Security from uncertainty:
Designing privacy-preserving verification methods using Noise

Thesis submitted in partial fulfillment
of the requirements for the degree of

*Master of Science in **Computer Science and Engineering** by Research*

Praguna Manvi
2021701031
<praguna.manvi@research.iiit.ac.in>

International Institute of Information Technology
Hyderabad - 500 032, INDIA
December 2023

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "**Security from uncertainty: Designing privacy-preserving verification methods using Noise**" by Praguna Manvi, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Adviser: Prof. Anoop Namboodiri

आपाद मौलि पर्यन्तं गुरूणां आकृतिं स्मरेत्
तेन विघ्नाः प्रणश्यन्ति सिध्यन्ति च मनोरथाः।

Reflect upon Guru's profound essence before you begin. Through this devotion, impediments shall fade, and aspirations shall blossom into realization

To my family and friends

Acknowledgments

I want to express my profound gratitude to Prof. Anoop Namboodiri for his unwavering support, guidance, and mentorship throughout my thesis journey. I'm thankful for our engaging discussions, his nurturing of my ideas, and his inspiration that consistently pointed me in the right direction. I consider myself truly fortunate to have learned from such a distinguished scholar.

I would also like to extend my sincere appreciation to Prof. Kannan Srinathan, without whom this work would not have been possible. I am grateful for his thought-provoking discussions and his mentorship, which helped me navigate domains previously unknown to me.

My heartfelt thanks go to Achintya, whose constant availability during the initial stages and his insights and guidance in MPCs significantly aided me.

I am also indebted to my dear friends in Basil Lab, including Amrit, Shivali, Gowri, Prateek, Arun, Nithin and Apoorva, who served as pillars of strength during this endeavor. Your encouragement, understanding, and camaraderie have been invaluable. Your unwavering belief in me, even during moments of doubt, kept me motivated and focused. I would also like to extend my gratitude to Saketh sir for his support within the lab and for engaging me in R & D projects.

I also want to acknowledge and thank my friends Sashank, Shubham, Rupak, Mehul, Chirag and many others for their steadfast support, engaging discussions and camaraderie.

Lastly, I express my deep appreciation to my family for their endless love and support. Your belief in me has been the driving force behind my determination to complete this thesis.

In conclusion, I am profoundly grateful to Prof. Anoop Namboodiri, Prof. Kannan Srinathan, Achintya, and my friends for their contributions to this thesis. Your collective guidance, encouragement, and friendship have not only made this academic endeavor possible but also enjoyable. I couldn't have asked for a better support system, and I am honored to have had you all on this journey.

Abstract

Biometric authentication plays an increasingly prominent role in today's products and services for verifying an individual's identity. It is not only efficient but also practical, as it establishes a unique link to an individual through their physical and behavioral characteristics [43]. Unlike conventional authentication mechanisms like passwords or documents, biometric traits are inherent to each individual, eliminating the need to memorize additional information [64]. However, the security and privacy of biometric templates used in authentication remain primary concerns, as biometric data is strongly and irrevocably tied to an individual, as emphasized in the article [42]. In the context of remote authentication, Secure Multiparty Computation (SMC) offers a powerful solution. SMC enables two parties to interactively compute a function using their private inputs without disclosing any information except for the output itself [19]. This approach ensures that biometric template comparison is carried out in a privacy-preserving manner, enhancing both security and privacy in authentication services.

In this thesis, we introduce a unique approach to iris, fingerprint, and face verification by incorporating "noise" into the authentication process. In our work,"noise" refers to signals obtained from non-discriminatory or unreliable regions of biometric characteristics. Our extensive empirical evaluation reveals a correlation among noise features, and we leverage this correlation in a novel Secure Two-Party Computation (STPC) design. This STPC design operates on quantified uncertainty between noise features, providing information-theoretic security. Our approach has low accuracy degradations, practical computational complexity, wide applicability making it suitable for practical real-time applications.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Biometrics

The term "**biometrics**" is derived from two Greek words: "bio," which means "life," and "metrics," which means "measurement" or "to measure." When combined, "biometrics" essentially means "the measurement and analysis of biological data." In the context of technology and security, biometrics refers to the measurement and analysis of unique physical or behavioral characteristics, such as fingerprints, facial features, or voice patterns, for the purpose of identity management.

One of the earliest known official uses of biometrics can be traced back to Babylonian clay tablets dating as far back as 500 BC [77]. These ancient records documented fingerprints for the purpose of conducting business transactions, representing an early instance of biometric data being employed for official and practical purposes. Across the course of human history, individuals have consistently turned to biometric characteristics such as faces or fingerprints as reliable means to identify both familiar and unfamiliar individuals. This seemingly straightforward task, although fundamental, grew in complexity as populations expanded and increased travel introduced new individuals into communities that were once small and tightly-knit. Consequently, the demand for accurate and dependable methods of identification has become increasingly crucial, reflecting the ongoing evolution of biometric recognition throughout civilization.

Not all biological measurements qualify to be a biometric, they must satisfy [46]:

- **Universality**: This principle asserts that every individual should possess the biometric characteristic under consideration. In other words, the trait should be present in all individuals, making it applicable for widespread use in identification.

- **Distinctiveness:** The biometric characteristic should exhibit the quality of distinctiveness, ensuring that any two individuals are significantly different from each other concerning this trait. This distinctiveness is essential to avoid confusion and ensure accurate identification.

- **Permanence:** Permanence implies that the biometric characteristic remains sufficiently consistent and invariant over time with respect to the chosen matching criteria. It should not change significantly over the course of time, allowing for reliable long-term identification.

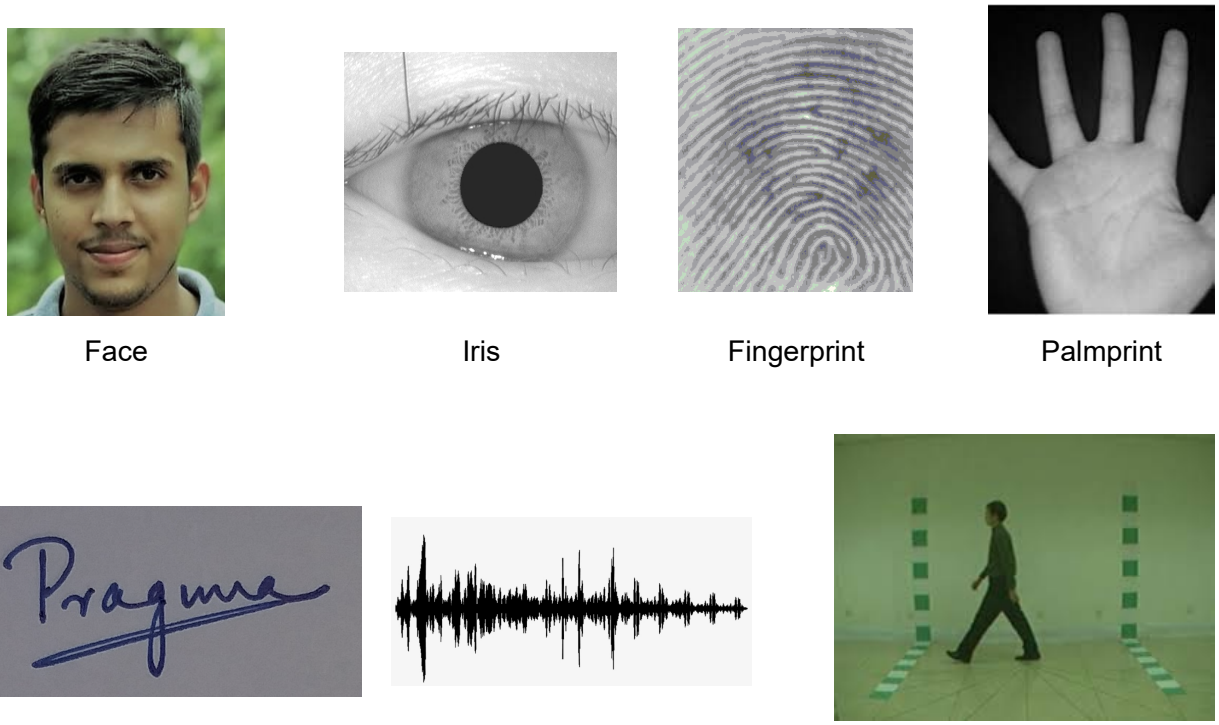

Figure 1.1: Body traits that can be used for biometric recognition. Anatomical traits include face, fingerprint and Iris, palmprint while gait, signature and voice form behavioral traits [2, 58, 63, 73])

- **Collectability:** Collectability relates to the ease with which the biometric characteristic can be quantitatively measured or collected. The characteristic should be readily measurable in a practical and feasible manner for effective use in identification processes.

Along with these principles, a practical biometric system should be accurate and must maintain standards of user acceptability and privacy. Commonly used biometric traits is shown in Figure 1.1. A biometric system is essentially a pattern recognition system designed to authenticate users based on their unique biological or behavioral characteristics.

1.2 Authentication

Biometric authentication which is a primary application of biometrics, has gained widespread adoption in various sensitive digital applications, including payment systems, access control systems, and crime/fraud detection systems, among others [78]. These authentication systems rely on biometric signals, which are unique and highly distinctive, providing a secure link to an individual's identity [68]. Unlike traditional security methods that involve memorizing passwords [80], biometrics offer the advantage of not requiring users to remember complex passwords, enhancing convenience. However, once biometric data is compromised, it

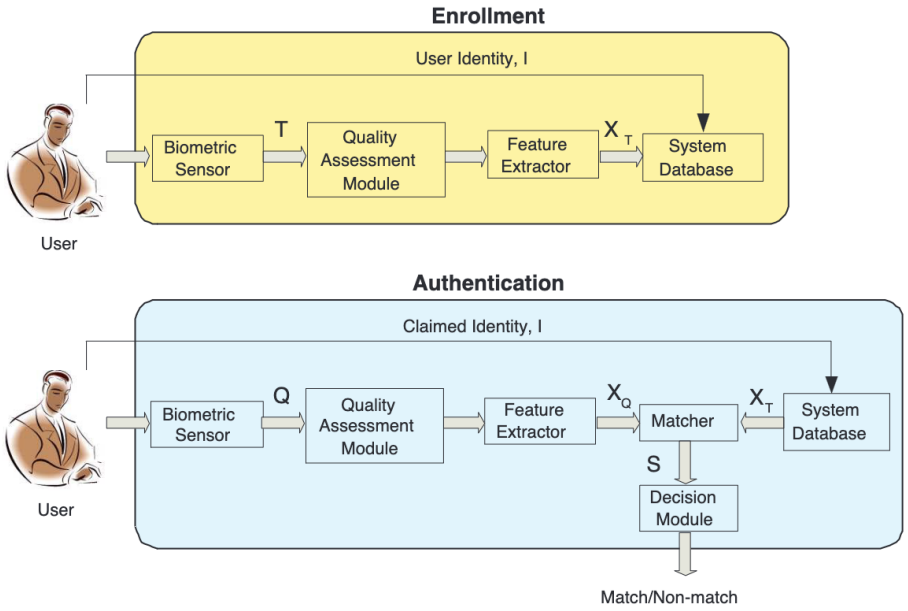

Figure 1.2: A schematic representation of a Biometric Authentication System, with user enrollment and authentication stages [47])

cannot be replaced or reset, and this poses a significant security risk not only for the individual but also for other systems that rely on the same biometric data.

Biometric authentication systems can operate in either verification mode or identification mode, depending on the specific application context [46]:

- **Verification Mode:** In verification mode, the biometric system authenticates an individual's identity by comparing their biometric data to their own stored biometric templates in the system's database. To initiate recognition in this mode, individuals typically provide a some form of personal identification like password. The system then conducts a one-to-one comparison to determine whether the claimed identity is valid. For instance, it checks if the provided biometric data matches the individual associated with it. Verification mode is commonly used for positive recognition, preventing multiple individuals from using the same identity.

- **Identification Mode:** In contrast, identification mode involves the system recognizing an individual by searching the biometric templates of all users in its database for a match. This mode employs a one-to-many comparison, attempting to establish an individual's identity without requiring explicit identity claims. Instead of asking, "Does this biometric data belong to X?" it poses the question, "Whose biometric data is this?" Identification mode is particularly valuable in negative recognition applications, where the system determines whether a person is who they implicitly or explicitly deny being.

In an authentication system, the typical process involves a user enrolling their biometric template, denoted as X_T, by storing it in a database referred to as D. Subsequently, during the online authentication phase, a new template X_Q is extracted, and this template is compared with the stored templates within the database D using a matching module denoted as M. The decision to grant or deny user access is made based on a threshold applied to the comparison function executed by the matching module $M(D \text{ or } X_Q, X_t)$. This thresholding process in the decision module based on score S determines whether the user's biometric template sufficiently matches those in the database, thus authorizing or denying access accordingly. The illustration for the same, adopted from [47] is shown in Figure 1.2.

1.3 Security Threats

Security threats in biometric recognition systems are multifaceted and can be categorized into three main areas [42]. First, at the sensor level, there are concerns related to **presentation attacks**. Second, in the feature extraction module, **adversarial attacks** can pose risks. Finally, at the database and matching modules, issues such as **template theft** and subsequent **template reconstruction attacks** are of concern. These security challenges have been the focus of extensive research in the biometrics community. However, certain aspects of these challenges remain unresolved, particularly regarding their applicability in detecting new types of attacks and adapting to new sensors that were not part of their original training. We summarize the three types of threats below:

- **Presentation Attacks (PAs):** Presentation attacks can take on two primary forms. Firstly, they can be used as obfuscation attacks, where the attacker aims to conceal their own identity. Secondly, they can be utilized as impersonation attacks, where the attacker seeks to imitate someone else's identity. This requires active participation of the attacker. Presentation attack detection (PAD) system, which are either hardware or software based are developed for mitigation.

- **Adverserial Attacks :** Adversarial attacks pose a significant challenge to the security and robustness of machine learning models, including those used in biometric recognition systems. These attacks involve making small, carefully crafted modifications to input data, such as images or sensor readings, with the goal of causing the model to make incorrect predictions or classifications. One of the key characteristics of adversarial attacks is their imperceptibility to the human eye, as the changes are often subtle and designed to evade human detection. Adversarial attacks can be categorized into white-box attacks, where the attacker has full knowledge of the model, and black-box attacks, where the attacker has limited information about the model's inner workings. In contrast to PAD's these attacks dom't require online participation of the attacker. Adverserial training strategies are employed for mitigation.

