get any information about the requested biometry or exploit the resulting comparison scores. In order to select the matching identity, *S* has to interact with *C* using some internal sub-protocols. The output, that can have more than one identity, is only known by *C*. The drawback of that approach is that the templates in *DB* are still stored unencrypted.

In [31] the proposed feature size reduction methods have a very limited impact on the accuracy of the biometric system and are significantly implemented using quantization.

Erkin et al. [32] presented an efficient protocol for biometric face recognition systems based on Eigenfaces [37], shown in fig. 2.5. Once more, the *honest-but-curious* entity, responsible for the comparison process, is unable to gain any privacy-sensitive information.

Another secure computation of the face identification protocol was introduced in [33]. The algorithm is based on fixed-length templates with a constant Hamming weight, outperforming the Eigenface approach in terms of efficiency and robustness, because computation of the Euclidean distances are more complicated than that of the Hamming distance.

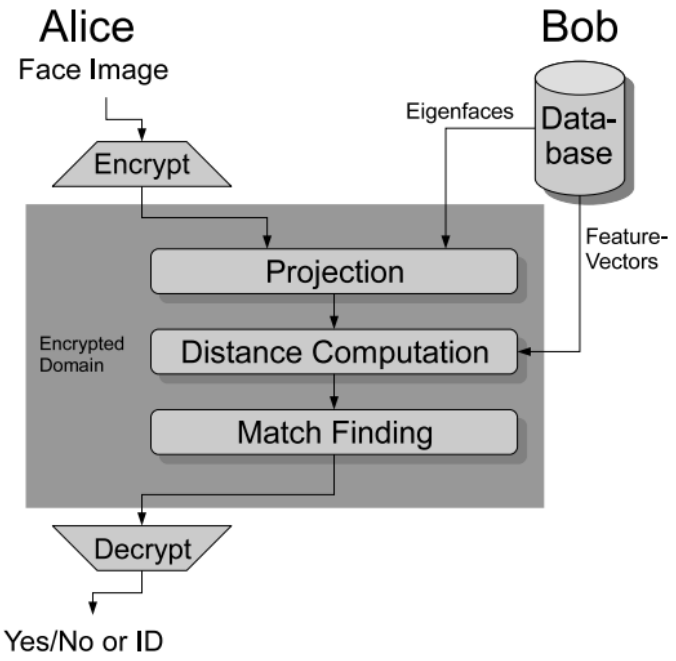

Figure 2.5: Efficient protocol for face recognition, cf. [32].

In [34], the client protocol called *GSHADE*, based on *oblivious transfers* [27], evaluates several metrics (Hamming distance, Euclidean distance, scalar product, Mahalanobis distance) and shows a significant improvement in terms of computation time towards the techniques based on HE protocols.

The listing of the reviewed papers with emphasis on Paillier encryption scheme for anonymous biometric authentication protocols, is given by tab. 2.1.

REFERENCE	CHARACTERISTIC	HE SCHEME	DATASET
Ye et al.[28]	Iris	Paillier	CASIA
Penn et al.[29]	Iris	Paillier, Goldwasser-Micali	CASIA V3 Interval
Barni et al.[30]	Fingerprint	Paillier	408 images acquired by a CrossMatch Verifier 300 sensor
Bianchi et al.[30]	Fingerprint	Paillier	408 images acquired by a CrossMatch Verifier 300 sensor
Erkin et al.[32]	Face	Paillier, DGK	"ORL Database of Faces" from AT&T Laboratories Cambridge
Osadchy et al.[33]	Face	Paillier	FERET

Table 2.1: Listing of papers dealing with Paillier encryption scheme for anonymous biometric authentication protocols.

2.4 SPEAKER RECOGNITION

Speaker verification systems are commonly divided into front-end and back-end stages [38]. A front-end is responsible for converting of speech signal into a sequence of the features vectors. Back-end provides speaker modelling and score computation.

2.4.1 *Front-End*

Because of the variable length and duration of acoustic speech signal, most of these variations are undesirable and cause mismatch between the training and testing conditions. Thus, normalization and adaptation methods are mandatory to minimize the mismatch.

2.4.1.1 *Voice Activity Detection*

Voice Activity Detection (VAD) [39] is a pre-processing algorithm used to distinguish between the presence or absence of human speech in

order to extract features only from the speech segments. The VAD is usually performed in two steps:

1. Features from the noisy section are extracted to get a representation that discriminates between speech and noise.
2. A detection algorithm is used to classify the section as speech or non-speech.

2.4.1.2 *Low-level features: sampling the acoustic space*

The analysis of the frequency spectrum of the signal is the most used technique to extract features. For this purpose, signal is usually segmented into 20-30 ms overlapping frames, and the Mel Frequency Cepstral Coefficents (MFCCs) [40] are obtained. For each frames the Fourier power spectrum is calculated. Nonlinearly spaced Mel filterbank is applied to the power spectra. The spectrum energy is summed in each filter, following by calculation of its logarithm. Finally, discrete cosine transform (DCT) is performed on the logarithm of the filterbank energy, and a number of leading DCT coefficients characterize MFFCs.

2.4.1.3 *Feature Normalization*

In order to provide more robustness to degradation among different speech environments, various feature normalisation techniques are proposed. These techniques can be distinguished between model-based and distribution-based ones. In the model-based approaches, such as cepstral mean and variance normalization [41], certain statistical properties are normalized to minimize the environmental mismatch. In the second category fall those approaches, that normalize the feature distribution to a reference distribution, such as feature warping [42] or short-time Gaussianization [43].

2.4.2 *Back-End*

The next step is modelling based on extracted features. Different speakers have different subspaces within the UBM [44], an universal acoustical cluster. Baum-Welch statistics [45] are extracted from UBM and represented in intermediate-sized vectors (i-vectors) [46], that effectively summarize utterances, allowing one to apply useful compensation methods because of its low-dimensional space.

2.4.2.1 *Gaussian Mixture Models*

A *Gaussian mixture model (GMM)* [47] is a parametric probability density function, defined as the weighted sum of Gaussian density components. Given C components, where $c = 1, 2 \ldots C$ and π_c represent

the component indices and the mixture weights of D-dimensional feature vector x_n, respectively, a mixture density for speaker S is defined as:

$$p(x_n|\lambda_s) = \sum_{c=1}^{C} \pi_c \mathcal{N}(x_n|\mu_c, \Sigma_c). \tag{2.28}$$

For a sequence of n training vectors $X = \{x_1, ..., x_N\}$ a mixture density is given by:

$$p(X|\lambda_s) = \prod_{n=1}^{N} p(x_n|\lambda_s). \tag{2.29}$$

The component densities are parametrized by a mean vector μ_c and covariance matrix Σ_c :

$$p_c(x) = \frac{1}{(2\pi)^{D/2}|\Sigma_c|^{1/2}} e^{-\frac{1}{2}(x-\mu_c)^T(\Sigma_c)'(x-\mu_c)} \tag{2.30}$$

The mixture weights, π_c, summed up, are resulting to:

$$\sum_{c=1}^{C} \pi_c = 1, \quad \forall c : 0 \leq \pi_c \leq 1. \tag{2.31}$$

The GMM for speaker s is defined as the tupel:

$$\lambda_s = (\pi_c, \mu_c, \Sigma_c). \tag{2.32}$$

In order to reduce the computational and improve performance, the linear combination of diagonal covariance basis Gaussians is used.

2.4.2.2 *Expectation-maximization algorithm*

Conventionally, GMMs are trained using the *expectation- maximization (EM)* algorithm [48]. The aim of this algorithm is to iteratively update the parameter values in eq. 2.32 in order to maximize the likelihood of the training data, given the current model.

The posterior probability γ_n of a mixture c given feature vector x_n is calculated:

$$\gamma_n(c) = p(c|x_n, \lambda_s) = \frac{\pi_c \, p(x_n|c, \lambda_s)}{\Sigma_{j=1}^{C} \pi_j \, p(x_n|c, \lambda_j)}. \tag{2.33}$$

In the M-step, the weights, means and variances of GMM are re-estimated:

$$\pi_j = \frac{1}{N} \sum_{n=1}^{N} \gamma_n(c), \tag{2.34}$$

$$\mu_j = \frac{\Sigma_{n=1}^{N} \gamma_n(c) x_n}{\Sigma_{n=1}^{N} \gamma_n(c)}, \tag{2.35}$$

$$\Sigma_j = \frac{\Sigma_{n=1}^{N} \gamma_n(c)(x_n - \mu_j)(x_n - \mu_j)^T}{\Sigma_{n=1}^{N} \gamma_n(c)}. \tag{2.36}$$

In the last step, log likelihood evaluation is performed:

$$\Lambda(X|\lambda_s) = \frac{1}{N} \sum_{n=1}^{N} log \sum_{s=1}^{S} \pi_c p_c(x_n), \tag{2.37}$$

where $p_c(x_n)$ is calculated as given in eq. 2.30.
The process is iterated until convergence.

2.4.2.3 *Universal background model*

In order to compute LLRs, an alternate speaker model is needed, usually referred to as *Universal background model (UBM)*. The UBM λ_{UBM} represents the speaker-independent distribution of features for all speakers in general.
In the GMM-UBM approach [44], LRs are computed utilizing GMMs representing either of the hypotheses: H_0(X is from s) and H_1(X is not from s):

$$LR = \frac{p(X|H_0)}{p(X|H_A)} = \log p(X|\lambda_s) - \log p(X|\lambda_{UBM}) \tag{2.38}$$

Usually, the data amount per speaker is limited, hence UBMs are computed first, such that speaker dependent models can be adapted.