- **Template Attacks :** Numerous studies have raised concerns about the security of biometric recognition systems, demonstrating that templates extracted from these systems, including deep face representations, minutiae-based fingerprint representations, and iriscode features, can be inverted back into the image space with remarkable fidelity, as illustrated in Figure 1.3. Furthermore, research has revealed that

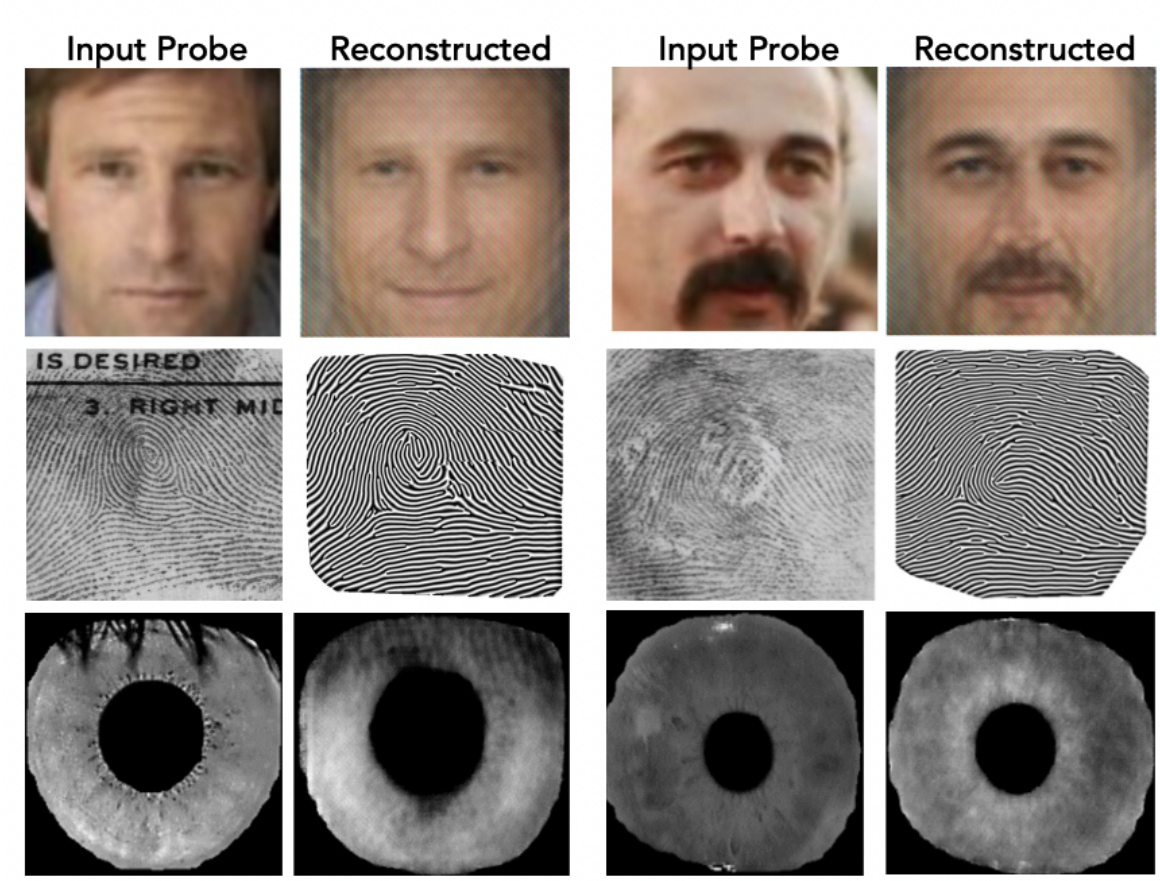

Figure 1.3: Template attacks in cases where reconstruction could successfully match the input probe, shown for iris, fingerprint and face modalities respectively [42])

certain ”soft” demographic attributes, such as age and gender, are inadvertently encoded into these biometric templates.

This poses a serious security risk, particularly in light of reported breaches of databases containing biometric templates. It's important to note that a biometric template can be compromised at various points in the system, either immediately after feature extraction when it resides in the enrollment database or even during the matching routine if the template must be decrypted for comparison. Therefore, it is imperative that biometric recognition systems employ robust encryption and security measures to safeguard templates from potential hackers throughout the entire authentication process.

In this thesis, we present a novel solution to counter template attacks by introducing a privacy-preserving approach. Within this approach, we perform the computations of the matching module within an encrypted domain, thereby effectively thwarting reconstruction attacks. Our approach is built upon provable cryptographic foundations, ensuring robust protection against potential security threats from template attacks.

1.4 Privacy Preservation in Biometrics

The statement made in [27], "The problems of privacy and authentication are closely related, and techniques for solving one can frequently be applied to the other," underscores the interconnected nature of privacy and authentication issues. Solutions developed for addressing one of these challenges often find applicability in tackling the other.

Cryptography, involves the application of data transformations with the primary objectives of rendering the data indecipherable to adversaries. These transformations serve as effective solutions to two fundamental challenges in data security: the privacy problem, which aims to thwart adversaries from extracting information from a communication channel, and the authentication problem, which seeks to prevent adversaries from introducing counterfeit data into the channel or modifying messages to alter their intended meaning.

In Chapter 2, we extensively explore different techniques for template protection, categorizing them into cancellable templates, cryptosystems, and computation in the encrypted domain. The research presented in this thesis primarily falls into the latter category. We introduce a novel privacy-preserving secure two-party computation protocol [32] designed for performing verification between two parties using Noise.

1.5 Noise is useful

Noise, in a broad sense, refers to undesired signals or inputs that are irrelevant or unwanted for a specific task. In the context of our work on biometric recognition, we specifically refer to noise signals as any signals that are considered unnecessary or uninformative. These signals may arise from regions that are non-discriminative or unrelated to the trait being verified. We refer to the binarized representations from noise signals as **"noise codes"**.

In our approach, we don't entirely discard noise signals from regions of a trait as completely useless. Instead, we recognize that there may still be a weak correlation or connection between the noise codes originating from these regions. We take advantage of this weak correlation by quantifying the uncertainty in bit mismatches and incorporating it into the design of the secure protocol's AND gate (we support with empirical findings in chapters 4 3). This approach allows us to achieve the AND operation with only one round of communication during the online phase of the protocol, enhancing its efficiency and security.

We provide a visual representation of this concept's implementation using noise in Figure 1.4. Within this illustration, HD denotes the Hamming distance, serving as a measure of the mismatch level by utilizing sampled 1-out-of-4 mismatch 4-bit pairs (n_p, m_p) from corresponding positions of the noise codes of each participating party (P_1, P_2). Furthermore, we introduce sub-protocols and utilities specifically designed to generate these pairs from noise, ensuring their compliance with the specified assumption. This framework serves as the core of our approach, enabling the secure computation of complex functions while upholding privacy and security through the integration of Noise.

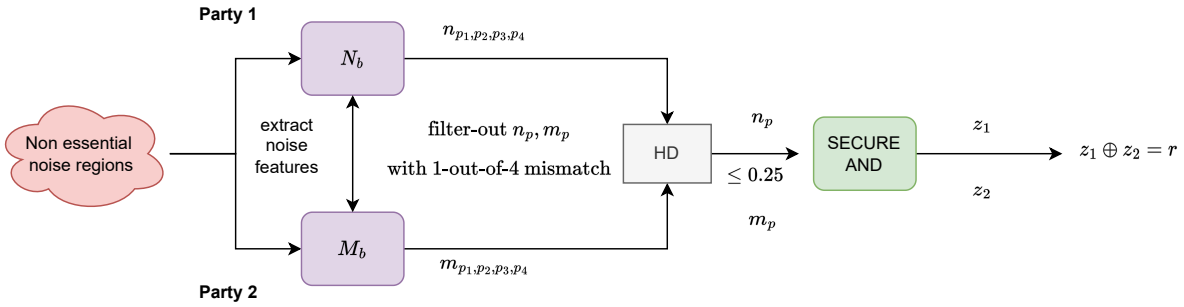

Figure 1.4: Using noise-code mismatch uncertainty for Secure AND

1.6 Privacy Preservation using Noise

With the expectation of the 1-out-of-4 mismatch property in the 4-bit pairs obtained from the noise codes, as discussed in the previous section, we establish a security definition. This definition is rooted in the concept that we perform binary hashing of binary sensitive inputs using random relations derived from noise codes, effectively serving as one-time pads for each AND gate computation. This approach ensures **information-theoretic security** of sensitive inputs.

Considering distributed shares with Party P_1 having x_1 and y_1, and Party P_2 having x_2 and y_2, along with random 4-bit pairs represented as octets $m = \langle m_1, m_2, m_3, m_4 \rangle$ and $n = \langle n_1, n_2, n_3, n_4 \rangle$, we perform the following value interchange for the final computation of z_1 at Party P_1:

- $e_1 = x_1 \oplus m_2 \oplus m_4$
- $e_2 = y_1 \oplus m_3 \oplus m_4$

We can note that e_1, e_2 does not reveal any sensitive information about x_1 or y_1 which are protected through noise at P_2. Similar encrypted values are received at P_1. We provide a formal proof alluding to the complete view of the protocol in the appendix.

1.6.1 Security Assumptions

Privacy is guaranteed in this protocol under the assumption of a semi-honest, passive adversary. A passive adversary adheres to the protocol and observes the information exchanged during the protocol execution. We also assume security against adversaries with unbounded computational power, which distinguishes this protocol from other methods that rely on computational security and key size constraints. In our work, we do not rely on any public-private key cryptography assumptions concerning the parties involved before executing the protocol. Furthermore, the construction of octets in our approach is not limited by encryption assumptions, as is the case with Oblivious Transfers (OTs). We, however assume each octet and key in our work is used only once, since they rely on OTP's security.

1.6.2 Security Compliance

ISO/IEC 24745:2022, as detailed in the IEEE standard [1], sets forth specific principles for crafting template protection schemes. It emphasizes the importance of safeguarding biometric references to meet diverse requirements for secrecy, irreversibility, and renewability during both the storage and transfer phases. The methods proposed in our work align with these principles, as they ensure secrecy, irreversibility, and renewability in both the storage and transfer of templates. This is achieved by storing encrypted templates and conducting operations on them while preserving privacy.

1.7 Motivation

Our motivation stems from the novel approach we introduce for verification, offers a unique approach without traditional cryptographic assumptions between involved parties. Instead, we harness seemingly unreliable regions and leverage inherent randomness to establish security. This unconventional strategy enables the design of a protocol that achieves both robust security and practical utility. Through theoretical analysis and performance verification on publicly available datasets, we demonstrate the potential feasibility of our approach for remote verification solutions across various biometric modalities, including Iris, Fingerprint, and Face verification.

1.8 Decoding the Thesis's Title

The inclusion of the term "uncertainty" in the title underscores our emphasis on leveraging the inherent randomness within noise regions, which is quantified probabilistically through the 1-out-of-4 mismatch assumption. We substantiate these findings empirically in chapters 3 and 4 through an examination of noise distributions. This element of uncertainty is pivotal in our novel privacy-preserving solution, ensuring both privacy and verification.

1.9 Contributions

In this section, we summarize the key contributions of this thesis:

- **Novel Secure Multi-Party Solution for Verification using Noise:** We have introduced a novel secure multi-party computation protocol for verification that leverages the concept of noise. This protocol ensures both privacy and security in the verification process.

- **Application of Noise-Based Verification to Iris, Face, and Fingerprint Modalities:** Our approach is versatile and can be applied to various biometric modalities, including Iris, Face, and Fingerprint recognition, making it applicable to a wide range of authentication scenarios.

- **Design of Noise Feature Extraction:** We have designed noise-based feature extraction methods that adapt to different conditions, enhancing the robustness and effectiveness of the base protocol introduced in Chapter 3.
- **Framework Extension:** We have extended the base protocol introduced in Chapter 3 into a comprehensive framework that can accommodate various use cases and scenarios.
- **Complexity, Security, and Verification Performance Analysis:** Through rigorous analysis, we have evaluated the complexity, security, and verification performance of our proposed method, demonstrating its effectiveness and efficiency.
- **Verification of Noise Properties:** We have conducted empirical studies on noise distributions to verify the properties of noise, providing empirical evidence to support our approach.
- **Comparison and Implementation Details:** We have compared our approach with existing methods and provided implementation insights, including time-cost analysis, to showcase the practicality of our method.

1.10 Thesis Organisation

The thesis is organized as follows :

- **Chapter 2** Summarizes the literature and referred works related to the proposed methods.
- **Chapter 3** Introduces Secure verification scheme for Iris.
- **Chapter 4** Extends the Chapter 3 to other modalities with additional utilities.
- **Chapter 5** Thesis is summarized here.
- **Appendix A** Includes formal security and correctness proofs.

Chapter 2

Literature Review

2.1 Introduction

In this chapter, we offer an overview of prior research relevant to our proposed methods in the upcoming chapters. Our work primarily relies on biometric feature extraction techniques and Secure Multiparty Computation (SMC) foundations. We briefly touch on methods and techniques in these areas. Additionally, as our proposed methods pertain to biometric template protection and privacy preservation, we provide a brief overview of existing approaches in this domain.

2.2 Feature Extraction in Biometrics

In a biometric system, the feature extraction module plays a crucial role in capturing a representation of the discriminatory information, often referred to as "features / descriptors", acquired by the biometric sensor module [43]. This representation is typically structured as a high-dimensional vector with values that can be continuous, discrete, or binary, and it serves as input to an authentication module.

Given the extensive literature on recognition and biometric representational learning, our review in this work focuses on methods directly employed, encompassing both traditional and deep learning approaches.

2.2.1 Tradtional Approaches

In this section, we outline non-deep learning approaches that are commonly employed in authentication schemes for feature extraction within the iris, face, and fingerprint modalities.

- **Iris Codes using Phase based Method [24]:** Each individual iris pattern is isolated using integro-differential operator identifying the iris regions, the next step involves demodulation to extract its phase information. This demodulation process is carried out using quadrature 2-D Gabor wavelets. Essentially, it entails quantizing the phase of the iris pattern on a patch-by-patch basis. This quantization is achieved by determining the quadrant in the complex plane where each resulting phasor lies when a

specific region of the iris is projected onto complex-valued 2-D Gabor wavelets:

$$h_{re,im} = sgn_{Re,Im} \int_{\rho} \int_{\phi} I(\rho, \phi) e^{-i\omega(\theta_0 - \phi)} . e^{-(r_0 - \rho)^2/\alpha^2} . e^{-(\theta_0 - \phi)^2/\beta^2} . \rho d\rho d\phi \tag{2.1}$$

where $I(\rho, \phi)$ is the iris image I transformed to polor coordinate system, α, β are multiscale 2-D wavelet size parameters, spanning an eight-fold range, ω is the wavelet frequency inversely proportional to β. (r_0, θ_0) are the polar coordinates of iris for which phase coordinates are computes as $h_{re,im}$ either 0 or 1 based on the sgn. A total of 2048 such phase bits are computed. Hamming distance is used for comparison.

- **Fingerprint Minutiae & Filterbank descriptors [56] [39]:** Fingerprint minutiae points play a critical role in recognition as they are highly discriminative and pivotal. In the recognition process, a ridge or bifurcation pattern is localized and extracted, and then transformed into polar coordinates after aligning the fingerprint. In a straightforward matching approach, the system computes pairwise distances between each pair of polar minutiae coordinates. The number of matching minutiae is then subjected to a threshold for verification. A higher number of minutiae matches indicates a higher degree of similarity in the biometric data, contributing to a more confident verification process.

  To expedite the authentication process, many schemes opt for FingerCode, a binary representation of fingerprint data that facilitates matching using euclidean distance. The fingerprint images are divided into sectors starting from a reference point. Each sector undergoes a series of processing steps, including normalization and filtering with a bank of Gabor filters. The resulting representations are then compared using the Euclidean distance, streamlining the matching process and reducing the computational load associated with minutiae extraction and matching.

- **Eigenfaces for face recognition [74]:** Eigenfaces is a facial recognition technique using Principal Component Analysis (PCA). It represents faces as linear combinations of eigenfaces, which capture primary image variations. The process involves finding a mean face, centering images, computing a covariance matrix, and performing eigenvalue decomposition to represent a face image as a combination of eigenfaces. This reduced-dimensional representation facilitates efficient facial recognition. A normalized face is projected using eigenfaces, with a subset of eigenvectors and euclidean distance is employed in finding the closest match.

2.2.2 Deep Learning Based Methods

In recent years, deep learning methods have witnessed widespread adoption due to improved hardware availability and their superior accuracy in various recognition tasks [61]. Typically, these methods involve extracting deep representations from each image, followed by comparisons using cosine distance. In the following section, we will review the methods employed for fingerprint and face feature extraction in our work.

- **Deep Fingerprint Representation [30]**: DeepPrint is a deep neural network designed to learn and extract fixed-length fingerprint representations of only 200 bytes. DeepPrint is notable for its integration

of domain knowledge specific to fingerprints, including alignment and minutiae detection, directly into the deep network architecture. This incorporation enhances the discriminative power of the representation it generates.

The compact DeepPrint representation offers several advantages over prevalent variable-length minutiae representations. First, it eliminates the need for computationally expensive graph matching techniques. Second, it is more amenable to robust security measures, such as homomorphic encryption. Third, DeepPrint excels in cases involving poor-quality fingerprints where minutiae extraction can be unreliable.

- **Deep Face Representation [70]**: FaceNet is a deep learning model designed to efficiently tackle large-scale face verification and recognition tasks. It learns to map face images directly into a compact Euclidean space, where the distances between points in this space directly represent measures of face similarity. This allows for straightforward implementation of tasks like face recognition, verification, and clustering using FaceNet embeddings as feature vectors.

  Unlike previous deep learning approaches that rely on intermediate bottleneck layers, FaceNet utilizes a deep convolutional network that directly optimizes these embeddings. Training is conducted using triplets of matching and non-matching face patches generated through an online triplet mining method. The key advantage of this approach is significant representational efficiency.

  While other efficient methods are available in the literature, we note that the results achieved with this method are comparable on small verification datasets. Moreover, this particular method has wide adoption in practical applications. Therefore, for the purpose of demonstrating the efficiency of the security protocols, we adhere to this method.

2.3 Biometric Template Protection

2.4 Taxonomy

In Figure 2.1, you can see the taxonomy of existing schemes in the Biometric Template Protection (BTP) domain. Traditionally, biometric template protection schemes [75], [15] have been classified into two categories: cancellables and cryptosystems. However, in recent years, a new category has emerged with comparisons conducted in encrypted domains. In this taxonomy, we've combined Homomorphic Encryption (H.E) and H.E + GC (a hybrid approach with garbled circuits) and introduced Secure Multiparty Computation (SMC), which is the category that the current work belongs to. Below, we provide an brief overview of these methods.

- **Cancellable Biometrics** : Cancelable Biometrics (CB) involve the deliberate and reproducible alteration of biometric signals using specific transforms. These transformations enable the comparison of biometric templates within the modified domain [59]. The Figure 2.2 provides an overview of the same. The cancelable biometric recognition process consists of two phases: Enrollment and

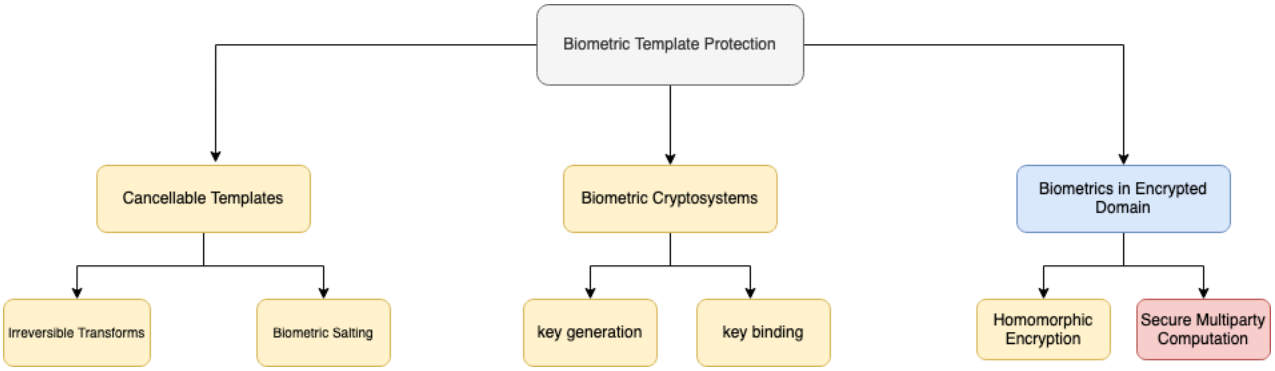

Figure 2.1: Biometric Template Protection Taxonomy (Adapted from [35])

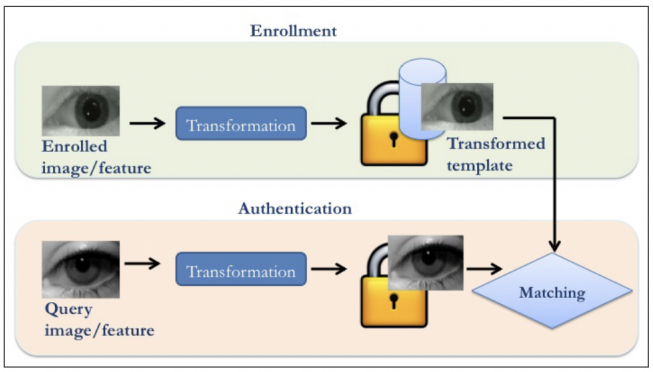

Figure 2.2: Cancellable Biometric System [66])

Authentication (Figure 2.2). During Enrollment, the user's biometric data is collected and features are extracted. A cancelable biometric template is then generated using techniques like Hashing, Filtering, or Cryptography, and stored securely. In the Authentication phase, a similar cancelable biometric template is generated. The probe data is matched against stored templates in the database, enabling user verification or identification. This approach ensures secure biometric recognition and data privacy.

- **Biometric Cryptosystems [49]** : A Biometric Cryptosystem (BCS) is formally defined as a framework supporting techniques for securing biometric data using a key to create a biometric template. This key can be either bound to or generated from the biometric data. The BCS comprises three phases: Enrollment, where data is captured, features are extracted, and a reference template is created; Authentication, which matches biometric data with the reference; and Encryption/Decryption, utilizing keys generated independently, monolithically bound, or derived from the biometric representation for secure data access. BCS ensures privacy and security while using biometrics for authentication and data protection. These phases enable secure processing and storage of biometric data, safeguarding sensitive information throughout the authentication and encryption process. The key generation process used in these methods is shown in Figure 2.3.

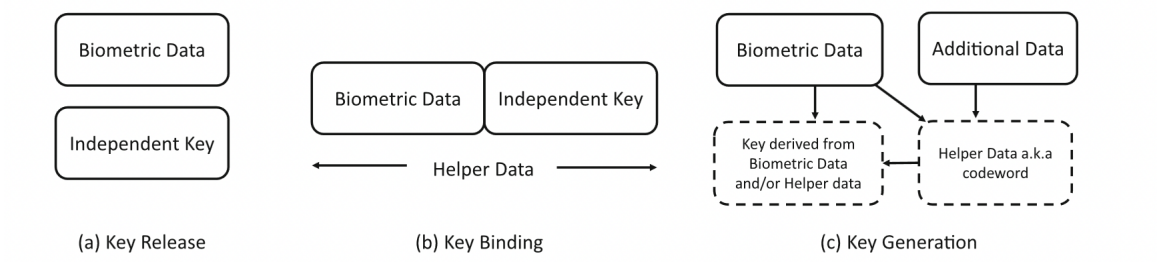

Figure 2.3: Key construction mechanism in Biometric Cryptosystems [49])

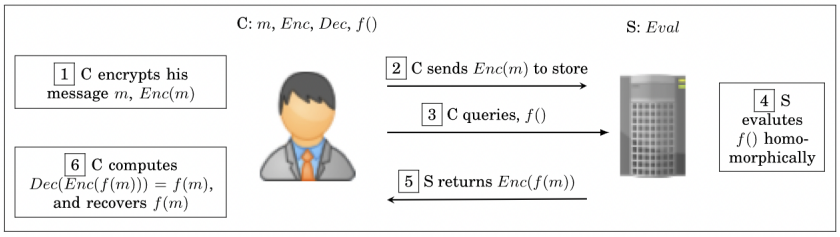

Figure 2.4: A simple illustration of Homomorphic Encryption Schemes [3])

- **Homomorphic Encryption** : Homomorphic Encryption (HE) is an encryption scheme that enables a third party, such as a cloud service provider, to perform specific computational functions on encrypted data while maintaining the properties of the function and the format of the encrypted data. Essentially, HE corresponds to a mathematical mapping within abstract algebra. For instance, in an additively homomorphic encryption scheme, one can obtain E(m1 + m2) from E(m1) and E(m2) for sample messages m1 and m2, without needing to know the explicit values of m1 and m2. Here, E represents the encryption function. H.E is further classified as fully, partially and somewhat homomorphic based on the extent to which homomorphic properties are supported.

  A basic example of Homomorphic Encryption (HE) [3] in a cloud application is depicted in Figure 2.4. In this scenario, the client (C) starts by encrypting their private data (Step 1) and then transmits this encrypted data to the cloud servers (S) (Step 2). When the client needs to perform a specific function (e.g., query), denoted as f(), on their data, they send this function to the server (Step 3). The server executes a homomorphic operation over the encrypted data using the Eval function, essentially calculating f() while keeping the data encrypted (Step 4). Subsequently, the server sends the encrypted result back to the client (Step 5). Finally, the client decrypts the data using their secret key and obtains f(m) (Step 6). This simple example illustrates that the homomorphic operation, Eval(), conducted at the server, doesn't require the client's private key and permits various operations, such as addition and multiplication, on the encrypted client data.

- **Secure Multiparty Computation (SMC)**: Secure Multi-Party Computation (SMC) is a fundamental cryptographic technique that allows collaborative computation while preserving privacy. Within the realm of cryptography, SMC is a crucial research area that addresses secure cooperative computation

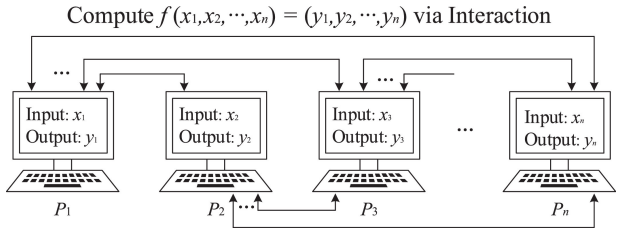

Figure 2.5: Schematic representation of SMC schemes [83])

among multiple participants, all of whom possess private data [83]. In simple terms, in an SMC scenario, two or more parties with private inputs aim to jointly compute a specific function without revealing their inputs to each other. Each participant in the computation receives their intended output without disclosing additional information. This concept extends to various cryptographic tasks, including encryption, authentication, zero-knowledge proofs, commitment schemes, oblivious transfer, and more, within distributed computing environments.

For interactive protocols, SMC establishes a provable security facilitating development of many modern security solutions. Schematically as shown in Figure [83]. Parties $P_i, ...P_n$ compute $f(x_i, ..., x_n)$ as $y_1...y_n$ in a distributed setting only knowing x_i and y_i at P_i and nothing more.

Traditional methods of template protection are known to significantly degrade performance compared to approaches that involve computation over encrypted input. Moreover, these traditional methods primarily focus on safeguarding data during storage and are shown to be vulnerable for various attacks. On the other hand, Homomorphic Encryption (H.E) methods exhibit an asymmetric control mechanism, permitting decryption on one side of the communication. In contrast, Secure Multi-Party Computation (SMC) approaches function interactively and offer provable proofs for information-theoretic security. This makes them resilient against both passive/malicious adversaries depending on the approach, enhancing security in various scenarios.

2.4.1 SMC based Iris Verification

Luo et al. [57] introduced an innovative privacy-preserving iris-code matching protocol leveraging garbled circuits to evaluate Secure Function Encryption (SFE). This novel approach effectively implements the iris code matching process, combining hamming distance with iris masks to eliminate erroneous code segments. The garbled circuit operates on a shared mask, generated from pre-aligned masks in the CASIA iris database [73], ensuring both parties are privy to only the final result. A hybrid method, as proposed by Blanton et al. [10], combines homomorphic computation and garbled circuits for biometric identification, optimizing the process by precomputing most operations. Further enhancements come from optimized multiplication protocols and the use of the DGK scheme [20] for comparison computation. Droandi et al. [6] introduced a multi-biometric authentication protocol based on the SPDZ tool [21], disclosing only the final binary decision. Pia et al. [7] introduced post-quantum Secure Multiparty Computation (SMC) for verification and identification, akin to the Secure XOR phase in this work. Bringer et al. [14] proposed a filtering technique

based on Hamming distances and SMC for iriscode representations, improving performance with a slight trade-off in False Non-Match Rate (FNMR).

This work, in contrast, utilizes the original mask vectors and is grounded in the noise ratio of random 4-bit sequences selected by the parties from noise codes/vectors. This protocol securely eliminates mask regions from templates without the need for key sharing. Operating within a semi-honest security model, our approach omits multiplication or division circuits, significantly reducing complexity. The Offline/Preprocessing Phase exchanges noise code positions without repetitions, safeguarding the protocol from adversaries attempting to guess the vector. We demonstrate the protocol's efficacy in achieving high biometric performance under the assumption of a noise ratio. The protocol features a parallelizable counter, offering substantial performance improvements through batching and combining smaller counters.

2.4.2 SMC based Fingerprint and Face Verification

Sadeghi et al. [69] introduced a pioneering hybrid protocol for privacy-preserving recognition, combining Eigenfaces with garbled circuits for minimum finding and homomorphic encryption for other operations. Building on this foundation, Huang et al. [31] enhanced communication cost efficiency by implementing packing and vertical partitioning in the same hybrid protocol, applied to fingerprint filterbank representations [45]. They employed Hamming distance for comparison in the SCiFI algorithm [65] to achieve privacy-preserving face identification.

Blanton et al. [11] put forth protocols for fingerprint and iris authentication. Fingerprint verification utilized garbled circuits for comparison and homomorphic encryption for Euclidean distance calculations on FingerCodes [44]. Bringer et al. [13] introduced GSHADE, an extension of the SHADE protocol (citeshade), enabling privacy-preserving computation of various metrics while operating on binary vectors.

In the realm of "Verification using noise," Chapter 3 presents SIAN, which leverages mismatches in noise codes to securely compute AND gates for Hamming distance calculations between iris codes. This concept has been adapted to two other biometric modalities, namely face and fingerprint recognition, within a framework designed to compute scalar products. This approach is computationally lightweight, with octets precomputed during the preprocessing phase, eliminating the need for oblivious transfer. During online comparison, only 2 bits are exchanged per Secure AND gate by each party.

Chapter 3

Secure Iris Verification using Noise

3.1 Introduction

This chapter introduces an novel technique that leverages biometric noise to enhance security in the context of iris verification. The method is specifically applied to iris modality due to its binary feature extraction and comparison using normalized Hamming distance, making it an ideal candidate for the proposed privacy-preserving approach.

Iris templates serve as a permanent and stable identifier for individuals, as the human iris rarely undergoes significant alterations over time. This permanence underscores the importance of privacy-preserving approaches when storing or transmitting biometric data for use in biometric services. The protocol proposed in this system addresses these concerns by providing information-theoretic security, particularly under the assumption of a passive adversary. Further details on the security analysis will be explored in the subsequent sections.

To achieve privacy preservation, the method leverages Secure Multiparty Computation (SMC). SMC deals with the computation of a function on inputs in a distributed setting where each party holds an input. SMC ensures several key properties, including the independence of inputs, computational correctness, and the guarantee that no more information is revealed to any party beyond what can be inferred from their inputs and the outputs of the computation [28]. This approach forms the foundation for the privacy-preserving protocol employed in the system, ensuring that sensitive biometric data remains confidential and secure during processing and computation.

Building upon the discussions presented in the preceding chapters, our approach adopts the Secure AND design under multi-party orderless channel asynchrony and adapts it for biometric applications. We recognize the inherent uncertainty associated with the biometric regions that are useful for verification. This recognition underscores the suitability of applying the Secure AND gate, which necessitates minimal communication, to address this uncertainty effectively.

To harness the power of Secure AND, we have designed preprocessing and distillation phases within the protocol. These phases are responsible for generating security octets, which play a crucial role in safeguarding the distributed inputs of Secure Multi-Party Computation (SMC) while performing the secure

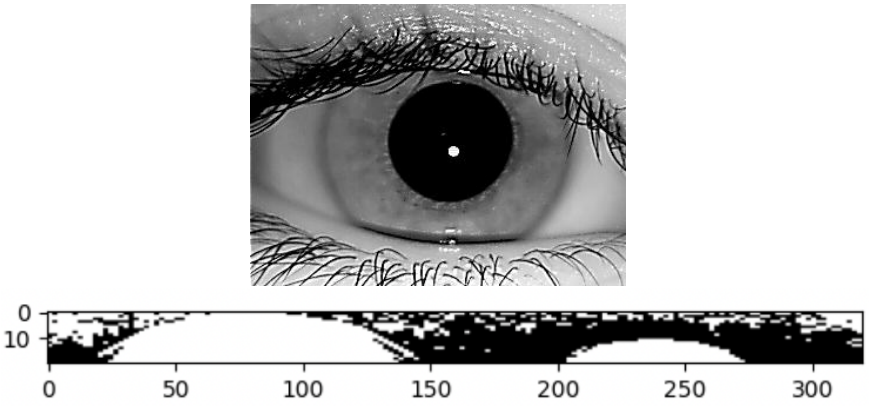

Figure 3.1: Noisy regions extracted from an Iris image

mated	non-mated
0.2144	0.3003

Table 3.1: Average Hamming distance for noise across datasets

AND computation. This approach allows us to achieve the desired privacy and security goals in the context of biometric data processing, ensuring that sensitive information remains protected throughout the computation.