2.4.2.4 *Maximum a posteriori adaptation*

Maximum a posteriori adaptation (MAP) is a common technique to adapt the speaker's model from UBM. MAP was proposed in [49], Reynolds et al. [44] applied this technique for speaker recognition. At first, the probabilistic alignment of the feature vector is calculated through the use of eq. 2.33 with the difference, that it is estimated with respect to UBM components:

$$\gamma_n(c) = p(c|x_n, \lambda_{UBM}) = \frac{\pi_c \, p(x_n|c, \lambda_{UBM})}{\Sigma_{c=1}^{C} \pi_c \, p(x_n|c, \lambda_{UBM})} \tag{2.39}$$

The values of $\gamma_n(c)$ are used to calculate the zero-, first-, and second-order Baum–Welch statistics [45]:

$$N_s(c) = \sum_{n=1}^{N} \gamma_n(c) \tag{2.40}$$

$$F_s(c) = \sum_{n=1}^{N} \gamma_n(c) x_n, \tag{2.41}$$

$$S_s(c) = \sum_{n=1}^{N} \gamma_n(c) x_n x_n^T. \tag{2.42}$$

MAP updates weight, mean and covariance for each mixture c:

$$\bar{\pi}_c = [\alpha_c N_s(c)/N + n(1-\alpha_c)\pi_c]\beta, \tag{2.43}$$

$$\bar{\mu}_c = \alpha_c \frac{F_s(c)}{N_s(c)} + (1-\alpha_c)\mu_c, \tag{2.44}$$

$$\bar{\Sigma}_c = \alpha_c \frac{S_s(c)}{N_s(c)} + (1-\alpha_c)(\Sigma_c + \mu_c \mu_c^T) - \mu_c \bar{\mu}_c^T, \tag{2.45}$$

where β is a scaling factor, computed over all the adapted mixture weights to ensure that they satisfy the constraint, given by eq. 2.4.2.1, and α is an adaptation variable, given by:

$$\alpha_c = \frac{n_c}{n_c + r}, \tag{2.46}$$

where r is a relevance factor, controlling, how the new speaker data affects the adapted parameters.

2.4.2.5 *Supervector*

One of the problem in speaker verification systems is to compare data of different length, allowing one to use such modelling techniques as factors analysis (FA) or support vector machine (SVMs) [50]. Thus, a fixed-dimensional representation of an utterance is needed. The solution is provided by *supervectors*, that represent the concatenations of GMM mean vectors. A supervector μ of dimension CF, where C is the number of Gaussian mixture components and F is the dimension of the acoustic feature vectors, contains the means of each mixture component. The simplification of this process is shown in fig. 2.6.

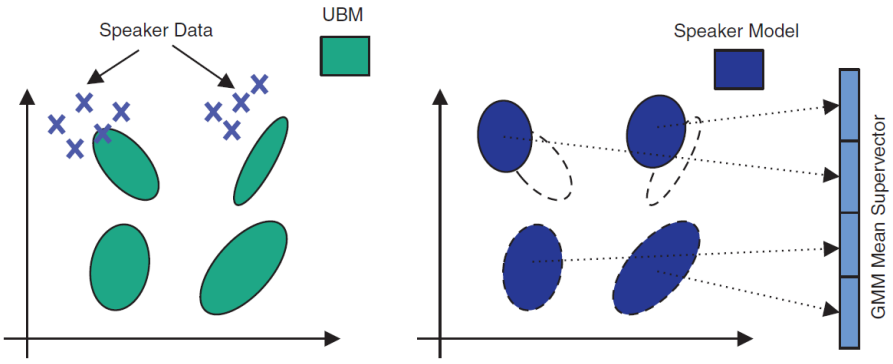

Figure 2.6: Concatenation of supervector performed on four Gaussian mixture components, based on [7].

2.4.2.6 *Joint Factor Analysis*

In practice, channel factors cannot be ignored in estimating the supervector μ. So the calculation of the joint posterior distribution of the

hidden variables x and y is needed to represent channel and speaker residual factors, respectively.
Utilizing *Joint Factor Analysis (JFA)*, supervectors can be represented as a linear combination of speaker independent, speaker dependent, channel dependent, and residual components.
A speaker-dependent GMM mean supervector μ_s is defined as:

$$\mu_s = \mu_{UBM} + Ux + Vy + Dz, \tag{2.47}$$

where speaker- and session-independent supervector μ_{UBM} is conventionally referred to as UBM, since both speaker and channel variability lie in lower dimensional subspaces (spanned by the matrices V und U, respectively) of the GMM supervector space, as it shown in fig. 2.7. In order to train the JFA model, at first the subspaces from development corpora need to be estimated (V followed by U and D) in order to obtain the speaker and session factors (x and y followed by z), for a given new target utterance. Estimation is performed by maximum likelihood and minimum divergence algorithms [51].

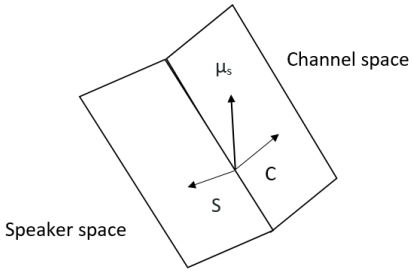

Figure 2.7: JFA model. μ_s presented as a sum of speaker components $S = \mu_{UBM} + V_y + D_z$ and channel components $C = U_x$.

2.4.2.7 *Intermediate-sized vectors (i-vectors)*

Dehak et al. [46] show that using JFA approach, it is not possible to completely separate speaker and channel variabilities, because the channel space contains speaker-dependent information, either. For this reason, [46] proposed the total variability space, that models both variabilities. The supervector μ_s is given by:

$$\mu_s = \mu_{UBM} + Tw \tag{2.48}$$

where μ_{UBM} is the UBM mean supervector, T is the low rank rectangular matrix representing the total variability space and estimated by *EM*. Conceptually, i-vectors w are hidden variables, which are assumed to be standard Gaussian distributed and can be estimated for a given utterance from $U = \{u_1, ..., u_k\}$, represented by a set of acoustic feature vectors of dimension F, by its posterior distribution, using Baum–Welch statistics from the UBM λ_{UBM} with C mixture components. The i-vector can be obtained as:

$$w = (I + T^t\Sigma^{-1}N(u)T)^{-1}T^t\Sigma^{-1}\tilde{F}(u), \tag{2.49}$$

where:

- I is a $CF \times CF$ identity matrix,
- $N(u)$ is a diagonal matrix with F x F blocks $N_c I(c = 1, 2, ..., C)$,
- $\tilde{F}(u)$ is a supervector of dimension CF x 1 formed by concatenating of the centralized first-order Baum-Welch statistics $\tilde{F}$ for a given speeach utterance u.
- The diagonal covariance matrix Σ of dimension $CF \times CF$ builds the residual variability not captured by T.

It is desired, that i-vectors are transformed into *unit sphere* to reduce the mismatch between training and testing i-vectors, thus optimizing the clustering. For that purpose, length normalization of the i-vectors is performed. Each i-vector is divided by its length. In order to avoid the concentration of the i-vectors in a small region of the unit sphere and the resulting decrease of discriminative power, centering and whitening of the i-vectors need to be performed before the length normalization. In the centering step, the global mean of the development set is subtracted from each i-vector, centering it at the origin of coordinates. In the whitening step, the i-vector space is normalized, by turning their covariance matrix into the identity matrix.

2.4.2.8 *Linear Discriminant Analysis*

The aim of *Linear Discriminant Analysis (LDA)* [52] is to reduce dimensionality and between-variabilities, at the same time preserving discriminant information. For this purpose, LDA searches for a reduced set of eigenvalues $A_{LDA} = [v_1 ... v_n]$ in the feature space, decomposing the following eigen-problem:

$$Bv = \Lambda W v, \tag{2.50}$$

where Λ is the diagonal matrix with the eigenvalues, B and W represent between-and within-class covariance matrices, respectively. In order to calculate both covariance matrices, the speaker-dependent μ_s and speaker-independent μ mean vectors need to be obtained:

$$\mu_s = \frac{1}{n_s} \sum_{i=1}^{n_s} w_{s,i}, \tag{2.51}$$

$$\mu = \frac{1}{S} \sum_{s=1}^{S} \frac{1}{n_s} \sum_{i=1}^{n_s} w_{s,i}, \tag{2.52}$$

where $w_{s,i}$ is an i-vector, extracted from ith utterance of speaker s from the number of utterances n_s of the same speaker, and S is the total number of all speakers in the set.
Between-and within-class covariance matrices are calculated:

$$B = \frac{1}{S} \sum_{s=1}^{S} (\mu_s - \mu)(\mu_s - \mu)^T, \tag{2.53}$$

$$W = \frac{1}{S}\sum_{s=1}^{S}\frac{1}{n_s}\sum_{i=1}^{n_s}(w_{s,i} - \mu_s)(w_{s,i} - \mu_s)^T. \tag{2.54}$$

Examining matrix of eigenvectors A_{LDA}, LDA maximizes S and minimizes W, respectively. The LDA projection of i-vector is calculated as follows:

$$\hat{w}_{LDA} = A_{LDA}^T w. \tag{2.55}$$

2.4.2.9 *Within Class Covariance Normalization*

Hatch et al. [53] introduced the *Within Class Covariance Normalization (WCCN)* approach to improve robustness of SVM-based speaker verification systems. Dehak et al. [52] proposed to use WCCN in i-vector approach to compensate intersession variability and guarantee conservation of directions of the feature space.
WCCN projection matrixes are obtained by the Cholesky decomposition of A_{WCCN}:

$$S_w^{-1} = A_{WCCN}A_{WCCN}^T. \tag{2.56}$$

WCCN projected i-vectors are obtained by:

$$\hat{w}_{WCCN} = A_{WCCN}^T w. \tag{2.57}$$

2.4.3 *Comparison*

In the following subsection two relevant in terms of this work scoring methods for comparison between i-vectors are given.

2.4.3.1 *Cosine similarity*

The cosine distance similarity scoring for speaker recognition was proposed in [52]. Scoring is defined as normalized dot product of reference i-vector w_r and probe i-vector w_p:

$$S_{cos}(w_r, w_p) = \frac{w_r \cdot w_p}{||w_r|| \cdot ||w_p||}. \tag{2.58}$$

2.4.3.2 *Probabilistic Linear Discriminant Analysis*

Probabilistic Linear Discriminant Analysis (PLDA) is a generative model which was adapted from face recognition [54] for speaker verification purposes, where PLDA is used to model i-vector distributions, distinguishing between-speaker variability from all other sources of undesired variability that cause distortions.
Kenny at al. [55] proposed a Heavy–Tailed PLDA model (HT-PLDA), based on heavy–tailed distributions for the model priors. However, this model is computationally expensive. For that reason, the most commonly used model is a Gaussian PLDA (G-PLDA) model, that using length normalized i-vectors [56] with LDA and WCCN, shows

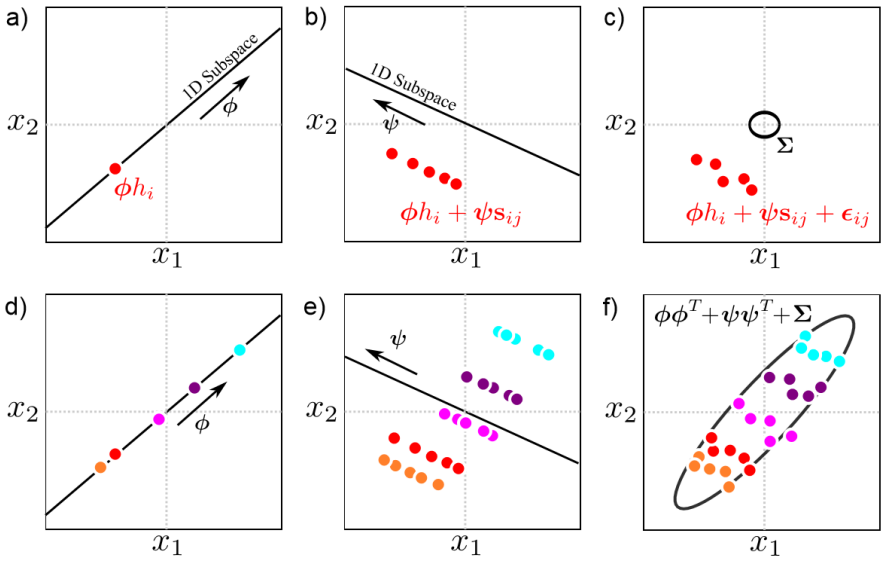

Figure 2.8: PLDA model [57]. a) Latent variable h_i is used to weight between-individual factors Φ. b) J latent variables $\{s_{ij}\}_{j=1}^{J}$ are used to weight within-individual factors Ψ. c) Adding residual noise with diagonal covariance Σ. d-f) The process is repeated for several speakers.

the comparable performance to HP-PLDA, but beeing computationally faster. In PLDA i-vector is presented as a combination of the follow components:

$$w_{ij} = \mu + \Phi h_i + \Psi s_{ij} + \epsilon, \tag{2.59}$$

where μ is the speaker-independent mean i-vector, low-rank matrices Φ and Ψ span the speaker identity subspace and channel subspace, respectively, and ϵ represents a residual noise with a zero mean and diagonal covariance Σ. The latent variables h_i and s_{ij} are assumed to follow a standard Gaussian distribution $\mathcal{N}(0, I)$. The PLDA modeling is shown in fig. 2.8.
In G-PLDA [56] Σ is often assumed to be a full covariance matrix and allows one to remove Ψ without performance loss:

$$w_{ij} = \mu + \Phi h_i + \epsilon. \tag{2.60}$$

2.4.3.3 *Two-covariance model*

The full-subspace PLDA model is referred to as the *two-covariance model (2Cov)* [58, 59], assuming that the speaker and inter–session subspaces span the entire i–vector space, thus, $rank(\Phi) = rank(T)$. Thus, an i-vector can be represented as:

$$w_{ij} = \mu + h_i + \epsilon, \tag{2.61}$$

where $h_{ij} \sim \mathcal{N}(h_i|\mu, B^{-1})$, $w|h_{ij} = \mathcal{N}(w|h_i, W^{-1})$ and B and W are the between–speaker and within–speaker covariance matrix, obtained

by eq. 2.53 and 2.54, respectively. 2Cov log–likelihood ratio for an i–vector pair is a quadratic function and can be given as [60]:

$$\begin{aligned} S_{2cov}(w_r, w_p) &= w_r^T \Lambda w_p + w_p^T \Lambda w_r + w_r^T \Gamma w_r + w_p^T \Gamma w_p \\ \Lambda &= \frac{1}{2} W^T \left((B + 2W)^{-1} \right) W^{-1} \\ \Gamma &= \frac{1}{2} W^T \left((B + 2W)^{-1} + (B + W)^{-1} \right) W. \end{aligned} \tag{2.62}$$

3

PROPOSED METHOD

The proposed methods utilize the Paillier homomorphic properties for transferring data privacy methods to speaker recognition. The proposed methods demonstrate how cosine and two-covariance comparators behave in the encrypted domain in terms of data and storage management as well as communication between the single parts of the system.

3.1 COMPUTATIONS ON NON-INTEGER VALUES

Paillier homomorphic encryption is solely defined for non-negative integers smaller than public key n. Thus, the representation of negative and real numbers are mandatory before they will be encrypted.

- Conversion of real values into integers: The conversion of real number into integers is implemented based on floating-point arithmetic. According the IEEE Standard for Floating-Point Arithmetic, defined by IEEE Std 754-2008 [61], each finite number can be represented by three integers, using single precision (32-bit) and double precision (64-bit) format:

$$float = s \cdot m \cdot 2^e, \tag{3.1}$$

  where s is a sign (s = 0 indicates the positive number, s = 1 - negative), m is a mantissa, that defines a fixed number of significant digits (p bits), scaled by exponent e (r bits) in base 2. The decision was made for the double precision because of more bits that mantissa (52 bits) and an exponent (11 bits) occupy in contrast to the single precision (23 and 8 bits, respectively). Thus, it is possible to increase the precision and the range of magnitudes that can be represented. The single and double formats of IEEE 754-2008 are shown in tab. 3.1.
  The main objective of this step is to find the mantissa, that stands for the integer representation and save the exponent for the following additive and multiplicative operation and the decoding of the final value. In case of addition of two encoded values, the highest exponent needs to be decreased to the lowest exponent, in case if multiplication the encoded values are just summed up.

- Conversion of integers into whole numbers: The basic property of congruence $a \bmod n = a + kn$ for any $k \in \mathbb{N}$ is applied, by

adding multiples of n to the a in order to get a positive representation of a. The maximum integer value int_{max} is defined in order to create two number ranges: all values smaller than int_{max} are positive and all values greater than $n - int_{max}$ are negative (It is actually the case since the property of congruence converts negative numbers into very huge positive values). The range between int_{max} and $n - int_{max}$ is used for overflow detection.

LEVEL	WIDTH	EXPONENT (R)	MANTISSA (P)	EXPONENT VALUES (E)
single	32 bit	8 bit	23 bit	-126≤e≤127
double	64 bit	11 bit	52 bit	-1022≤e≤1023

LEVEL	RANGE AT FULL PRECISION	PRECISION
single	$\approx 1 \cdot 10^{-38}$ to $\approx 3 \cdot 10^{38}$	≈ 7 decimal digits
double	$\approx 2 \cdot 10^{-308}$ to $\approx 2 \cdot 10^{308}$	≈ 16 decimal digits

Table 3.1: Single and double formats of IEEE 754-2008. [61]

3.2 CONVENTIONAL ENCRYPTION SCHEME FOR COSINE SIMILARITY

3.2.1 *Background*

As mentioned in sect. 2.4.3.1, one of the scoring techniques is the cosine similarity, that is defined as the length normalised dot product of probe w_p and reference w_r i-vectors. According to [62], eq. 2.58 can be rewritten as:

$$S_{cos}(w_r, w_p) = \frac{w_r \cdot w_p}{||w_r|| \cdot ||w_p||} = \sum_{f=1}^{F} \frac{w_{r_f} \cdot w_{r_f}}{||w_p|| \cdot ||w_r||} = \sum_{f=1}^{F} \frac{w_{p_f}}{||w_p||} \cdot \frac{w_{r_f}}{||w_r||}, \tag{3.2}$$

where f defines one of F i-vector dimensions.

Thus employing Paillier homomorphic property in eq. 2.3.2.5, the cosine similarity can be encrypted as:

$$enc(S_{cos}) = \prod_{f=1}^{F} enc\left(\frac{w_{r_f}}{||w_r||}\right)^{\frac{w_{t_f}}{||w_t||}} \tag{3.3}$$

where the encrypted reference i-vectors $enc(w_r)$ are raised to the corresponding values of w_p, calculating the encrypted score, that decrypted returns the same result as a dot-product of two plain i-vectors, given by eq. 3.2.

In case of storing more than one reference i-vector for a certain user, all single scores must be summed up. Based on the property in eq. 2.3.2.5, the product of all single scores between probe i-vector w_p and the reference i-vectors w_r^n, with $n = 1...N$, in encrypted domain is calculated:

$$dec\left(\prod_{N}^{n=1} enc\left(S_{cos}^{n}\right)\right) = \sum_{N}^{n=1} S_{cos}^{n}. \tag{3.4}$$

3.2.2 *Architecture*

The architecture of the cosine approach is motivated by the architecture, proposed in [62]. There are three system entities:

The **client** C is responsible for the extraction of probe i-vectors w_t, computation of the encrypted score $enc(S_{cos})$ and sending it to the comparator.