3.1.1 Core Observation

In the process of biometric registration, the signals extracted may contain noise introduced due to imperfect imaging conditions, ambient factors, and the user's interaction with the sensor [40]. This noise typically originates from the areas surrounding the biometric trait, such as the pupils and eyelashes. A binary mask vector is used to encode these noisy or uncertain regions in the image. This mask vector is employed in removing noise during post-processing, which occurs after feature extraction. Figure 3.1 provides a visualization of a mask vector obtained for an iris image, highlighting regions like the pupil and eyelashes.

However, it's important to note that these masks can also encode certain soft biometric traits of an individual, especially when noise is captured from the regions surrounding the biometric trait [41]. While these traits are not distinctive or permanent, they do provide some evidence about the user's identity [12]. This means that signals encoded in the mask regions (the noise code) from the same individual can be more similar to each other than to those from different individuals [53].

From Table 3.1, we can observe that the average Hamming distance between the noise codes (representing noise ratios) across three datasets used in experimentation is lower for mated pairs (less than 25%) than for non-mated pairs. This observation suggests that the noise ratio from the same individual (mated pairs) will be lower than that from different individuals (non-mated pairs). Specifically, any random 4-bit subset from the noise codes of non-mated pairs has an average noise ratio of 50% (corresponding to a 2-bit mismatch), as there is an equal probability of the noise ratio falling anywhere from 0% to 100%. In contrast, the average

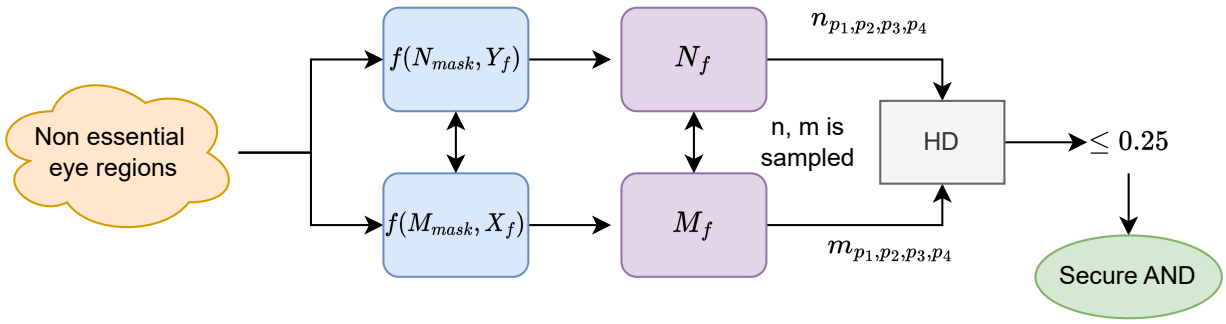

Figure 3.2: High level overview into octet creation and working of secure AND using Noise

noise ratio for mated pairs ranges from 0% to 25% (corresponding to 0-1 bit mismatch), as their noise codes are more similar than those of non-mated pairs.

These empirical observations are demonstrated on three well-known iris datasets, and they are incorporated into the design of the protocol. The protocol is designed to expect a certain degree of noise and assumes that two templates cannot register the same noise. Additionally, noise is utilized as an ancillary feature, with templates being rejected when their noise ratios fall outside the range of (0%, 50%), with a higher probability for higher noise ratios. Thus, the binary mask vectors, which are typically discarded, play an integral role in the proposed method.

Figure 3.2 provides an illustration of the key concept underlying the proposed method, which involves the use of noise (Note: HD is the Hamming distance, indicating mismatch in n, m). Formally, if N_{mask}, M_{mask} are mask vectors $N_f = Y_f[M_{mask}], M_f = X_f[N_{mask}]$ form the noise-codes which are the unreliable values within the iris-codes. The security octets are extracted from these regions as $n_{p_1,p_2,p_3,p_4}, m_{p_1,p_2,p_3,p_4}$, as illustrated in the figure.

Noise codes are grouped as octets, forming four pairs of distributed bits at corresponding noise positions. The central idea here is that if an entire noise code is similar to another one, then a random four-bit selection is likely to have only a 0-1-bit mismatch, as elaborated upon in the preceding paragraphs. This observation serves as the foundation for designing a distillation mechanism that selectively elects octets with a 1-bit mismatch, which would comprise the majority of 3-bit mismatch octets. The Secure AND gate within the protocol only functions if this condition of a 1-bit difference in the octet is met. Further details regarding this mechanism will be explored in subsequent sections, providing a comprehensive understanding of its operation.

3.1.2 Vulnerabilities of Iris Templates

Iris templates, especially when stored in plain text, can be vulnerable to reconstruction attacks, where an attacker could potentially reconstruct the original iris images from the binary templates. Given that iris recognition is considered a stable biometric, the security risks associated with such reconstructions can be significant. Therefore, safeguarding the binary codes commonly used to represent iris features is of paramount importance for biometric applications to function securely.

The proposed system addresses this challenge by providing protection for the templates, both during storage and during data transfer. Importantly, this protection is achieved using information-theoretic security measures. The system operates within a two-party framework, ensuring that neither the server nor the client possesses knowledge of each other's templates during the comparison process. This approach guarantees a high level of security and privacy for the iris biometric data, making it suitable for applications that require robust protection of sensitive biometric information.

3.1.3 Injecting Security using Noise

The octets that are extracted from noise, which are the corresponding values in noise positions, a combination of itself in each party through positional XORs, are employed using binary hash on sensitive inputs and in the relations used in computing secure AND, thereby proving using in providing security, through inherent useful randomness in noise.

Each octet is used only once on plain-text sensitive information; hence, more noise that generates more octets proves more beneficial in the proposed method.

3.1.4 Motivation

Our motivation is rooted in the fact that the method we propose for verification doesn't necessitate traditional cryptographic assumptions between the parties involved. Instead, it capitalizes on what may appear to be unreliable regions in the iris feature extraction process to enhance security. This unconventional approach allows us to design a protocol that offers both robust security and practical utility.

Theoretical analysis, combined with verification performance on public datasets, demonstrates the potential practicality of our approach for remote iris verification solutions. By challenging the need for traditional cryptographic assumptions and leveraging the security-enhancing aspects of iris feature extraction, we aim to provide a secure and effective method for iris biometric verification in remote scenarios.

3.2 Proposed Scheme

In this chapter, we introduce SIAN (Secure Iris Authentication using Noise), which is a secure authentication protocol. SIAN leverages secure multi-party computation, utilizes biometric noise for enhanced security, and operates on fixed-size templates between two parties. The overall scheme used in this work is depicted in Figure 3.3. Here is an overview of the key steps in the protocol:

- **Feature Extraction:** The feature extractor module generates iris codes from images. These iris codes are used for subsequent authentication.
- **Privacy-Preserving Hamming Distance:** The Hamming distance between iris codes is computed in a privacy-preserving manner using the protocol. This ensures that sensitive biometric information remains secure during the comparison process.

- **User Enrollment:** The user enrolls the extracted iris code and mask as (Y_f, Y_{mask}) on the server. These templates are represented as (Y, N), where $Y = Y_f \oplus R$ and $N = N_f \oplus R$. Here, N_f is a noise vector containing values from uncertain regions marked by Y_{mask}, and R is a random bit vector of the same size as Y_f, generated by the client. This additional step enhances the security of data storage on the server side.
- **Protocol Execution:** The proposed protocol operates using these encrypted feature inputs (X, M) and (Y, N), ensuring secure and privacy-preserving authentication.

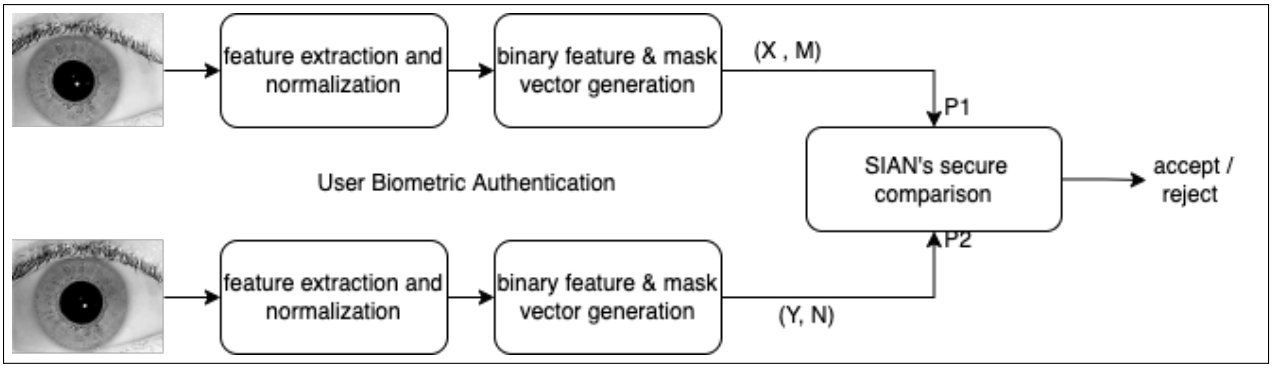

Figure 3.3: Proposed Scheme that extracts iris codes and matches them using a secure two-party computation

Overall, SIAN provides a secure and privacy-preserving method for iris authentication, safeguarding biometric data during both storage and computation phases.

3.2.1 Fixed Length Iris Vector Generation

The proposed protocol, SIAN, is designed to operate effectively with biometric templates using the Hamming distance as the distance measure. In the context of this study, SIAN is evaluated using templates derived from iris datasets. The methodology for generating binary feature vectors is based on the approach outlined in [24]. Here is a breakdown of the specific steps involved:

- **Extraction of Pupil and Iris Regions:** The initial step involves the extraction of both the pupil and iris regions from the iris image. This is accomplished using the integro-differential Daugman operator [23] and the circular Hough transform [38]. Additionally, regions corresponding to the upper and lower eyelids, as well as eyelashes, are introduced as noise components through a thresholding process.
- **Normalization of Extracted Regions:** Following the region extraction step, the obtained regions undergo normalization using the Wildes method [79]. Normalization ensures that the extracted regions are brought to a consistent and standardized format for further analysis.
- **Feature and Mask Generation:** Post-normalization, the feature and mask (which encodes noise) vectors are generated by applying Log-Gabor filters [22]. To configure the filters, angular and radial resolution parameters are initialized as (64, 16). The resulting feature and mask vectors have a size of 2048, with noise regions in the mask designated by a value of 1.

By following this process, binary feature vectors are generated from iris images. These binary feature vectors are then utilized as input for the SIAN protocol.

3.2.2 Authentication Protocol

The proposed authentication scheme is designed to facilitate secure communication between two parties denoted as P_1 and P_2, each possessing vector pairs (X, M) and (Y, N), along with their respective masks. The ultimate goal of the scheme is to compute the Hamming distance (h) between X and Y using the following formula:

$$h = \frac{||(X \oplus Y) \cap X'mask \cap Y'mask||}{||X'mask \cap Y'mask||} \tag{3.1}$$

This computation is performed independently on both sides of the communication. The server, represented as Party P_2, rejects the communication if the computed h exceeds a predefined threshold value denoted as τ. It's important to note that the party initiating the communication is designated as the client, with Party P_1 serving as the client and Party P_2 as the server in the current implementation.

The distribution of vectors is executed as follows: Party P_1 distributes X as X_1 to itself and X_2 to Party P_2, ensuring that $X = X_1 \oplus X_2$. Similarly, Party P_2 distributes Y as Y_1 and Y_2.

The secure computation of the Hamming distance relies on several fundamental operations performed on distributed vectors (X_1, Y_1, M) and (X_2, Y_2, N). These operations include Secure XOR, Secure AND, and Secure Matching, with preprocessing and distillation phases aimed at selecting sequences / octets from noise vectors that exhibit a 25% mismatch. These phases are integral to the overall functionality of the authentication scheme.

3.2.3 Secure XOR

Computing XOR between 2 bits is a local operation and does not require any communication. To compute XOR between X_i and Y_i, both the parties P_1 and P_2 compute : $Z_{1i} = X_{1i} \oplus Y_{1i}$ and $Z_{2i} = X_{2i} \oplus Y_{2i}$ Since $X_i = X_{1i} \oplus X_{2i}$ and $Y_i = Y_{1i} \oplus Y_{2i}$, $Z_{1i} \oplus Z_{2i} = X_i \oplus Y_i$.

3.2.4 Preprocessing Phase

The preprocessing phase involves the creation of m octets, with each octet consisting of 4 bits selected from each party's noise vector. Specifically, let's denote Party P_1's noise vector as $\langle s0, s1, s2, s3 \rangle$, and Party P_2's noise vector as $\langle r0, r1, r2, r3 \rangle$. The selection process ensures that:

$s0 \oplus s1 \oplus s2 \oplus s3 \neq r0 \oplus r1 \oplus r2 \oplus r3$

This condition guarantees either a 1-bit or a 3-bit mismatch in the resulting octet, effectively filtering out the possibilities of 0% and 50% mismatch. It's important to note that this preprocessing phase is a one-time occurrence in the authentication scheme and is not repeated.

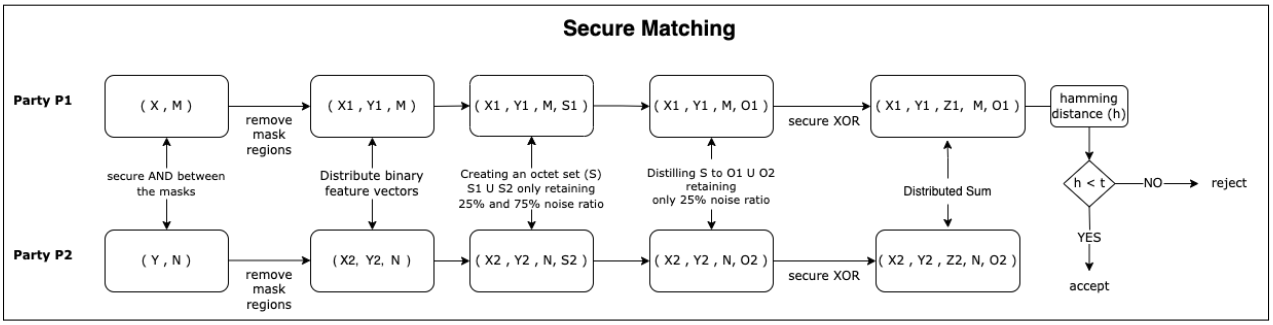

Figure 3.4: Secure Matching Scheme with Party P_1 and P_2 holding (X, M) and (Y, N) binary vectors

The critical aspect of this preprocessing phase is that the positions revealed while comparing the XORs between the parties are not repeated, thus safeguarding the vector from being leaked. The steps involved in this preprocessing phase can be found in Algorithm 1.

Algorithm 1 Preprocessing

1: Let $\langle r0, r1, r2, r3\rangle$ be 4 randomly selected bits from M. P_1 Sends XOR value and bit positions to P_2

2: Let $\langle s0, s1, s2, s3\rangle$ be 4 corresponding bits from N

3: If the XOR values of parties P_1 and P_2 match or more than 1 out of 4-bit positions are repeated, repeat the process from step 1

4: The above process is repeated until m different octets $\langle r0, r1, r2, r3, s0, s1, s2, s3\rangle$ are obtained.

3.2.5 Secure AND

The computation of a distributed AND between two bits, denoted as x and y, involves two steps: selecting a random octet from the octets generated in the Preprocessing phase and performing the Computation phase. This process is outlined in Algorithm 7.

In the Computation phase, partial sums are computed at each party ($z1$ and $z2$) using the biometric noise from the octet ($\langle r0, r1, r2, r3, s0, s1, s2, s3\rangle$). These partial sums are then combined as $z1 \oplus z2$ to obtain the result of the logical AND operation, represented as $x \wedge y$.

The AND gate is applied to the octet at each party to obtain the following sums: $a1 = r2 \oplus r3$, $b1 = r1 \oplus r3$, $c1 = r3$, $a2 = s2 \oplus s3$, $b2 = s1 \oplus s3$ and $c2 = s3$.

The operation $a1 \oplus a2$ yields 1 when either $r2 \oplus s2 = 1$ or $r3 \oplus s3 = 1$, indicating a 1-bit mismatch at position 3 or 4. A similar observation holds for $b1 \oplus b2$, which yields 1 when there is a 1-bit mismatch at position 2 or 4. Finally, $c1 \oplus c2$ yields 1 when there is a mismatch at position 4.