The **database server** $DB_{subject}$ stores the encrypted reference i-vectors $enc_{pk}(w_r)$ and send them to C during verification.

The **authentification server** $AS_{subject}$ is a comparator of the system, that holds a key pair, that consists of secret key sk and public key pk. $AS_{subject}$ decrypts the score after receiving it from the C, in order to make a final decision about rejection or acceptance based on a predefined threshold.

The verification includes the following six steps, as shown in fig. 3.1:

1. C captures the probe audio sample and extracts a probe i-vector w_p, following compensation methods.
2. $DB_{subject}$ sends the beforehand stored and encrypted with the public key pk reference i-vectors $enc_{pk}(w_r)$ to the client.
3. C computes the encrypted score $enc_{pk}(S_{cos})$. If w_p and w_r are unit sphere i-vectors, the eq. 3.3 can be simplified to $enc_{pk}(S_{cos}) = \prod_{f=1}^{F} enc_{pk}(w_{r_f})^{w_{t_f}}$.
4. C sends $enc_{pk}(S_{cos})$ to the authentication server $AS_{subject}$.
5. $AS_{subject}$ decrypts the $enc_{pk}(S_{cos})$, using secret key sk.
6. The decision D about the acceptance or rejection of the system user is made based on pre-defined threshold η.

It is assumed, that the model is honest-but-curious, as described in sec. 2.2. Thus, all parties perform the actions correctly, but try to learn about other parties' inputs. For the malicious scenario, it must be pointed out that only one of three entities (C, DB_{ref},AS_{ref}) can be

compromised at a time, and if this is the case, the attacker cannot attacks the other entities. Additionally, all entities of the system are separated and never collude.

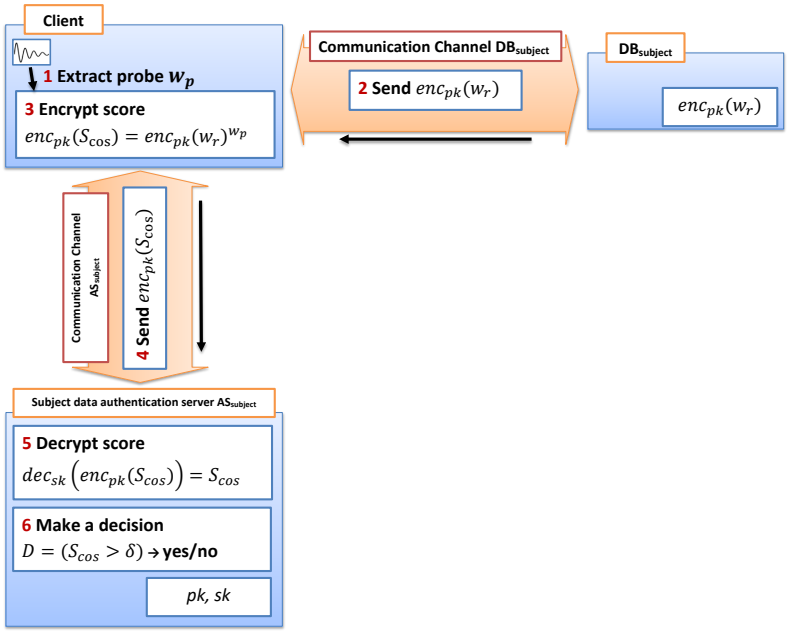

Figure 3.1: Architecture of cosine similarity. A client C extracts the probe i-vector w_p and requests the reference i-vector $enc_{pk}(w_r)$ from the database $DB_{subject}$. The final score is calculated in C, using Paillier HE, and send to the authentication server $AS_{subject}$, that holds the key pair (pk, sk) for the following decryption of the score. Based on pre-defined threshold, $AS_{subject}$ outputs the decision D.

3.3 ENCRYPTION SCHEME FOR FULL SUBSPACE PLDA COMPARATORS

Λ and Γ are obtained from a large collection of labeled development data, so the 2Cov training can be highly expensive. Thus, in the interests of system vendors to encrypt model hyper-parameters, which incorporate the knowledge from the system training data as well as the processing and development costs.

3.3.1 *Privacy Architecture solely for Subject References*

3.3.1.1 *Background*

The Paillier scheme can be adopted for the 2Cov comparator. The scoring in unencrypted domain is obtained, using eq. 2.62:

$$S_{2cov}(w_r, w_p) = w_r^T \Lambda w_p + w_p^T \Lambda w_r + w_r^T \Gamma w_r + w_p^T \Gamma w_p. \tag{3.5}$$

where Λ and Γ are the model hyper-parameters, holding between and within covariances, respectively. The assumption is, that the model hyper-parameters Λ and Γ are freely available to C and $DB_{subject}$, so applying the Paillier properties to the eq. 3.5, the encrypted score can be estimated as:

$$\begin{aligned} enc(S_{2cov}(w_r, w_p)) = enc(w_r^T)^{\Lambda w_p} + enc(w_r)^{\Lambda w_p^T} \\ + enc(w_r^T)^{\Gamma w_r} + enc(w_p^T)^{\Gamma w_p}. \end{aligned} \tag{3.6}$$

For the purpose of later model encrpytion, the terms $enc(w_p^T)^{\Gamma w_p}$ and $enc(w_r^T)^{\Gamma w_r}$ are not further simplified.

3.3.1.2 *Architecture*

The verification includes the following steps, illustrated in Figure 3.2:

0. During enrolment reference i-vector w_r is stored in $DB_{subject}$ as a tupel, containing the encryption of the i-vector $enc(w_r)$, the encrypted transpose $enc(w_r^T)$ and the encrypted component $enc(w_r^T)^{\Gamma w_r}$, where the multiplication of Γ and w_r is performed in the plaintext domain. Hence, this component, used in the eq. 3.6, can be encrypted only during enrolment, since w_r is coming unencrypted to $DB_{subject}$.
1. The client C captures the probe audio sample and extracts a probe i-vector w_p and encrypts the second addend of the eq. 3.6, $enc(w_p^T)^{\Gamma w_p}$.
2. $DB_{subject}$ sends a tuple to the client.
3. The third addend of the eq. 3.6, $enc(w_r^T)^{\Lambda w_p}$, is estimated, involving Λ and the both i-vectors for comparison.
4. The forth addend of the eq. 3.6, $enc(w_r)^{\Lambda w_p^T}$, is estimated, involving Λ and the both i-vectors for comparison.
5. All addends are summed up to the encrypted score, using Paillier homomorphic property in eq. 2.3.2.5.
6. The client sends $enc(S_{2cov})$ to the authentication server $AS_{subject}$.
7. $AS_{subject}$ decrypts the $enc(S_{2cov})$, using secret key sk.
8. The decision D is made based on pre-defined threshold σ.

3.3.2 *Privacy Architecture for Subjects and System Vendors*

In this section, a fully homomorphic encrypted data protection architecture is proposed for the 2Cov comparator.

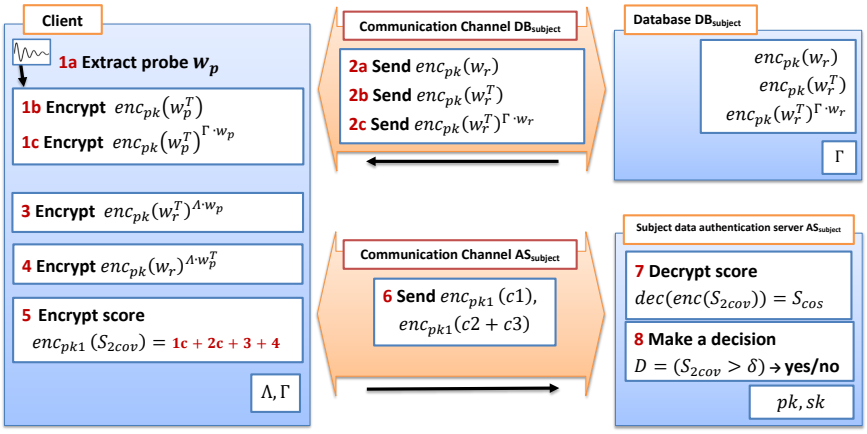

Figure 3.2: Architecture of 2Cov scoring without encryption of vendor hyper-models.

3.3.2.1 *Background*

Cumani proposed in [60], that $S_{2cov}(w_r, w_p)$ can be estimated as a dot-product in an i-vector pairs expanded space. For three matrices in the bilinear form $x^T Ay$ the Frobenius inner product is calculated as:

$$x^T Ay = \langle A, xy^T \rangle = vec(A)^T vec(xy^T), \tag{3.7}$$

where $\langle A, B \rangle$ denotes a dot-product of matrices A and B, and the operator $vec(\cdot)$ converts the matrices into column vectors. The model hypo-parameters are stacked as model vector m:

$$m = \begin{bmatrix} vec(\Lambda) \\ vec(\Gamma) \end{bmatrix} = \begin{bmatrix} w_\Lambda \\ w_\Gamma \end{bmatrix}. \tag{3.8}$$

Then, an expansion of the reference and probe i-vectors is defined as:

$$\varphi(w_r, w_p) = \begin{bmatrix} vec(w_r w_p^T + w_p w_r^T) \\ vec(w_r w_r^T + w_p w_p^T) \end{bmatrix} = \begin{bmatrix} \varphi_\Lambda(w_r, w_p) \\ \varphi_\Gamma(w_r, w_p) \end{bmatrix}. \tag{3.9}$$

S_{2cov} can be presented as the dot–product of a model vector m and an expanded i-vector pair $\varphi(w_r, w_p)$:

$$\begin{aligned} S_{2cov}(w_r, w_p) &= S_\Lambda(w_r, w_p) + S_\Gamma(w_r, w_p) \\ &= m_\Lambda^T \varphi_\Lambda(w_r, w_p) + m_\Gamma^T \varphi_\Gamma(w_r, w_p) \\ &= m^T \varphi(w_r, w_p). \end{aligned} \tag{3.10}$$

3.3.2.2 *Architecture*

In terms of Paillier cryptographic systems, eq. 3.10 fulfils either of the homomorphic properties. The encrypted models Λ and Γ are raised

to the power of an expanded i-vector pair $\varphi(w_r, w_p)$ that should be available in plaintext. Thus, the system requires another authentication sever, that is responsible for decryption of $\varphi(w_r, w_p)$ after receiving it encrypted from the client, and the calculation of the dot-product with the model vector m. The next step are mandatory and visualized in fig. 3.3):

0. Enrolling biometric users at a biometric system in terms of their reference, the following encryption of the reference with the public key pk_1 and storing it into the database $DB_{subject}$ are performed. Additionally, the encrypted multiplication result of the reference with its transpose $enc_{pk1}(w_r^{w_r^T})$ is estimated, thus both values can be combined into a single resulting tuple.