The final expression evaluates the distributed Secure AND operation between x and y in the following cases:

- When only $a1 \oplus a2 = 1$ (indicating a 1-bit mismatch at the 3rd position).

- When only $b1 \oplus b2 = 1$ (indicating a 1-bit mismatch at the 2nd position or 4th position).
- When both $a1 \oplus a2$ and $b1 \oplus b2$ are 1 (indicating a 1-bit mismatch at the 4th position).
- When all combined sums of the AND gate yield 0 (indicating a 1-bit mismatch at the 1st position).

This process works reliably at a 1-bit mismatch or 25% noise ratio. The overall steps for the Secure AND operation are described in Algorithm 3.

Algorithm 2 Computation Phase

1: Let x and y be the input bits with parties P_1 and P_2 respectively for which AND gate needs to be computed. Party P_1 has x_1, y_1 and P_2 has x_2, y_2

2: Both the parties generate a logical truth table of AND gate with 2 inputs, denoted as M. Matrix M is as follows:

$$M = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}_{4\times 3} \tag{3.2}$$

3: Parties P_1 and P_2 compute :

$$\langle a_1, b_1, c_1 \rangle = \langle r0, r1, r2, r3 \rangle_{1\times 4}.M_{4\times 3} \tag{3.3}$$

$$\langle a_2, b_2, c_2 \rangle = \langle s0, s1, s2, s3 \rangle_{1\times 4}.M_{4\times 3} \tag{3.4}$$

4: Both parties share following values over noiseless channel using error-correcting codes:

- P_1 sends $x_1 \oplus a_1$ and $y_1 \oplus b_1$ to P_2
- P_2 sends $x_2 \oplus a_2$ and $y_2 \oplus b_2$ to P_1

5: Both parties will generate shared values X_A and Y_B where $X_A = x_1 \oplus a_1 \oplus x_2 \oplus a_2$ and $Y_B = y_1 \oplus b_1 \oplus y_2 \oplus b_2$

6: $x \wedge y = z_1 \oplus z_2$ where z_1, z_2 are computed by parties P_1, P_2 and shared with each other respectively.

$$z_1 = (X_A \wedge Y_B) \oplus (x_1 \wedge Y_B) \oplus (y_1 \wedge X_A) \oplus c_1 \tag{3.5}$$

$$z_2 = (x_2 \wedge Y_B) \oplus (y_2 \wedge X_A) \oplus c_2 \tag{3.6}$$

Algorithm 3 Secure AND

1: Let x and y be the input bits with parties P_1 and P_2 respectively for which AND gate needs to be computed. Both parties create shares of their input as x_1, x_2, y_1, y_2. Party P_1 has x_1, y_1 and P_2 has x_2, y_2

2: P_1 and P_2 select a random octet $\langle r0, r1, r2, r3, s0, s1, s2, s3\rangle$ and perform Computation Phase (7).

3.2.6 Distillation Phase

In the Preprocessing Phase, only the octets with a 25% or 75% mismatch were retained. However, Secure AND expects only a 25% mismatch; hence, other noise ratios must be filtered out. The Distillation Phase aims to remove octets with a 75% mismatch with a high probability. In this phase, the results of the Computation phase for a pair of random octets are compared. If the distributed AND result is unequal (on a random bit pair), both octets are eliminated. If the result is equal, one of the octets is retained randomly. This process is repeated iteratively for a specified number of times, denoted as n (where $n \leq m$). The resultant octet set will consist of elements that agree on the results of the distributed AND operation.

For an initial octet set with a majority of 25% noise ratio, octets with a 75% noise ratio are removed because their results don't match with many other octets. The steps for this phase are shown in Algorithm 4.

Algorithm 4 Distillation

1: Let S be the set which has a collection of m octets $\langle r0, r1, r2, r3, s0, s1, s2, s3\rangle$

2: Two random octets a and b are picked and Computation Phase (7) is performed on random bits of M and N.

3: P_1 computes $z_1 = z_{11} \oplus z_{12}$ and P_2 computes $z_2 = z_{21} \oplus z_{22}$ where z_{i1} and z_{i2} are the results of Computation Phase for a and b respectively for both parties.

4: Both parties share the XOR values with each other. If the XOR value matches, then one of the octet is retained in the set or else both are discarded

5: We repeat step 2 to step 4 n times.

3.2.7 Secure Comparison

The utilities introduced in the previous sections, including Secure AND, Secure XOR, Preprocessing, and the Distillation Phase, are combined in the Secure Matching phase. Figure 3.4 illustrates the overall scheme. Initially, octets are created from the masks of the parties during the Preprocessing Phase and refined in the Distillation Phase. The masks are securely removed from the feature vectors using Secure AND. Then, on the unmasked vectors, the Hamming distance (Eqn. 4.1) is computed by summing bits from the distributed XOR (Algorithm 9), resembling the logic of a full adder. The steps involved in Secure Matching are detailed in Algorithm 5.

Algorithm 5 Secure Matching

Let Party P_1 and P_2 hold (X, M) and (Y, N) respectively

P_1 and P_2 perform Preprocessing creating m octets (set S)

P_1 and P_2 perform Distillation on S and retain valid octets with a high probability

P_1 and P_2 perform Secure AND between X'_{mask} and Y'_{mask} respectively and compute $X'_{mask} \cap Y'_{mask}$

P_1 and P_2 compute $X \cap (X'_{mask} \cap Y'_{mask})$ and $Y \cap (X'_{mask} \cap Y'_{mask})$ removing noisy regions

P_1 and P_2 distribute vectors and hold (X_1, Y_1, M) and (X_2, Y_2, N) respectively

P_1 and P_2 compute Secure XOR : $Z_1 = X_1 \oplus Y_1$ and $Z_2 = X_1 \oplus Y_2$ where $Z_1 \oplus Z_2 = X \oplus Y$

P_1 and P_2 compute W_1 and W_2 using Distributed Sum (9) and share with each other where $||W_1 \oplus W_2||$ is the number of bits that differ between X and Y

P_2 accepts if $\frac{||W_1 \oplus W_2||}{||X'_{mask} \cap Y'_{mask}||} < \tau$

Algorithm 6 Distributed Sum

Let Z_x be the Distributed XOR computed by Party P_x

$W_x \leftarrow [Z_{x0}]$

$i \leftarrow 1$

while $i \leq d$ **do**

  $v \leftarrow Z_{xi}$

  $i \leftarrow i + 1$

  $j \leftarrow len(W_x) - 1$

    while $j \geq 0$ **do**

      $w \leftarrow W_{xj}$

      $f \leftarrow w \oplus v$

      $g \leftarrow$ Secure AND between w and v

      $W_{xj}, v \leftarrow f, g$

      $j \leftarrow j - 1$

  v is inserted at position 0 of W_x

    end while

end while

3.3 Security

The protocol's security relies on what each party can observe during execution. We adopt a semi-honest security model, often called the honest-but-curious model. Under this model, both parties follow the protocol without deviation but may try to gather information about the other party's input during their interactions. This choice of security model is practical, as each party can detect any deviations in behavior. We also assume security against an adversary with unlimited computational power, ensuring strong security.

In simple terms, a protocol is considered secure in this semi-honest model if a party's view can be replicated using only its input and output [32]. This means there's a simulator in an ideal world that can reproduce a party's view based solely on its input. Mathematically, given the input, the probability distribution across all possible transcripts is indistinguishable from the probability distribution resulting from a single party's view. Consequently, the protocol offers security under a semi-honest security model, protecting each party from misconduct by the other party. A formal proof of this security can be found in the appendix.

3.4 Complexity

The total number of communication bits over the entire protocol execution can be calculated as follows:

- During the input distribution phase, each party communicates n bits to each other, resulting in a total communication of $2n$ bits.
- For secure XOR operations, there is no additional communication cost involved as these operations are performed locally by each party on their respective inputs
- The Secure AND operation has a variable communication cost, which depends on the number of times the preprocessing phase has to be repeated. Let's assume that the preprocessing phase is repeated K times. In each preprocessing phase, for m octets, there is a communication of $m \cdot K \cdot 8 \cdot (\log n) + 2 \cdot K \cdot m$ bits.The total number of bits for all AND gate operations is $m \cdot K \cdot 8 \cdot (\log n) + 2 \cdot K \cdot m + 4n$ bits.

The total communication cost for the entire protocol execution is the sum of these costs.

3.4.1 Datasets & Empirical Observations in Noise Distributions

We conducted an extensive evaluation of our protocol using three widely recognized public iris datasets: the IIT Delhi Iris Database version 1.0 [51, 52], the MMU Iris Dataset v1 [76], and UBIRIS v1 [67]. These datasets were chosen because our adopted iris feature extraction method consistently produced reliable features on them.

The IIT Delhi Iris Database v1.0 consists of 1,120 Near-Infrared (NIR) images captured in a controlled environment, comprising 448 distinct classes (with 224 classes each for the left and right eye). In contrast, the MMU Iris Dataset v1 contains 450 images collected from 45 individuals (90 unique classes). These images were captured using a dedicated iris scanning sensor, which introduced challenges such as eye rotation and eyelash obstruction.

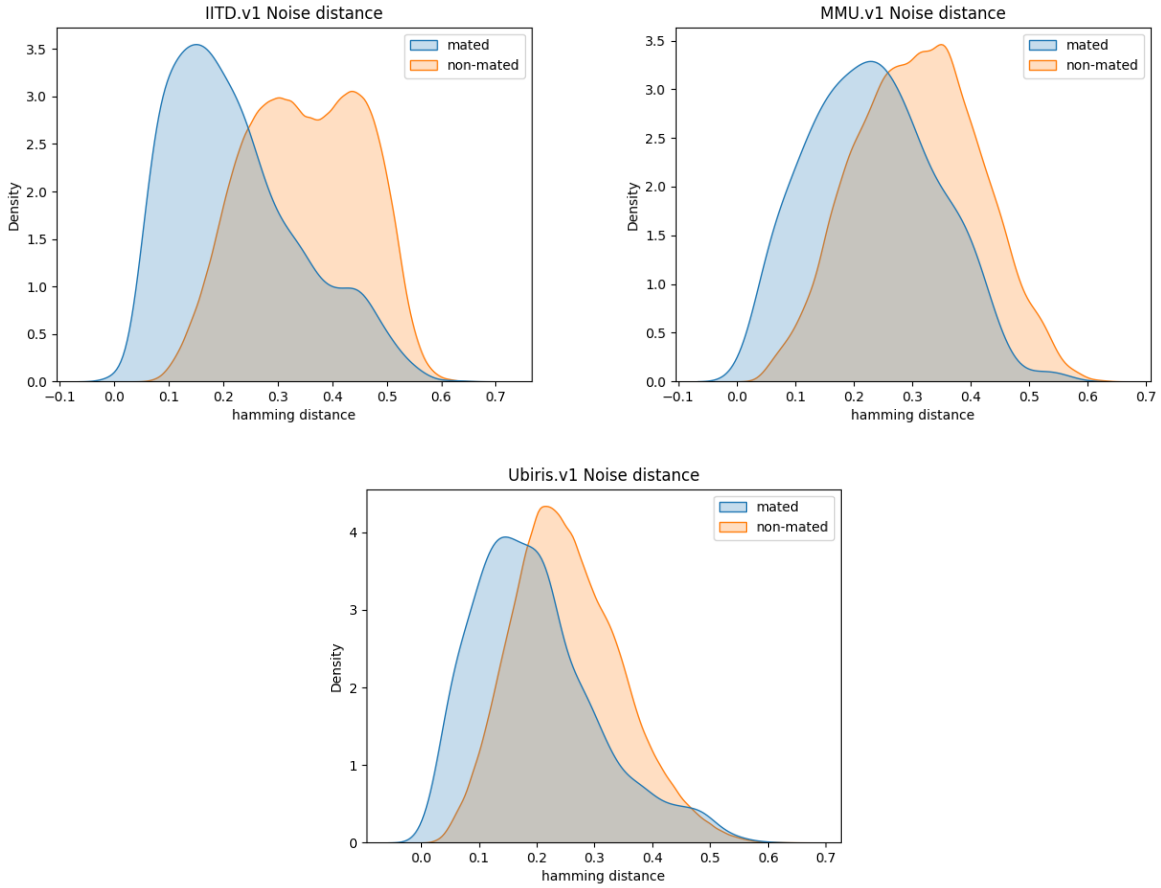

Figure 3.5: Distribution of the Hamming distance between noise codes for each dataset

The UBIRIS v1 dataset contains noisy images captured in a less constrained environment. For our experimentation, we focused on session 1 of the monochromatic subset, which is less noisy and comprises 1,205 images distributed across 241 classes.

In Figure 3.5, we present the distribution of Hamming distances between noise vectors for both mated and non-mated pairs. Notably, we observe that the distribution of non-mated scores consistently falls to the right of the mated scores across all datasets. This characteristic indicates that the average noise ratio for mated pairs remains below 25%, while it exceeds 25% for non-mated pairs (as detailed in Table 3.2).

We find that our protocol demonstrates superior performance when the noise ratio remains at or below 25%. These real-world dataset observations align with the underlying assumptions of our protocol and are further corroborated by its strong verification performance.

3.5 Parameters of the Protocol

The parameter m within the Preprocessing Phase plays a crucial role in determining the number of octets available for Secure AND operations. Specifically, it governs the size of the octet set used in the protocol. As

Dataset	mated	non-mated
MMU.v1	0.2255	0.2983
IITD.v1	0.2217	0.3498
Ubiris.v1	0.1959	0.2528

Table 3.2: Average noise code Hamming distance for mated vs non-mated comparisons

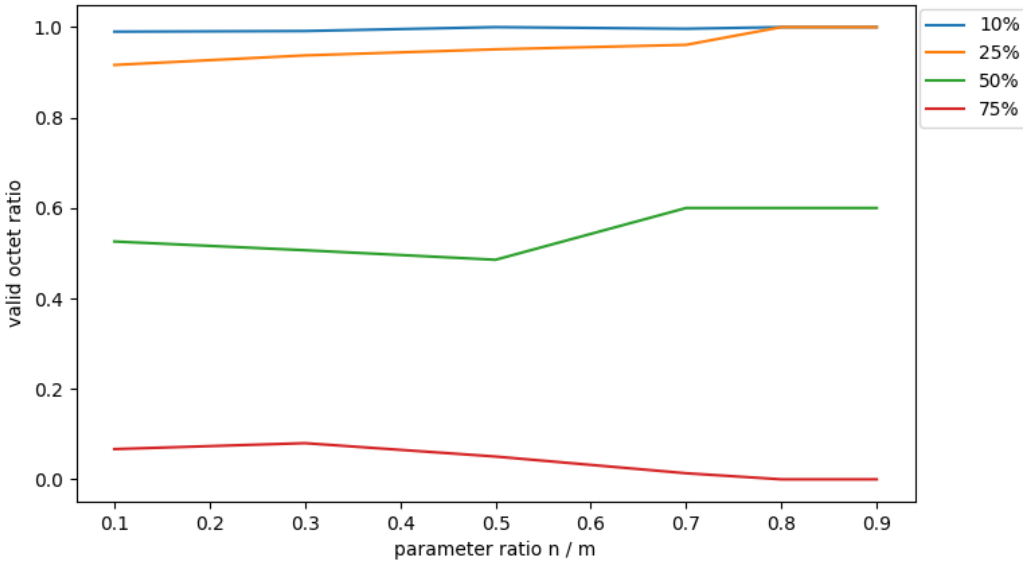

Figure 3.6: Percentage of valid octets at different noise ratio levels with the increase in parameter ratio n/m after Distillation

m increases, and as the ratio n/m becomes larger, the probability of selecting valid octets (those with a 1-bit difference) after the Distillation Phase also increases. This is primarily because a larger m leads to a wider set of available octets to choose from, enhancing the likelihood of finding valid octets.

However, it's important to note that there are trade-offs associated with choosing larger values of m. For instance, as m becomes larger, finding octets with one or three-bit differences becomes more challenging, especially when dealing with noise ratios other than the standard 25% and 75%. This increased difficulty in finding these specific octet differences can slow down the protocol because it requires more attempts in the Preprocessing Phase.

To strike a balance between protocol speed and the ability to select valid octets, the value of m is typically set to be the minimum of 100 and one-fourth of the feature size in bits (N).