1. The client captures a probe voice sample and extract the depending i-vector w_p. Then, the client creates a tuple of encrypted i-vector $enc_{pk1}(w_p)$ and encrypted element-wise square of the i-vector $enc_{pk1}(w_p^{w_p^T})$.

2. $DB_{subject}$ send the encrypted tuple to C.

3-4. Having two encrypted tuples: one sent from $DB_{subject}$ and one created from a probe, the client estimates two encrypted components $enc(c_1)$ and $enc(c_2)$, applying the homomorphic properties to the eq. 3.9.

5. The client sends the encrypted components $enc(c_1)$ and $enc(c_2)$ to the subject data authentication server $AS_{subject}$.

6. The model comprising between and within covariances is encrypted with the second public key pk_2 (the corresponding private key sk_2 has only the vendor authentication server AS_{vendor}) are sent to $AS_{subject}$.

7-9. $AS_{subject}$ decrypts firstly the both components with the secret key sk_1 and use the encrypted models to calculate the encrypted final score $enc(S_{2cov})$, according to the eq 3.10.

10. $AS_{subject}$ sends $enc(S_{2cov})$ to AS_{vendor}.

11-12. AS_{vendor} uses a secret key sk_2 to decrypt $enc(S_{2cov})$ and makes a decision $D = (S_{2cov} > \sigma)$, based a pre-defined threshold σ. The decision outcome is decrypted on a vendor server, which can be passed back to an application depending on its design, employing conventional cryptographic methods.

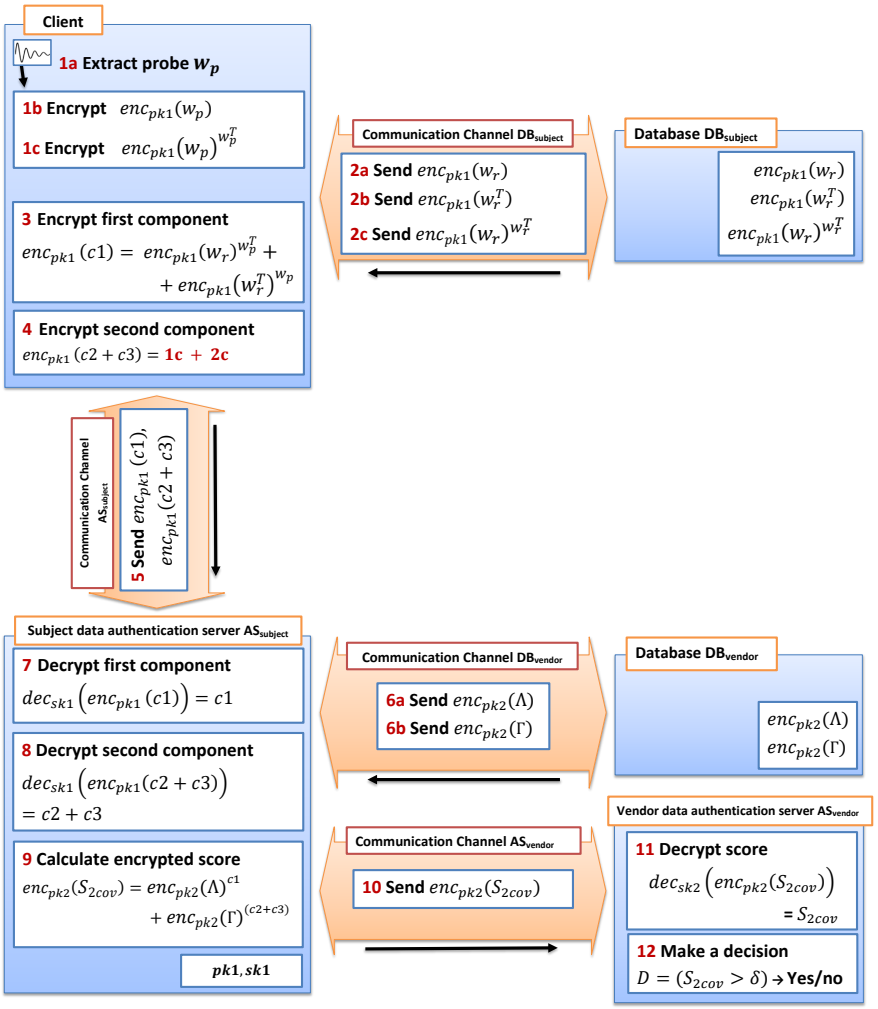

Figure 3.3: Architecture of 2Cov scoring with encryption of vendor hyper-models.

3.3.3 *Mean adaptation*

Considering the non-zero mean feature spaces e.g., in order to compensate language domain shifts, a mean vector μ needs to be accounted for as well. In this section, encryption for the generalized 2Cov model is depicted. Stored in DB_{vendor}, it is requested by $AS_{subject}$ to perform necessary data shift before the score calculation. According to [60], two additional hyper-parameters k and c, based on μ, are obtained:

$$\begin{aligned} c &= W^T \left((2W+B)^{-1} - (W+B)^{-1} \right) B\mu \\ k &= \overline{k} + 0.5 \left((B\mu_l)^T \left((2W+B)^{-1} - 2\,(W+B)^{-1} \right) B\mu \right), \end{aligned} \tag{3.11}$$

where the parameter k is calculated as:

$$\overline{k} = 2 * \log (W+B)^{-1} - \log (B) - \log (2W+B)^{-1} + \mu^T B \mu. \tag{3.12}$$

c and k are stacked into the column vector in eq. 3.8 as m_c and m_k. An expended i-vector pair in eq. 3.9 is extended by $w_r + w_p$ and 1 as φ_c and φ_k. Hence, the third component c_3, defined as the encrypted sum of w_r and w_p needs to be calculated in C. After the decryption in $AS_{subject}$, $m_c^T\, \varphi_c(w_r, w_p)$ and $m_k^T\, \varphi_k(w_r, w_p)$ are added to the score, obtained in eq. 3.10.

3.4 I-VECTOR ENCRYPTOR

Conventional front-end and back-end following the i-vector / PLDA paradigm is used for obtaining the necessary input data for the Paillier encryption. The flow of the implementation process is clearly perceptible. Firstly, the system generates a public and private key pair and returns the instances of classes **PublicKey** and **PrivateKey**, that contain the encryption and decryption methods, respectively. Every single value that represents one of dimensions of the input i-vector is encrypted by method *encrypt*() of **PublicKey**. Due to the fact, that i-vector are real valued features and Paillier works solely on integers, the corresponding encoding method *encode*() of the newly created instance of class **EncodedValue** is applied as a mandatory preprocessing step. Having the **EncodedValue** instance, it is possible to create an instance of the class **EncryptedValue**, that contains the ciphertext and the exponent of encoding. All necessary operations in the encrypted domain, such as *add*() for addition of two encrypted values, or *mult*() for the multiplication of encrypted value with plaintext, are performed by the methods of the **EnctryptedValue** instance.
On the other hand, the instance of **PrivateKey**, responsible for decryption, returns the decrypted and decoded plaintext of **EncryptedValue**, using *decrypt*() and *decode*() methods of **PrivateKey** and **EncodedValue** instances, respectively. All classes of the Paillier proposed algorithm are listed in tab 3.2.

CLASS	FUNCTION
PublicKey	Holds a public key and corresponding encryption method
PrivateKey	Holds a private key and corresponding decryption method
PrivateKeyBundle	Collects a number of private keys, using any of them for decryption
EncodedValue	Represents a float or integer value encoded for encryption
EncryptedValue	Represents the encryption of a float or integer value.

Table 3.2: Classes of the proposed Paillier algorithm.

4 EVALUATION

In the following, the evaluation of the proposed methods is performed in terms of various metrics and complexity.

4.1 EXPERIMENTAL SET-UP

The i-vectors from *Speaker Recognition i-Vector Machine Learning Challenge* [63] are used for implementation of the proposed encryption schemes. *Sidekit* [64] provides the chain of tools required to perform speaker recognition.

- **The enroll/test partition** consists of 1,306 target speaker models (comprised of 6,530 i-vectors) and 9,634 test i-vectors.
- **The development partition** consists of 36,572 i-vectors, used for the calculation of the system's hyper-parameters.
- **The trials** consist of all possible pairs defining the comparison between target speaker model and a single i-vector test segment. Thus the total number of trials is 12,582,004. In terms of the Challenge the trials were divided into two subsets: a progress subset (40% of all trials), visible for all participants, and an evaluation subset (60% of all trials), used for generation of the official final score determined at the end of the Challenge. For implementation purpose only the progress subset is relevant.