Additionally, the value of n used in the Distillation Phase is set to be 90% of m ($0.90 \times m$). This choice is based on observations, as shown in Figure 3.6, where it was found that the percentage of valid octets

(repeated for ten iterations) reaches 100% when n is set to $0.90 \times m$, particularly for noise ratios at or below 25%.

3.6 SIAN Accuracy comparsion over Noise Ratios

The accuracy mentioned here pertains to the Hamming distance, specifically the agreement between Hamming distances computed with and without the protocol. In the case of Secure AND (SIAN), this accuracy reaches 100% when the noise ratio is at or below 25% due to the Distillation Phase, which effectively retains only octets with a 1-bit difference. This is empirically demonstrated through the calculation of averaged accuracy values across different noise levels after 100 iterations, where uniform random noise is applied (see Table 3.3).

Figure 3.7 provides a graphical representation of this phenomenon. It shows that accuracy decreases gradually until it reaches 75% at the 50% noise ratio mark. This decrease can be explained by the gradual increase in the selection of octets with 3-bit differences after the Distillation Phase.

However, accuracy drops to 0 for a uniform random noise ratio of 75% and above. This drop occurs because the probability of selecting octets with a 25% noise ratio (1-bit difference) after the Distillation Phase becomes zero when the noise ratio exceeds 75%.

In summary, the accuracy of the protocol is closely tied to the noise ratio, with optimal performance observed at low noise levels, while it degrades as the noise ratio increases and more 3-bit difference octets are selected after Distillation.

These illustrations effectively showcase the expected behavior of the protocol within its designed assumptions. However, it's important to acknowledge that real-world datasets may not always strictly adhere to the assumed noise ratios, despite being concentrated around these values, particularly for mated and non-mated pairs. This variability in noise levels highlights the necessity of developing mechanisms to address and mitigate the inaccuracies that may arise in practice.

Real-world datasets often exhibit a degree of noise and variability that can challenge the robustness of any protocol or algorithm. Therefore, the development of techniques or strategies to handle and adapt to such real-world variations is essential for ensuring the protocol's effectiveness and reliability in practical applications. These mechanisms may include noise modeling, adaptive parameter tuning, or advanced error correction techniques to enhance the protocol's performance in less ideal conditions.

3.7 Correction Mechanism

The protocol encounters difficulties when noise ratio constraints exceed 25%. This issue is evident in Figure 3.7, especially when the noise ratio goes beyond 50%. Our observations show that the Hamming distance errors (h) in these scenarios fall outside the expected range of values between 0 and 1, rendering them invalid. To address this problem, we have developed a scheme, as depicted in Figure 3.8, aimed at mitigating these errors.

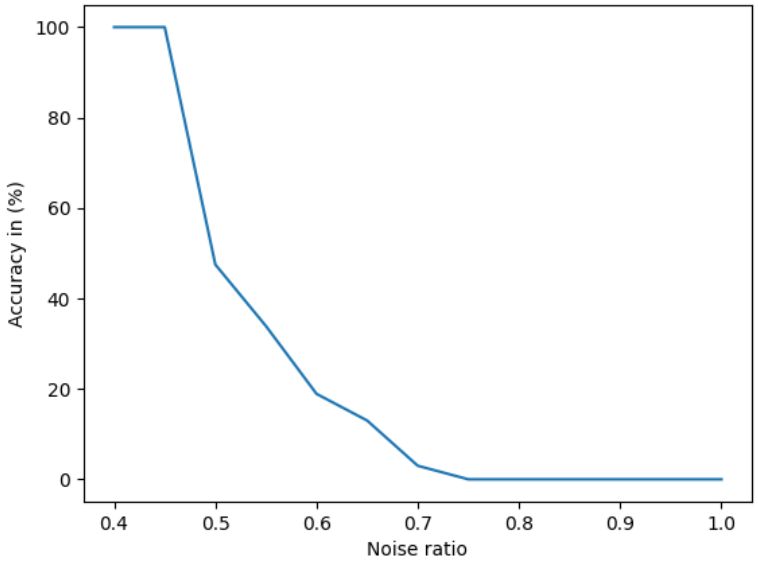

Figure 3.7: Accuracy drop for noise ratio ≥ 0.5

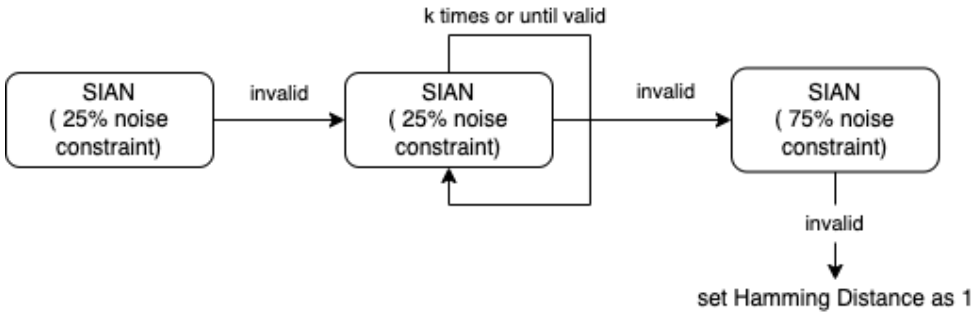

Figure 3.8: Correction Mechanism

Table 3.3: Average time and accuracy for uniform random noise

Noise Ratio %	Time Cost (s)	Accuracy %
10	0.460	100
15	0.457	100
25	0.432	100
50	0.445	47.5
75	0.438	0

In cases where the initial Hamming distance is deemed invalid, we make an assumption that places it within the 50% noise ratio domain. To rectify this, we engage in a repeated secure computation process, iterating it a total of k times until a valid distance is successfully derived. If, by any chance, the obtained distance remains invalid even after these iterations, we introduce a modified version of the protocol.

In this modified version, we employ a transformation matrix M represented as:

$$M = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}_{4\times 3} \tag{3.7}$$

Notably, this version incorporates a truth table for NAND operations instead of AND. This adaptation is designed to accommodate scenarios with a presumed noise ratio greater than or equal to 75%. The result of employing this modification can yield an accuracy rate of up to 100%, but it comes at a computational cost. To strike a balance between accuracy and efficiency, we set the value of k to 5, a choice that effectively resolves all error cases, except when the noise ratio results in an average accuracy of less than 20% (for noise ratios greater than or equal to 65%, as illustrated in Figure 3.7). In such instances, the value of the remaining invalid distances is then set to 1.

3.8 Verification Performance

To assess the performance of SIAN (Secure Iris Authentication Network), we utilize two key evaluation metrics: FNMR (False Non-Match Rate) at FMR0 (False Match Rate) and EER (Equal Error Rate) across all three datasets. To ensure a reliable evaluation, we focus on a subset of images where the feature generation algorithm effectively isolates the iris region.

In our analysis, we conduct mated comparisons of iris images within the same class and comparisons of non-mated pairs. For the latter, we select one sample from each class and compare it to a sample from each of the other classes. These comparisons are performed after applying eight shifts in both left and right

directions. Subsequently, we record the minimum Hamming Distance for both scenarios: with and without the protocol computations.

Upon examining the results presented in Table 3.4, it becomes evident that the IITD v1 dataset [51, 52] exhibits the lowest FNMR, while the Ubiris v1 dataset [67] displays the highest FNMR. This discrepancy can be attributed to the presence of more noisy images in the Ubiris v1 dataset, which challenges the accuracy of the iris matching process.

Dataset	FNMR / FMR0%	EER%	EER% (local)	Difference in (%)
MMU.v1	5.92	4.14	4.07	0.07
IITD.v1	2.643	1.86	1.856	0.04
Ubiris.v1	8.36	5.44	5.39	0.05

Table 3.4: Verification Performance on Datasets

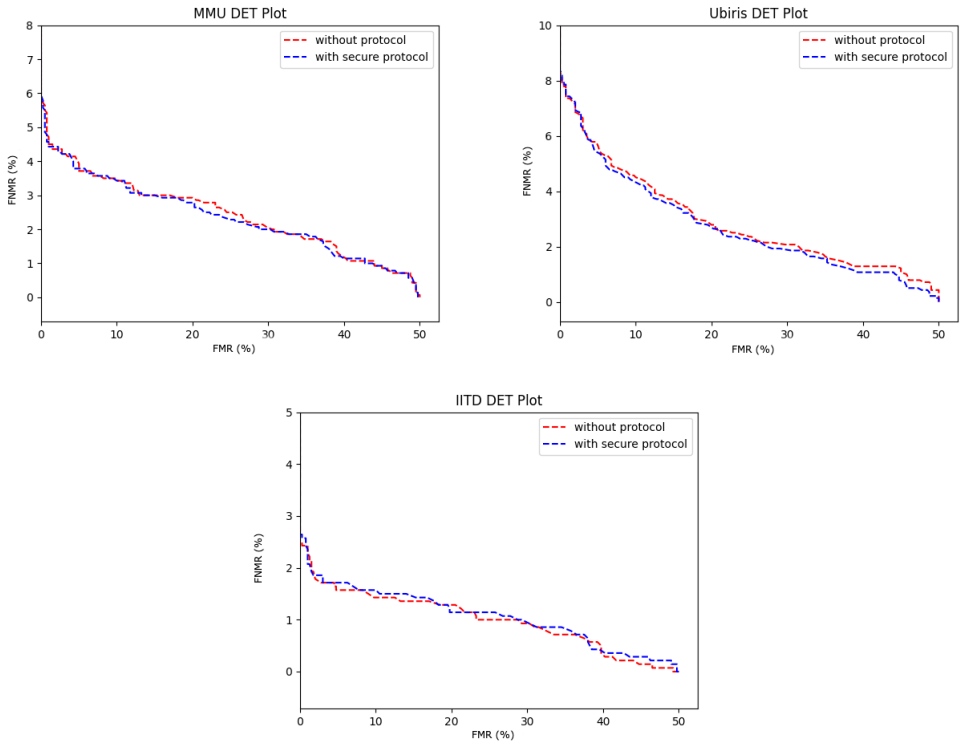

Figure 3.9: DET plot for each dataset

In Table 3.4, the term EER (local) pertains to computations conducted locally, excluding the protocol. Interestingly, the protocol's EER closely aligns with the local computation, indicating a minimal discrepancy between them. Specifically, the average difference in the protocol's EER compared to the local EER across the datasets can be calculated as follows:

$$d = (0.07 + 0.04 + 0.05)/3 = 0.056\% \tag{3.8}$$

The choice of the correction mechanism parameter, denoted as k and set to 5, takes into account this small error, and higher values of k could potentially reduce the average error even further.

The DET plots in Figure 3.9 illustrate a slight deviation of SIAN from the path of local computations. This deviation diminishes as one moves from right to left on the FMR scale, approaching near equality at FNMR0. The introduction of this deviation primarily stems from the protocol's treatment of invalid Hamming Distances, where they are set to 1. This practice results in an increase in FNMR and a decrease in FMR values for the protocol. Consequently, it causes lower FNMR values to be observed at larger FMR values, acting as an ancillary factor in the evaluation.

3.9 Time Cost Analysis

The protocol's implementation was carried out in Python 3 and assessed on a machine powered by the Apple M1 CPU with 16 GB of RAM. To optimize the computation of the Distributed Sum in Secure Matching, we applied batch processing by dividing the vector into sizes of 128 and then consolidating the sums. This approach was employed to enhance computational efficiency.

Notably, SIAN exhibits favorable computational characteristics, rendering it suitable for testing on less resource-intensive machines. To gauge its time efficiency under various uniform noise conditions, we conducted evaluations across 100 iterations and calculated the average results, as outlined in Table 3.3.

It's worth highlighting that SIAN demonstrates its highest time performance at a 25% noise ratio, as illustrated in Table 3.3. However, it's important to note that in real-time scenarios, SIAN may encounter timeouts during both the no-noise and 100% noise scenarios at the Preprocessing Phase, we ignore them since they rarely occur.

Table 3.5: Comparing the time and memory performance of the protocol for larger binary feature vectors (N)

N (bits)	Average time cost (s)	After offline preprocessing (s)	Bandwidth (KB)
2408	0.432	0.214	148.98
9600	1.897	1.571	468.75
19200	3.862	3.108	1293.67
28800	5.652	5.130	1383.92
38400	7.727	7.067	1842.46

Enhancing the protocol's efficiency is achieved through an innovative caching mechanism, wherein octets are stored after the initial user authentication by the device. This caching procedure results in a notable reduction in processing time, specifically by a significant margin of 200-800 milliseconds. This optimization is categorized as offline preprocessing.

Table 3.5 provides insights into the relationship between the time cost and the size of the feature vector (N). It reveals that the time cost increases linearly as the feature vector size grows larger. However, it is reassuring to note that the protocol's bandwidth usage remains below 2 megabytes, ensuring that it remains lightweight and memory-efficient.

Importantly, the increase in feature vector size does not lead to a decline in accuracy. The protocol maintains its accuracy across various feature vector sizes, indicating its robustness and scalability in handling larger datasets.

	N(Bits)	Time Cost(s)	Proposed Method		
			online & offline	online	online without mask removal
Luo et al. [57] (mask)	2408	0.56	0.432	0.214	0.118
Luo et al. [57] (mask)	9600	2.5	1.897	1.571	1.184
Bringer et al. (1:1) [14]	2048	16.8	0.432	0.214	0.118
Droandi et al. (online) [6]	6400	0.120	1.330	0.928	0.788
Pia et al. [7]	5120	0.503	0.897	0.626	0.408

Table 3.6: Comparison of latency with other methods

Efficiency is a key aspect when evaluating the performance of SIAN in comparison to other methods, and Table 3.6 presents a comprehensive comparison across three different modes: including the Preprocessing phase (offline + online), caching octets (offline), and excluding the mask removal process (online without mask removal) during the Secure Matching Phase, recognizing that some methods do not employ mask removal.

The overall latency of SIAN proves to be superior when compared to the considered variants of methods presented in [57] and [14]'s verification techniques. It's important to note that while the approach outlined in [6] exhibits better performance in the online phase, when evaluating the total cost, which encompasses both offline and online phases (where the offline phase can be costly for SPDZ [4]), SIAN maintains a competitive advantage.

Furthermore, SIAN's time cost is assessed in comparison to [7], a post-quantum security approach. In this evaluation, it is observed that the online phase without mask removal in SIAN demonstrates lower latency compared to the alternative approach, reaffirming its efficiency in the context of post-quantum security considerations.

Chapter 4

Secure Face & Fingerprint Verification using Noise

4.1 Introduction

This chapter delves into the enhancement of noise-based verification techniques applied to fingerprint and facial recognition modalities, which stand out as the most popular choices in the realm of biometrics. Their widespread adoption underscores the significance of the topics we delve into in this chapter.

Many conventional approaches rely on the Euclidean distance metric to compare templates, often employing classical methods like eigenface and minutia-based feature extraction. In contrast, we leverage cutting-edge deep learning methods to extract features, thereby enhancing versatility. These deep learning techniques not only yield more precise verification results but also expedite feature extraction, especially when harnessed with modern GPUs and TPUs, thereby enabling the creation of scalable solutions.

Furthermore, in addition to harnessing deep learning for feature and noise extraction, we introduce supplementary phases aimed at augmenting the octet count, building upon the preprocessing stages outlined in Chapter 2. Given that many deep learning models employ the dot product in template comparison, we've devised a robust verification protocol to ensure secure dot product operations. Additionally, we've developed a correction mechanism to complement this proposed verification protocol. These utilities are integrated into a comprehensive framework where we explore various methods of noise extraction tailored to each modality while performing the verification process.

Our methods undergo rigorous testing on a total of seven publicly available datasets, yielding promising results that highlight their applicability in real-world scenarios. To ensure a thorough assessment, we undertake a multifaceted evaluation that includes security analysis, complexity analysis, verification performance analysis, and time-cost performance analysis. We also conduct comparative assessments, comparing our proposed approach with alternative methods, thus offering a holistic view of its strengths and advantages. We also dedicate attention to investigating its limitations and areas for future research and development.

In the interest of transparency and reproducibility, we provide pseudocode for all the algorithms employed throughout the chapter. Additionally, we delve into the implementation aspects, focusing on efficiency measures, experimental settings that we have incorporated for the same purpose. This documentation and analysis aim to provide a comprehensive and actionable understanding of the methods, their potential, and their practical implications in biometrics and related fields.

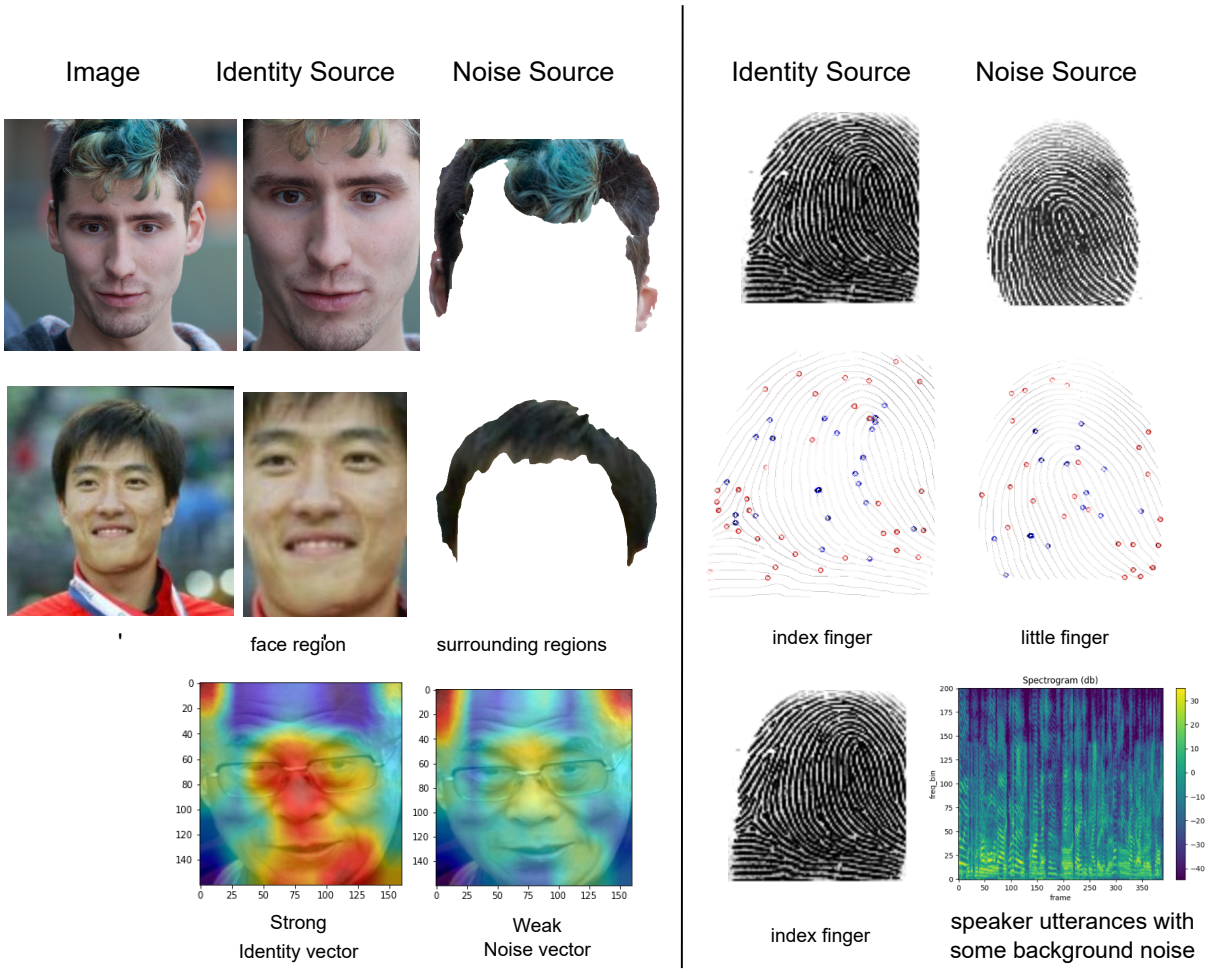

Figure 4.1: Noise extraction mechanisms for face (left) and fingerprint (right) modalities

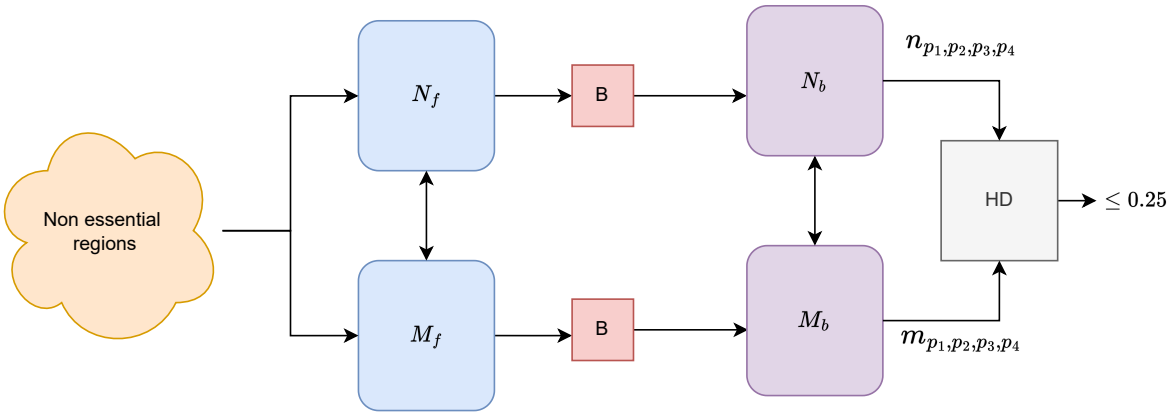

Figure 4.2: Constructing an octet set with elements m_{p_1,p_2,p_3,p_4} and n_{p_1,p_2,p_3,p_4} from N_f and M_f obtained from noise / non-essential regions for verification

Figure 4.1 offers a concise overview of the diverse noise extraction strategies employed in this study. These strategies are instrumental in enabling the verification protocol to effectively utilize the uncertainty stemming from the noise features. To provide a high-level summary, we present noise extraction under four distinct strategies:

- **Noise-from-Same-Source-Same-Modality :** This strategy involves extracting a noise vector from the surrounding regions of the face biometric.
- **Noise-from-Different-Source-Same-Modality :** Here, a noise vector is extracted from a small surface area of the fingerprint (specifically, the little finger), while verification is conducted using an index finger.
- **Noise-from-Different-Source-Different-Modality :** In this approach, a noise vector is extracted from a different modality, namely noisy speaker utterances, while fingerprint verification is the primary focus.
- **Noise Vector Generator :** We introduce a self-supervised training mechanism designed to generate a weak noise vector from a face biometric that is uncorrelated with its identity vector. The saliency map illustrating this process is presented in Figure 4.1.

4.1.1 Core Observation

The central observation remains consistent with what was previously discussed in the preceding chapter. Our underlying hypothesis revolves around the notion that features extracted from non-essential regions of a biometric sample, which are not directly relevant to the current authentication application, can still play a role in enhancing security, especially when their mismatch rate is in the vicinity of approximately 25%.

We make use of the security octets, which are obtained from the binarized noise features processed from the non-essential regions for verification. They are expected to satisfy the mismatch rate noise ratio of 25% In the upcoming sections, we delve into a detailed exploration of the binarization process for these noise features. We aim to establish the desired relationship between the noise templates, further solidifying the foundation of our security-enhancing approach.

In a formal context as shown in the Figure 4.2, we define the binarization process for two noise features, denoted as N_f and M_f, as follows: we obtain binarized representations, $N_b = B(N_f)$ and $M_b = B(M_f)$, which we shall refer to as "noise codes," as introduced in the preceding chapter. Here, B represents a binarization function that ensures the preservation of the same correlation in the Hamming space as in the cosine space, expressed as $HD(M_b, N_b) \sim N_f \cdot M_f$, where HD is the normalized Hamming Distance.

We observe that when $HD(M_b, N_b) \sim 0.25$, empirically selecting random 4-bit subsets, denoted as m_{p_1,p_2,p_3,p_4} and n_{p_1,p_2,p_3,p_4} and expressed as m_p and n_p, respectively, most of these constructed subsets exhibit a Hamming distance, denoted as $HD(m_p, n_p)$, that is less than or equal to 0.25. This behavior arises because, for a long-bit vector with an HD of 0.25, there must be a substantial degree of similarity between the elements at corresponding positions.

To harness this observation, we leverage the mechanisms of preprocessing, distillation introduced in the previous chapter to prevail the elements within the set that possess the majority's property.

4.1.2 Vulnerabilities of Deep Biometric templates

Deep templates have gained widespread acceptance in modern biometric applications, surpassing classical approaches in face and fingerprint recognition. However, the adoption of deep templates also brings about significant security concerns. When a deep template is compromised in plaintext, it opens the door to identity theft, allowing attackers to exploit the associated services for which the template is utilized. Additionally, deep templates are susceptible to black-box attacks if the feature extraction model is known; adversaries can reconstruct the biometric using an alternative adversarial model.

Given these vulnerabilities, it is paramount to reiterate the primary goal of the methods proposed in this chapter. Our focus is on achieving privacy-preserving verification for these extensively employed modalities within a two-party system situated remotely. The approach we present ensures that no information leaks between the two parties during the verification process. Both parties operate within the Secure Multi-Party Computation (SMC) paradigm, wherein "two parties interactively compute a function using their private inputs without revealing any information other than the output itself." This framework guarantees the higher levels of privacy and security in the verification of these critical biometric modalities.

4.1.3 Usefulness of Noise for Security

In our approach, we employ the octet set constructed from noise features to perform Secure AND gate operations, as detailed in the previous chapter. Within the Secure Multi-Party Computation (SMC) framework, XOR and NOT gates are considered local operations, which means their complexity is relatively low. However,

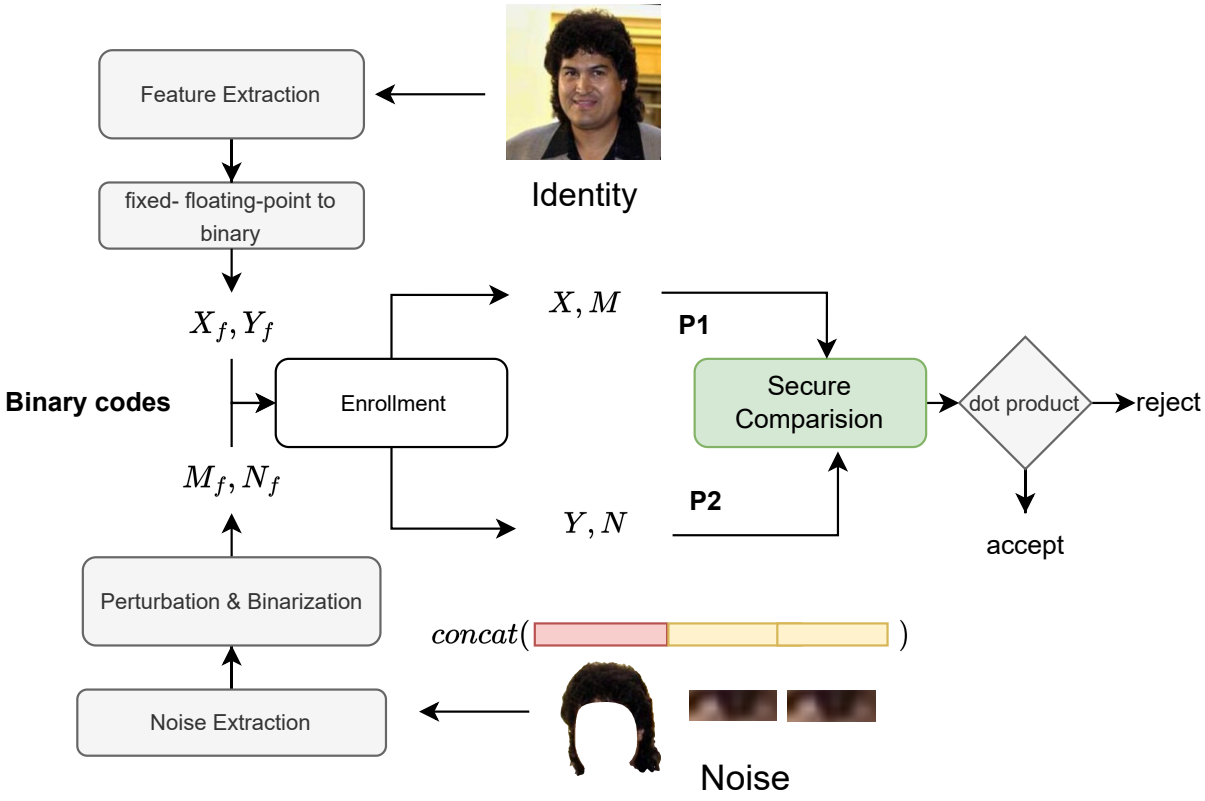

Figure 4.3: S-BAN Approach, with its application to face modality

the complexity significantly increases when computing a distributed AND gate. Subsequently, we combine these gates to compute various functions, ensuring the security of the computation.

The security assurances are primarily facilitated by the presence of random noise octets, which function as security keys and are used for hashing binary private distributed inputs in the derived relation for distributed AND. Hence, noise plays an important role in securing private inputs during transfer.

4.1.4 Motivation

The central motivation behind this chapter centers on the application of a novel design that enables secure remote verification using features extracted from seemingly unimportant regions. Furthermore, we aim to expand this approach to encompass various modalities within a framework that operates on deep templates—a prevalent choice in modern applications—instead of relying on classical comparison functions. What sets our method apart is its independence from cryptographic assumptions between the parties, relying instead on the inherent properties of noise to deliver its functionalities.

4.2 Methodology

In this chapter, we introduce a robust verification protocol that leverages secure two-party computation while utilizing biometric noise for enhanced security. The primary objective of this protocol is to compute the dot product between two fixed-sized templates securely. Furthermore, we establish mechanisms for extracting the necessary noise codes crucial for the verification of both face and fingerprint modalities. Together, these two integral components comprise the proposed S-BAN (Secure Biometric Authentication using Noise) framework, as illustrated in Figure 4.3. To provide a brief overview of the process, we start by obtaining noise embeddings N_f and M_f from either the same or different modalities, which are subsequently converted into binary codes or noise codes denoted as N_b and M_b. Simultaneously, the feature embeddings X_f and Y_f are transformed into a fixed floating-point binary representation, denoted as X_{fb} and Y_{fb}.

The protocol operates on the distributed shares of feature templates X_{fb} and Y_{fb} and utilizes a selected set of random 4-bit pairs/octets for security, which are derived from N_b and M_b. Additionally, we enroll X_{fb} and Y_{fb} as $X = X_{fb} \oplus R$ and $Y = Y_{fb} \oplus R$, where R represents a random bit-vector. This arrangement ensures the secure storage of the templates X_{fb} and Y_{fb}.

In the S-BAN framework, we compute a secure dot product using the distributed shares of encrypted inputs X, Y, assisted by a share of R, and the octets obtained from N and M. This approach guarantees the privacy and integrity of biometric verification.

4.2.1 Verification Protocol

In the design of this protocol, we utilize the underlying utilities introduced in the SIAN scheme [60], which not only offers information-theoretic security but also enables the composition of complex binary functions through its secure AND gate utilizing noise. Below, we provide an overview of the SIAN method and describe the utilities proposed within our protocol.

SIAN involves preprocessing and distillation phases to construct a set of octets, with each element represented as $o = \langle r, s \rangle$. These octets are distributed across two parties, P_1 and P_2, where $r = \langle r_0, r_1, r_2, r_3 \rangle$ and $s = \langle s_0, s_1, s_2, s_3 \rangle$. Notably, r and s are independently obtained from noise codes M and N at parties P_1 and P_2, respectively. Each o_i constructed during the offline phase adheres to a 1-out-of-4 mismatch or a 25

The Secure AND gate, defined in the computation phase of SIAN, adapts Beaver's triplets [8], employing r and s and leveraging the identities established after applying a logical AND gate M_a to r and s, respectively. This enables the computation of a distributed AND as: $p = x \wedge y = z1 \oplus z2$.