In order to create comparable results, *EER* and and *DET* diagram are chosen. The application-dependent metrics *DCF* and *minDCF* are used for the evaluation of the systems, either. In terms of the biometric performance application-independent cost measures C_{llr} and C_{llr}^{min} are of great significance, since *DCFs* are dependent on a ratio of the mated and non-mated hypothesis.

The computations are performed on a Debian 8.7 64-bit system having an Intel i7-6770HQ CPU (2.60 GHz) and 32 GB DDR3-RAM.

4.2 DATA ANALYSIS

The raw i-vectors from the i-Vector Challenge are transformed according to the official baseline system, presented in [63]. Mean and whitening transformation are computed, using the i-vectors from the development set in order to center and whiten all i-vectors with the following projection into unit sphere. Since each target model consist

of five i-vectors, a speaker model is generated by taking mean over them and projecting the average model i-vector into unit sphere. The average model i-vectors defines the reference i-vectors w_r, stored in database. All the i-vectors from the i-Vector Challenge exhibit 600 dimensions, that are reduced to 250 by performing LDA. Additionally, $WCCN$ can be applied to compensate for residual channel effects in the speaker factor space. For the implementation of 2Cov approach, within and between covariances are computed based on the development set. Fig. 4.1 shows what steps are already implemented and what are need to be estimated.

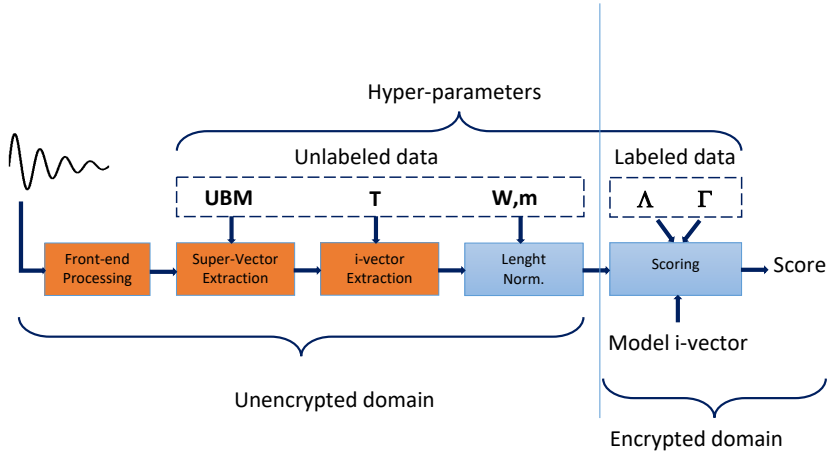

Figure 4.1: Diagram of i-vector processing and scoring, based on [63]. Orange blocks are the precomputed steps, the blue ones are need to be performed.

4.3 VALIDATION OF BASELINE ENRYPTION SCHEME ON COSINE COMPARATORS

The performance of the cosine comparator in the plaintext and encrypted domain are examined on the i-vector set, defined in fig. 4.2. The *Detection Error Trade-Off (DET)* curve for both systems is depicted for the cosine distance. The EER, $minDCF$, C_{llr} and C_{llr}^{min} are displayed in tab. 4.1. As expected, the performance is equal, moreover the proposed encryption scheme is not decreasing the baseline performance in discrimination or calibration.

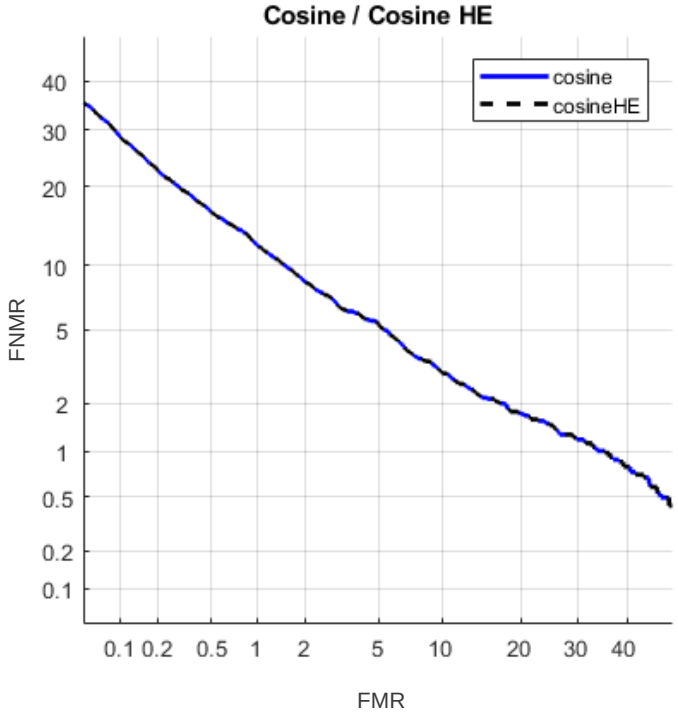

Figure 4.2: Performance analysis of for cosine similarity in the plaintext (blue) and encrypted (dashed black) domains.

	EER	$minDCF$	C_{llr}	C_{llr}^{min}
Cosine	0.0509	0.1320	0.9074	0.1893
Cosine (encrypted)	0.0509	0.1320	0.9074	0.1893

Table 4.1: EER, $minDCF$, C_{llr} and C_{llr}^{min} for cosine similarity in the plaintext and encrypted domains.

4.4 VALIDATION OF ENCRYPTION SCHEMES ON 2COV COMPARATORS

The performance of the 2Cov comparators are analysed in terms of the EER, $minDCF$, C_{llr} and C_{llr}^{min}, as shown in tab. 4.2, and visualized in fig. 4.3, again equal performance is sustained by either of the proposed homomorphic encrpytion approaches.

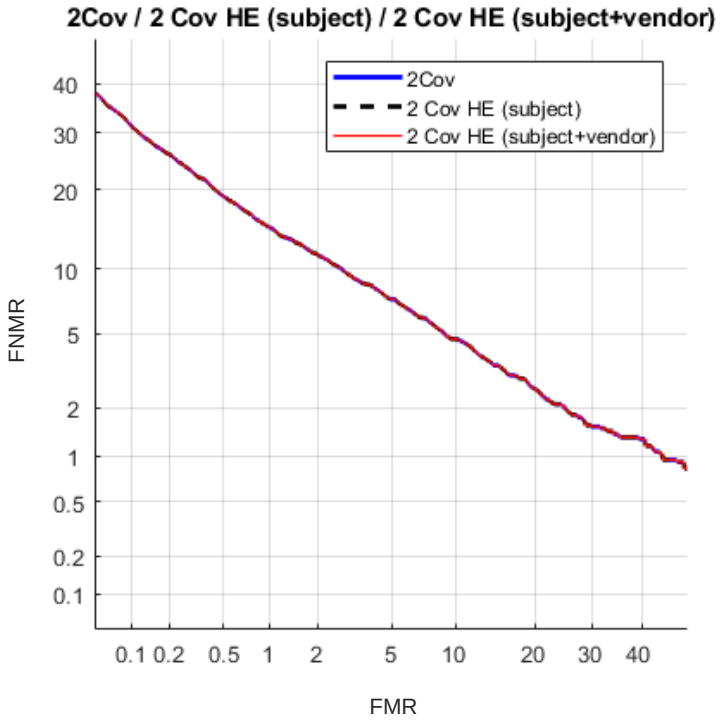

Figure 4.3: Performance analysis of 2Cov comparators in the plaintext (blue) and encrypted domains (dashed black - the system with protection of solely subject data, red - the system with protection of subject and vendor data).

4.5 COMPLEXITY ANALYSIS

Each approach can be analysed due to the amount of resources required for running it. The resources are defined as a number of operations performed in the encrypted domain as well as a size of encrypted components sent over a channel. The channel bandwidth and the database storage capacity can be estimated as:

$$capacity = r \cdot d \cdot 2n, \tag{4.1}$$

where r is a number of reference i-vectors, d defines the total number of elements of a vector or a matrix, and $2n$ is a size of chipertext in bits (since a Paillier ciphertext is a number c modulo n^2, the chipertext is $2n$ bits wide).

During the key generation the same conditions should hold for n as for the size of the modulus in the RSA cryptosystem, which is at least 2048 bits according to NIST recommendations [65]. Thus the Paillier ciphertexts, used in the proposed system, is $2 \cdot n = 4096$ bits wide. Utilizing eq. 4.1 for i-vector of dimension 250, the approach

	EER	$minDCF$	C_{llr}	C_{llr}^{min}
Cosine	0.0634	0.1609	0.9099	0.2277
2Cov (encrypted, subject)	0.0634	0.1609	0.9099	0.2277
2Cov (encrypted, subject + vendor)	0.0634	0.1609	0.9099	0.2277

Table 4.2: EER, $minDCF$, C_{llr} and C_{llr}^{min} for 2Cov comparators in the plaintext and encrypted domains.

for the cosine similarity in sect. 3.2 requires $4096 \cdot 250 = 1,024,000$ bits $=$ 128 KB for the storage of every reference i-vector w_r in the database $DB_{subject}$. The final score, transmitted to the authentication server AS_{vendor}, is a single encrypted value with the size of 0,512 KB.

In case of the 2Cov training, described in sect. 3.3, $DB_{subject}$ stores a tuple of various representations of encrypted w_r. The storage requirements increase rapidly, since the tuple involves the element-wise square of w_r. Being a matrix of dimension 250×250, with each value encrypted, it requires the multiplication by a factor of 250 and expands the storage size of DB_{ref} for every w_r to $4069 \cdot 250 \cdot 250 + 2 \cdot (4096 \cdot 250) = 258,048,000$ bits $=$ 32.256 MB. Since all the tuple values are sent to the client C, the bandwidth of the communication channel $DB_{subject}$ needs to be scaled up in the same manner. Building two components in C, the size of the chipertexts sent over communication channel to $AS_{subject}$ is $2 \cdot 4069 \cdot 250 \cdot 250 =$ 64 MB. Merely, the size of the final score is 0,512 KB.
An overview is provided in tab. 4.3.

	COSINE	2COV
N^o encryptions decryptions	0 1	4 3
N^o multiplications (encrypted)	250 (C)	500 (C) 250 ($AS_{subject}$)
N^o additions (encrypted)	249 (C)	498 (C) 249 ($AS_{subject}$)
Stored data	128 KB ($DB_{subject}$)	32,256 MB ($DB_{subject}$)
Exchanged data	128 KB ($DB_{subject} \rightarrow C$) 0,512 KB ($C \rightarrow AS_{subject}$)	32,256 MB ($DB_{subject} \rightarrow C$) 64 MB ($C \rightarrow AS_{subject}$) 0,512 KB ($AS_{subject} \rightarrow AS_{vendor}$)

Table 4.3: Complexity analysis for the scoring and the 2Cov training. The dimension of the i-vector is 250, and the comparison is performed between one probe and one reference i-vector.

4.6 ANALYSIS OF IRREVERSIBILITY AND UNLINKABILITY

According to [62], to fulfil the requirements in [4], the following conditions should be met:

- Nobody except client can see the plain probe i-vectors w_p
- Reference i-vectors are always encrypted in the system. This affects the storage in the database as well as any operations performed on them. In case of the 2Cov training, the same criterion is valid for the vendor data.
- The final score, which is sent over the communication channel, is encrypted, to prevent any score-based attacks.

Irreversibility is granted, since the reconstruction of the original i-vectors and vendor data is not possible without the knowledge of the corresponding private keys. The protection of the private keys, used for decryption of the scores or necessary components, as in a intermediate step (decryption of two components of eq. 3.9 in $AS_{subject}$ in sect. 3.3) is guaranteed by the secure implementation of authentication server(s). Due to the fact, that the score is calculated in encrypted domain, the system is also protected against Hill Climbing attack [66],

where the malicious client iteratively tries to increase the score similarity, making small changes in biometric samples.

The system ensures irreversibility, either. Since Pailier homomorphic encryption is secured against *IND-CPA*, as explained in sect. 2.3.2.6, the attacker, having two encrypted reference i-vectors, cannot guess with better probability than 1/2, which encrypted reference i-vector stems from which reference i-vector in plaintext. Also combining of encrypted i-vectors or comparing of encrypted distances does not lead to disclosure of relationships. Additionally, the randomness in the Paillier homomorphic encryption algorithm yields different ciphertexts, every time the certain feature f is encrypted with the same public key.

4.7 SUMMARY

As desired, the cosine and 2Cov comparators show the same verification performance both in unprotected and encrypted domains in terms of well-established application-dependent metrics and application-independent metrics on discrimination and calibration metrics.

In case of 2Cov comparators, the approach, that provides protection for both subject and system vendors data does not suffer any performance drop towards the approach based on assumption, that vendor data is freely available. This fact makes it more valuable for the system vendor, if the protection of e.g., labeled data and comparison models, is of high interest. A possible drawback of the 2Cov system could be the amount of data transferred over the communication channels, since certain encryption steps require the operations over the large dimensional quadratic matrices, which however is the typical trade-off for sustaining privacy and security.

The architecture of the systems is secure against honest-but-curious behaviour of the system components due to the encrypted communication between them. The attacks aiming at reconstruction of the score or the reference i-vectors, as well as cross-matching attacks, are not possible, thus fulfilling two main requirements of the ISO/IEC IS 24745 for biometric template protection, particularly: irreversibility and unlinkability.

5

CONCLUSION AND FUTURE WORK

Three system architectures providing data privacy are introduced based on the state-of-the-art i-vector paradigm in speaker recognition. In the proposed architectures, disclosure of sensitive data is prevented by carrying out numerical comparison operations in a homomorphic encrypted domain. Systems employing these architectures are secure against the honest-but-curious behaviour of any single party of the system, that follows the system protocol, but tries to learn additional information about the data.

Motivated by the architecture proposed in [62], properties of Paillier homomorphic encryption are adapted to cosine similarity comparison of i-vectors, i.e to features of assumed normal distribution. Then, the architecture is extended to a parametric and generative comparison approach, the 2Cov comparator. Finally, a fully encrypted comparison scheme of subject and vendor data is proposed, which requires an additional authentication server in order to perform vendor depending encryption and decryption steps. Also, mean adaptation for the fully encrypted comparison scheme is implemented.

The performance of baseline cosine and 2Cov comparators is preserved, whereas storage and bandwidth requirements are increased due to the high-dimensional encrypted data, serving the purpose of ensured data privacy.

This thesis examines irreversibiliy and unlinkability of ISO/IEC IS 24745 Standard, that are provided in all the systems.

The proposed 2Cov comparator in the encrypted domain becomes the basis for further research, since 2Cov can be seen as a special case of the PLDA (the only difference is in the covariance matrices). Thus, the next steps will be to apply HE properties to the full subspace PLDA model.

Future work may investigate on fully HE for other comparators of the PLDA family.

Also, studies on end-to-end HE for systems employing generative models are relevant, e.g. when considering that feature extractors and comparators may originate from different vendors.

Hash functions are promising for fast searching and indexing of data. Future research might examine how hash value of encrypted reference can be related to hash value of encrypted score, such that the verifier, placed into the authentication server, can use this information to prove the integrity of the score.

BIBLIOGRAPHY

[1] Regulation (EU) 2016/679. *Regulation (EU) 2016/679 of the European Parliament and of the Council 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)*. URL: http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32016R0679.

[2] ISO/IEC 19795-1:Principles and framework. *ISO/IEC 19795-1: 2006 Information technology – Biometric performance testing and reporting – Part 1: Principles and framework*. International Organization for Standardization. 2006.

[3] Directive (EU) 2015/2366. *Directive (EU) 2015/2366 of the European Parliament and of the Council of 25 November 2015 on payment services in the internal market, amending Directives 2002/65/EC, 2009/110/EC and 2013/36/EU and Regulation (EU) No 1093/2010, and repealing Directive 2007/64/EC (Text with EEA relevance)*. URL: http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32015L2366.

[4] ISO/IEC 24745:2011 Biometric information protection. *ISO/IEC 24745:2011 Information technology - Security techniques - Biometric information protection*. International Organization for Standardization. 2011.

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

[6] Jerzy Neyman and Egon S Pearson. „On the problem of the most efficient tests of statistical hypotheses.“ In: *Breakthroughs in statistics*. Springer, 1992, pp. 73–108.

[7] John HL Hansen and Taufiq Hasan. „Speaker recognition by machines and humans: A tutorial review.“ In: *IEEE Signal processing magazine* 32.6 (2015), pp. 74–99.

[8] George R Doddington, Mark A Przybocki, Alvin F Martin, and Douglas A Reynolds. „The NIST speaker recognition evalu ation–overview, methodology, systems, results, perspective.“ In: *Speech Communication* 31.2 (2000), pp. 225–254.

[9] Niko Brümmer. „Measuring, refining and calibrating speaker and language information extracted from speech.“ PhD thesis. University of Stellenbosch, 2010.

[10] Directive 95/46/EC. *Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data*. URL: http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex:31995L0046.

[11] Final Report on Strong Customer Authentication. *Final Report: Draft Regulatory Technical Standards on Strong Customer Authentication and common and secure communication under Article 98 of Directive 2015/2366 (PSD2)*. URL: https://www.eba.europa.eu/documents/10180/1761863/Final+draft+RTS+on+SCA+and+CSC+under+PSD2+%28EBA-RTS-2017-02%29.pdf.

[12] Ronald L Rivest, Adi Shamir, and Leonard Adleman. „A method for obtaining digital signatures and public-key cryptosystems." In: *Communications of the ACM* 21.2 (1978), pp. 120–126.

[13] Carlos Aguilar-Melchor, Simon Fau, Caroline Fontaine, Guy Gogniat, and Renaud Sirdey. „Recent advances in homomorphic encryption: A possible future for signal processing in the encrypted domain." In: *IEEE Signal Processing Magazine* 30.2 (2013), pp. 108–117.

[14] Taher ElGamal. „A public key cryptosystem and a signature scheme based on discrete logarithms." In: *IEEE transactions on information theory* 31.4 (1985), pp. 469–472.

[15] Pascal Paillier et al. „Public-key cryptosystems based on composite degree residuosity classes." In: *Eurocrypt*. Vol. 99. Springer. 1999, pp. 223–238.

[16] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. „Relations among notions of security for public-key encryption schemes." In: *Advances in Cryptology—CRYPTO'98*. Springer. 1998, pp. 26–45.

[17] Pascal Paillier and David Pointcheval. „Efficient public-key cryptosystems provably secure against active adversaries." In: *Asiacrypt*. Vol. 99. Springer. 1999, pp. 165–179.

[18] Nalini K. Ratha, Jonathan H. Connell, and Ruud M. Bolle. „Enhancing security and privacy in biometrics-based authentication systems." In: *IBM systems Journal* 40.3 (2001), pp. 614–634.

[19] Nalini K Ratha, Sharat Chikkerur, Jonathan H Connell, and Ruud M Bolle. „Generating cancelable fingerprint templates." In: *IEEE Transactions on pattern analysis and machine intelligence* 29.4 (2007), pp. 561–572.

[20] Anil K Jain, Karthik Nandakumar, and Abhishek Nagar. „Biometric template security." In: *EURASIP Journal on Advances in Signal Processing* 2008 (2008), p. 113.

[21] Andrew BJ Teoh, Alwyn Goh, and David CL Ngo. „Random multispace quantization as an analytic mechanism for biohashing of biometric and random identity inputs.“ In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.12 (2006), pp. 1892–1901.

[22] Ari Juels and Martin Wattenberg. „A fuzzy commitment scheme.“ In: *Proceedings of the 6th ACM conference on Computer and communications security*. ACM. 1999, pp. 28–36.

[23] Yao-Jen Chang, Wende Zhang, and Tsuhan Chen. „Biometrics-based cryptographic key generation.“ In: *Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on*. Vol. 3. IEEE. 2004, pp. 2203–2206.

[24] Claus Vielhauer, Ralf Steinmetz, and Astrid Mayerhofer. „Biometric hash based on statistical features of online signatures.“ In: *Pattern Recognition, 2002. Proceedings. 16th International Conference on*. Vol. 1. IEEE. 2002, pp. 123–126.

[25] Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. „Fuzzy extractors: How to generate strong keys from biometrics and other noisy data.“ In: *International conference on the theory and applications of cryptographic techniques*. Springer. 2004, pp. 523–540.

[26] Andrew Chi-Chih Yao. „How to generate and exchange secrets.“ In: *Foundations of Computer Science, 1986., 27th Annual Symposium on*. IEEE. 1986, pp. 162–167.

[27] Michael O Rabin. „How To Exchange Secrets with Oblivious Transfer.“ In: *IACR Cryptology ePrint Archive* 2005 (2005), p. 187.

[28] Shuiming Ye, Ying Luo, Jian Zhao, and Sen-ChingS Cheung. „Anonymous biometric access control.“ In: *EURASIP Journal on Information Security* 2009.1 (2009), p. 865259.

[29] Georg M Penn, Gerhard Pötzelsberger, Martin Rohde, and Andreas Uhl. „Customisation of Paillier homomorphic encryption for efficient binary biometric feature vector matching.“ In: *Biometrics Special Interest Group (BIOSIG), 2014 International Conference of the*. IEEE. 2014, pp. 1–6.

[30] Mauro Barni, Tiziano Bianchi, Dario Catalano, Mario Di Raimondo, Ruggero Donida Labati, Pierluigi Failla, Dario Fiore, Riccardo Lazzeretti, Vincenzo Piuri, Alessandro Piva, et al. „A privacy-compliant fingerprint recognition system based on homomorphic encryption and fingercode templates.“ In: *Biometrics: theory applications and systems (BTAS), 2010 Fourth IEEE International Conference on*. IEEE. 2010, pp. 1–7.

[31] Tiziano Bianchi, Stefano Turchi, Alessandro Piva, Ruggero Donida Labati, Vincenzo Piuri, and Fabio Scotti. „Implementing fingercode-based identity matching in the encrypted domain." In: *Biometric Measurements and Systems for Security and Medical Applications (BIOMS), 2010 IEEE Workshop on*. IEEE. 2010, pp. 15–21.

[32] Zekeriya Erkin, Martin Franz, Jorge Guajardo, Stefan Katzenbeisser, Inald Lagendijk, and Tomas Toft. „Privacy-preser-ving face recognition." In: *International Symposium on Privacy Enhancing Technologies Symposium*. Springer. 2009, pp. 235–253.

[33] Margarita Osadchy, Benny Pinkas, Ayman Jarrous, and Boaz Moskovich. „Scifi-a system for secure face identification." In: *Security and Privacy (SP), 2010 IEEE Symposium on*. IEEE. 2010, pp. 239–254.

[34] Julien Bringer, Herve Chabanne, Melanie Favre, Alain Patey, Thomas Schneider, and Michael Zohner. „GSHADE: faster privacy-preserving distance computation and biometric identification." In: *Proceedings of the 2nd ACM workshop on Information hiding and multimedia security*. ACM. 2014, pp. 187–198.

[35] Julien Bringer, Hervé Chabanne, and Alain Patey. „Privacy-preserving biometric identification using secure multiparty computation: An overview and recent trends." In: *IEEE Signal Processing Magazine* 30.2 (2013), pp. 42–52.

[36] Oded Goldreich. *Foundations of cryptography: volume 2, basic applications*. Cambridge university press, 2009.

[37] Matthew Turk and Alex Pentland. „Eigenfaces for recognition." In: *Journal of cognitive neuroscience* 3.1 (1991), pp. 71–86.

[38] Tomi Kinnunen and Haizhou Li. „An overview of text-independent speaker recognition: From features to supervectors." In: *Speech communication* 52.1 (2010), pp. 12–40.

[39] Jongseo Sohn, Nam Soo Kim, and Wonyong Sung. „A statistical model-based voice activity detection." In: *IEEE signal processing letters* 6.1 (1999), pp. 1–3.

[40] Steven Davis and Paul Mermelstein. „Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences." In: *IEEE transactions on acoustics, speech, and signal processing* 28.4 (1980), pp. 357–366.

[41] Sadaoki Furui. „Cepstral analysis technique for automatic speaker verification." In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 29.2 (1981), pp. 254–272.

[42] Jason Pelecanos and Sridha Sridharan. „Feature warping for robust speaker verification." In: (2001).

[43] Bing Xiang, Upendra V Chaudhari, Jiři Navrátil, Ganesh N Ramaswamy, and Ramesh A Gopinath. „Short-time Gaussianization for robust speaker verification.“ In: *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*. Vol. 1. IEEE. 2002, pp. I–681.

[44] Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn. „Speaker verification using adapted Gaussian mixture models.“ In: *Digital signal processing* 10.1-3 (2000), pp. 19–41.

[45] Leonard E Baum and George Sell. „Growth transformations for functions on manifolds.“ In: *Pacific Journal of Mathematics* 27.2 (1968), pp. 211–227.

[46] Najim Dehak, Reda Dehak, Patrick Kenny, Niko Brümmer, Pierre Ouellet, and Pierre Dumouchel. „Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification.“ In: *Tenth Annual conference of the international speech communication association*. 2009.

[47] Douglas A Reynolds and Richard C Rose. „Robust text-independent speaker identification using Gaussian mixture speaker models.“ In: *IEEE transactions on Speech and Audio Processing* 3.1 (1995), pp. 72–83.

[48] Arthur P Dempster, Nan M Laird, and Donald B Rubin. „Maximum likelihood from incomplete data via the EM algorithm.“ In: *Journal of the royal statistical society. Series B (methodological)* (1977), pp. 1–38.

[49] J-L Gauvain and Chin-Hui Lee. „Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains.“ In: *IEEE transactions on speech and audio processing* 2.2 (1994), pp. 291–298.

[50] William M Campbell, Douglas E Sturim, and Douglas A Reynolds. „Support vector machines using GMM supervectors for speaker verification.“ In: *IEEE signal processing letters* 13.5 (2006), pp. 308–311.

[51] Patrick Kenny. „Joint factor analysis of speaker and session variability: Theory and algorithms.“ In: *CRIM, Montreal,(Report) CRIM-06/08-13* 215 (2005).

[52] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. „Front-end factor analysis for speaker verification.“ In: *IEEE Transactions on Audio, Speech, and Language Processing* 19.4 (2011), pp. 788–798.

[53] Andrew O Hatch, Sachin S Kajarekar, and Andreas Stolcke. „Within-class covariance normalization for SVM-based speaker recognition.“ In: *Interspeech*. 2006.

[54] Simon JD Prince and James H Elder. „Probabilistic linear discriminant analysis for inferences about identity.“ In: *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE. 2007, pp. 1–8.

[55] Patrick Kenny. „Bayesian speaker verification with heavy-tailed priors.“ In: *Odyssey*. 2010, p. 14.

[56] Daniel Garcia-Romero and Carol Y Espy-Wilson. „Analysis of i-vector Length Normalization in Speaker Recognition Systems.“ In: *Interspeech*. Vol. 2011. 2011, pp. 249–252.

[57] Simon JD Prince. *Computer vision: models, learning, and inference*. Cambridge University Press, 2012.

[58] Niko Brümmer. *A farewell to SVM: Bayes factor speaker detection in supervector space*. 2006.

[59] Niko Brümmer and Edward De Villiers. „The speaker partitioning problem.“ In: *Odyssey*. 2010, p. 34.

[60] Vasileios Vasilakakis, Pietro Laface, and Sandro Cumani. „Pairwise Discriminative Speaker Verification in the I-Vector Space.“ In: (2013).

[61] IEEE Standards Committee. „754-2008 IEEE standard for floating-point arithmetic.“ In: *IEEE Computer Society Std* 2008 (2008).

[62] Marta Gomez-Barrero, Julian Fierrez, Javier Galbally, Emanuele Maiorana, and Patrizio Campisi. „Implementation of Fixed Length Template Protection Based on Homomorphic Encryption with Application to Signature Biometrics.“ In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2016, pp. 191–198.

[63] Craig S Greenberg, Désiré Bansé, George R Doddington, Daniel Garcia-Romero, John J Godfrey, Tomi Kinnunen, Alvin F Martin, Alan McCree, Mark Przybocki, and Douglas A Reynolds. „The NIST 2014 speaker recognition i-vector machine learning challenge.“ In: *Odyssey: The Speaker and Language Recognition Workshop*. 2014, pp. 224–230.

[64] Anthony Larcher, Kong Aik Lee, and Sylvain Meignier. „An extensible speaker identification sidekit in Python.“ In: *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE. 2016, pp. 5095–5099.

[65] Elaine Barker, Lily Chen, Allen Roginsky, and Miles Smid. „Recommendation for pair-wise key establishment schemes using discrete logarithm cryptography.“ In: *NIST special publication* 800 (2013), 56A.

[66] Andy Adler. „Vulnerabilities in biometric encryption systems." In: *International Conference on Audio-and Video-Based Biometric Person Authentication*. Springer. 2005, pp. 1100–1109.