In addition to the utilities offered by SIAN, we also observe the following properties to be useful in the design of our protocol:

- Secure XOR between two distributed shares of $x = x_1 \oplus x_2$ and $y = y_1 \oplus y_2$ can be performed as $z_1 = x_1 \oplus y_1$ and $z_2 = x_2 \oplus y_2$ at each party, where $x \oplus y = z_1 \oplus z_2$.
- Secure NOT can be computed on $x = x_1 \oplus x_2$ as $\neg x = \neg x_1 \oplus x_2$, effectively flipping one of the party's share.

- Secure AND between distributed shares and constant values, such as $0 \oplus 0$ or $1 \oplus 0$, can be computed locally, yielding $p = x \wedge 0 = 0$ and $p = x \wedge 1 = x = x_1 \oplus x_2$.

In the protocol, it is essential for the feature vectors X_f and Y_f to undergo $l2$ normalization before they are converted into X and Y. We perform the conversion of (X, M) from (X_f, Y_f) using the Q [5] format, with a size of $Q1.8$ (m-bits for sign and integral part, n-bits for the fractional component) for each value within an n-dimensional feature vector. After multiplication, the size of the data increases to $Q2.16$. However, this increase in size is not a significant concern, given that the range and domain are known to be within $(-1, 1)$ after normalization.

With the transformed values of (X, Y, M, N) at hand, we securely compute the following function between two parties, P_1 and P_2.

$$c = \sum_i X_{fi} Y_{fi} \tag{4.1}$$

The value c is compared with threshold t by P_2 acting as the server in our implementation for 1:1 comparison. P_1 and P_2 hold the shares $\langle X_1, Y_1 \rangle$ and $\langle X_2, Y_2 \rangle$ satisfying $X = X_1 \oplus Y_1$, $Y = Y_1 \oplus Y_2$ and $R = R_1 \oplus R_2$, where R_2 is shared to P_2 by P_1.

4.2.1.1 Secure AND on Encrypted Storage

Let $R = [r_0, r_1, \ldots, r_i, \ldots, r_n]$, where x and y are located at some position within X_b and Y_b, respectively. In this specific context, Secure AND needs to compute:

$$(x \oplus r_i) \wedge (y \oplus r_j) = (x \wedge y) \oplus (y \wedge r_i) \oplus (r_i \wedge r_j) \oplus (r_j \wedge x).$$

To isolate and obtain the result for $x \wedge y$, both P_1 and P_2 must perform additional computations. They need to calculate $(y \wedge r_i)$, $(r_i \wedge r_j)$, $(r_j \wedge x)$, and also $(x \oplus r_i) \wedge (y \oplus r_j)$ using the Secure AND operation. After these computations, the resulting shares are combined using Secure XOR, which is a local operation, to obtain the distributed results z_1 and z_2 at each party.

4.2.1.2 Secure Comparison

The Secure Comparison process involves a series of steps, as detailed in Algorithm 7. This procedure utilizes Distributed Multiplication, as outlined in Algorithm 10, incorporating Secure AND operations on encrypted storage. Furthermore, Addition, described in Algorithm 9, is applied to the distributed input pairs $\langle X_1, Y_1 \rangle$ and $\langle X_2, Y_2 \rangle$ to yield the output variable c (referenced as Eq. 4.1).

The Secure Comparison protocol leverages noise codes with dimensions (N, M) to ensure security. To enhance the protocol's performance during Distributed Multiplication, we adopt a strategy that involves batching independent communications and parallelizing the computation of product shares. This approach significantly reduces the time required for the computation.

A comprehensive overview of the entire scheme, including the preprocessing steps involved in Secure AND with Independent Addition Noise (SIAN), is provided in Figure 4.4.

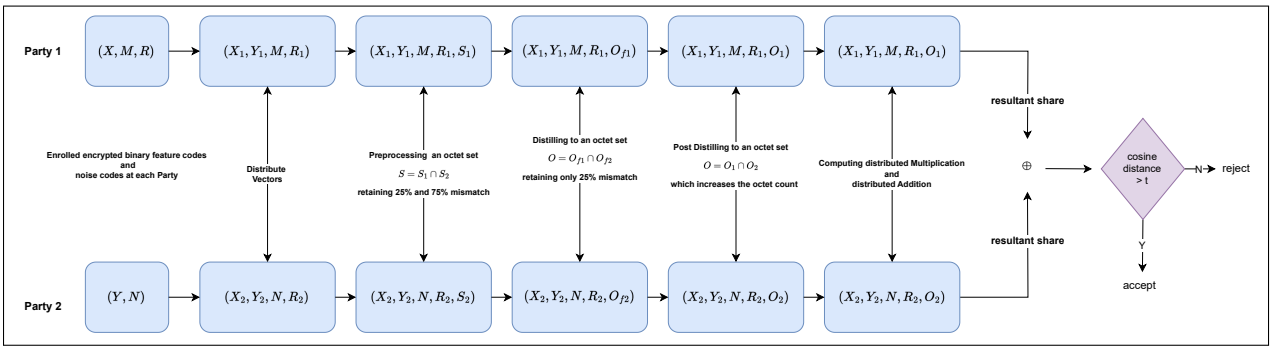

Figure 4.4: Secure Comparison with Party P_1 and P_2 holding (X, M, R) and (Y, N) binary vectors

Algorithm 7 Secure Comparison

1: Let X, Y and M, N be the input feature and noise codes at each party P_1, P_2

2: P_1, P_2 distributes shares of $\langle X, Y \rangle$, R as X_1, Y_1 and $\langle X_2, Y_2, R_2 \rangle$

3: P_1, P_2 perform Preprocessing and Distillation as detailed in SIAN [60], retaining octets with noise ratio of 25% with a high probability to obtain octet set O_f

4: P_1, P_2 perform Post-Distillation 8 on O_f to obtain O

5: P_1, P_2 perform Distributed Multiplication (supplementary)

6: P_1, P_2 perform Distributed Addition (supplementary) of all the products to compute the final result $c = c_1 \oplus c_2 = \sum_i X_{fi} Y_{fi}$, P_2 compares it with a threshold t for Authentication

Algorithm 8 Post-Distillation

Let O be an m element filtered octet set after distillation and S, the original octet set
$i \leftarrow len(O)$
while $i < k$ **do**
    P_1, P_2 select an octet e_1 from S, not part of O
    P_1, P_2 select an octet e_2 from O
    P_1, P_2 compute Secure AND using e_2 and e_1
    With distributed results at P_1, P_2 for e_1, e_2
        $b_1 = z_{11} \oplus z_{21}$ and $b_2 = z_{12} \oplus z_{22}$
        is computed at each party P_x and shared
    if $b_1 = b_2$ **then**
            add e_2 to O
            $i \leftarrow i + 1$
    end if
end while

Moreover, we introduce a post-distillation phase (outlined in Algorithm 8). This phase serves to augment the number of generated octets, which is a desirable feature for enhancing security. Each Secure AND operation involves the random selection of an octet. In cases where noise codes exhibit a noise ratio/mismatch averaging at approximately 25%, the post-distillation phase is generally unnecessary, as it can efficiently generate a substantial number of octets with only 1-bit mismatches. However, in noise ratio regions approaching 50%, the majority of octets are nullified during distillation. Here, the post-distillation phase becomes invaluable.

Additionally, by utilizing random four-bits, we conserve the usage of noise codes, as each octet can only be employed once. Through post-distillation, it is possible to create an octet set with a noise ratio equivalent to that of the distilled set, preserving the overall security of the system.

4.3 Feature Vector Generation

The proposed verification protocol is designed to be compatible with fixed-length biometric templates and employs cosine distance as the distance measure. In this study, we have applied this protocol to two distinct biometric modalities: face and fingerprint recognition.

For face recognition, we have utilized the pretrained Facenet model [70], which is based on the Inception-ResNet-v1 architecture. This model provides 512-dimensional embeddings for face templates. These embeddings are further processed by normalizing them and converting each value to a $Q1.8$ binary representation, as outlined in Section 4.2.1.

Algorithm 9 Distributed Addition

Let A_x and B_x be the vetcor shares at party P_x

$\wedge_s$ denotes Secure AND, $\neg_s$ denotes Secure NOT

Initial carry c_x is $A_{dx} \wedge_s B_{dx}$

$S_x \leftarrow [A_{dx} \oplus B_{dx}]$

$i \leftarrow d - 1$

while $i \geq 1$ **do**

  $s \leftarrow A_{ix} \oplus B_{ix} \oplus c_x$

  $o \leftarrow A_{ix} \wedge_s B_{ix}$, $p \leftarrow B_{ix} \wedge_s c_x$, $q \leftarrow c_x \wedge_s A_{ix}$

  $c_x \leftarrow \neg_s(\neg_s(o) \wedge_s \neg_s(p \oplus q))$

  $S_x \leftarrow [s] + S_x$

  $i \leftarrow i - 1$

end while

Algorithm 10 Distributed Multiplication

Let A_x and B_x be the vector shares at party P_x

$\wedge_e$ denotes Secure AND on encrypted inputs

P_x compute partial products by performing $\wedge_e$ at each A_{xi} against B_{xi} positions, from right to left

P_x for each partial product O_{xi} pads i zeros to the right

P_x performs sign extension by repeatedly adding MSB of O_{xi} to its left for $2 * d - len(O_{xi})$ times

P_x performs 2's complement on the last partial product O_{xd} using Secure NOT and Distributed Addition 9 of 1

P_x performs Distributed Addition 9 of all partial products O_{xi}

In the case of fingerprint recognition, we have employed Deeprint [30], which provides fused representations for fingerprints. These fused representations are 192-dimensional. Similar to the face modality, we normalize these embeddings and then convert each value to a $Q1.8$ binary representation using the procedure detailed in Section 4.2.1.

This approach allows us to effectively compare biometric templates from both face and fingerprint modalities using the cosine distance measure within the proposed verification protocol. By converting the embeddings to a common binary representation, we facilitate the comparison process, ensuring compatibility and accuracy across different biometric modalities. The use of pretrained models and normalization techniques ensures that the biometric data is appropriately processed for reliable verification.

4.4 Noise Code Generation

In our research, we introduce the concept of noise codes, which are binarized representations deliberately engineered to be unreliable for conventional biometric recognition. However, these noise codes still retain some correlations that are strategically harnessed by our proposed verification protocol. The primary purpose of these codes is to enable the construction of an octet set with an anticipated mismatch rate of approximately 25%.

We design noise codes to capture one or both of the following properties for different types of biometric signals:

- **Unused Feature Regions:** Noise codes can be derived from regions of the biometric data that are typically left out by the feature extraction modules. These regions may not contain highly discriminative information, but they still exhibit some correlation with the original data.

- **Weak Feature Extraction:** Alternatively, noise codes can be generated by employing a feature extractor that is intentionally designed to be weak. Such feature extractors may produce representations that are uncorrelated with the identity vector, yet they retain some degree of correlation within the same region.

Once these noise representations or vectors are obtained, we normalize them using the l_2 norm and subsequently map them to binary values while preserving the cosine distance in the Hamming hypercube. This preservation of cosine distance ensures that the correlations within the noise codes are retained, even in their binarized form.

It's worth noting that noise codes often contain signals that are considered unimportant for recognition. However, in scenarios where weak feature extraction is applied to the same biometric image, there may be a risk of data inference attacks. To mitigate this risk, we introduce a protective measure. We obfuscate the representation with a differentially private output obfuscation technique while still preserving most of the correlational utility. This ensures that even though noise codes are used, the privacy and security of the biometric data are maintained, preventing potential data leakage or inference attacks.

4.4.1 Noise-from-Same-Source-Same-Modality

In this section, we discuss the generation of noise specifically from the same source and modality, focusing on the face. This noise is intended to be used for noise code construction and subsequently incorporated into the verification protocol. The rationale behind this process is to utilize regions of the face that are typically left unused by feature extractors due to their high intra-class variability.

The following points below summarize this approach to obtain the noise representation, illustrated in Figure 4.3:

- **Selection of Unused Regions:** Regions surrounding the face, such as the hair and ears, are often disregarded by feature extractors since these areas exhibit significant variability even within the same identity class. These regions are chosen for noise extraction.

- **Higher-Resolution Eye Features:** To generate noise codes, higher-resolution features from the eye region are extracted directly from the original image. This differs from the feature extractor, which typically uses lower-resolution values after resizing the image.

- **Hair Region Extraction:** For noise representation, features are also extracted from the hair region. Semantic segmentation, implemented using DeepV3Lab+ [17]'s architecture, is employed to identify the hair region. Subsequently, the segmented hair overlay is smoothened with a face mesh obtained from the media-pipe library [55].

- **Concatenation of Features:** The final noise representation is constructed by concatenating the features extracted from both the hair and independent eye regions. We use pretrained imagenet-features from the ResNet-50 [36] model for extracting features.

4.4.2 Noise-from-Different-Source-Same-Modality

In Fingerprint Verification systems, the choice of which fingers to use for analysis is crucial, and typically, fingers with larger surface areas like the index and middle fingers are preferred. This preference stems from the fact that larger fingers offer the potential to capture a greater number of minutiae points, as illustrated in Figure 4.1. Minutiae points serve as key markers for fingerprint analysis, and the quality of a fingerprint image is directly related to the quantity of minutiae points it contains. A higher count of minutiae points increases the likelihood of accurately identifying individuals based on their fingerprints.

However, it's important to note that features extracted from fingerprints of smaller-surface-area fingers can still provide valuable information. These smaller fingers may exhibit correlational properties that satisfy the protocol's constraints effectively. To harness this information, we employ a pretrained DeepPrint model, as detailed in [30], to extract features from the little fingers. These extracted features are then transformed into noise codes, which can be utilized to enhance the robustness of the fingerprint verification process. Meanwhile, the index fingers continue to serve as a source for feature vectors, contributing to the overall accuracy of the system.

4.4.3 Noise-from-Different-Source-Different-Modality

Noise codes can also be extracted from various modalities, and voiceprints, particularly for speaker or person identification, offer another promising avenue. In real-world scenarios, voice recordings often include background disturbances and environmental noise, as illustrated in the spectrogram shown in Figure 4.1. This inherent variability in audio data, especially in unconstrained-length utterances, can pose challenges in extracting discriminative voiceprints. However, these voiceprints can still fulfill the correlational properties required by the protocol.

To address this, we take a two-pronged approach. First, we extract noise codes from the voiceprints obtained in such challenging environments. These noise codes capture the background disturbances and other contextual information, which can be valuable in enhancing the robustness of the identification process.

Secondly, we use fingerprints as the primary feature vectors during the verification process. This approach leverages the well-established uniqueness of fingerprint patterns for identity verification.

To extract speaker representations from utterances, we employ a pretrained ECAPA-TDNN model [26]. This model is well-suited for capturing the distinctive characteristics of speakers, even in the presence of background noise and unconstrained-length utterances.

In summary, while voiceprints can be challenging to extract from in-the-wild audio data, we focus on utilizing noise codes from voiceprints in challenging acoustic environments and rely on fingerprint data as feature vectors for identity verification. This multi-modal approach capitalizes on the strengths of both modalities to achieve robust and accurate speaker/person identification.

4.4.4 Noise Vector Generator

In the previous sections, we discussed various methods for extracting noise codes from both face and fingerprint modalities. However, for improved efficiency during inference and enhanced usability, we employ a dual-headed network that outputs both an embedding/identity vector and a noise vector. Figure 4.5 provides a visual representation of this network architecture designed for the face modality. The Noise vector Generator (NgNet) network is specifically trained to minimize its correlation with the identity vector while maximizing its correlation with augmentations of that identity vector. This unique training approach allows the NgNet head to acquire the capacity to generate a weak feature vector as required by the protocol. To achieve this, we utilize the SimClr [18] framework, leveraging the NT-Xent loss, and additionally minimize the log-cosh loss pertaining to the cosine distance between the identity and noise vectors. The loss function for a given sample x_i is defined as:

$$L_i = -\lambda log \frac{\exp(sim(z_i, z_j)/\tau)}{\sum_{k=1}^{k=2N} 1_{[k \neq j]} \exp(sim(z_i, z_k)/\tau} + log(cosh(sim(e_i, h_i) + \epsilon)) \tag{4.2}$$

In this equation, $1_{[k \neq j]}$ is an indicator function that equals 1 if $k \neq j$, τ represents the temperature parameter, and λ is a hyperparameter used to scale the losses accordingly.

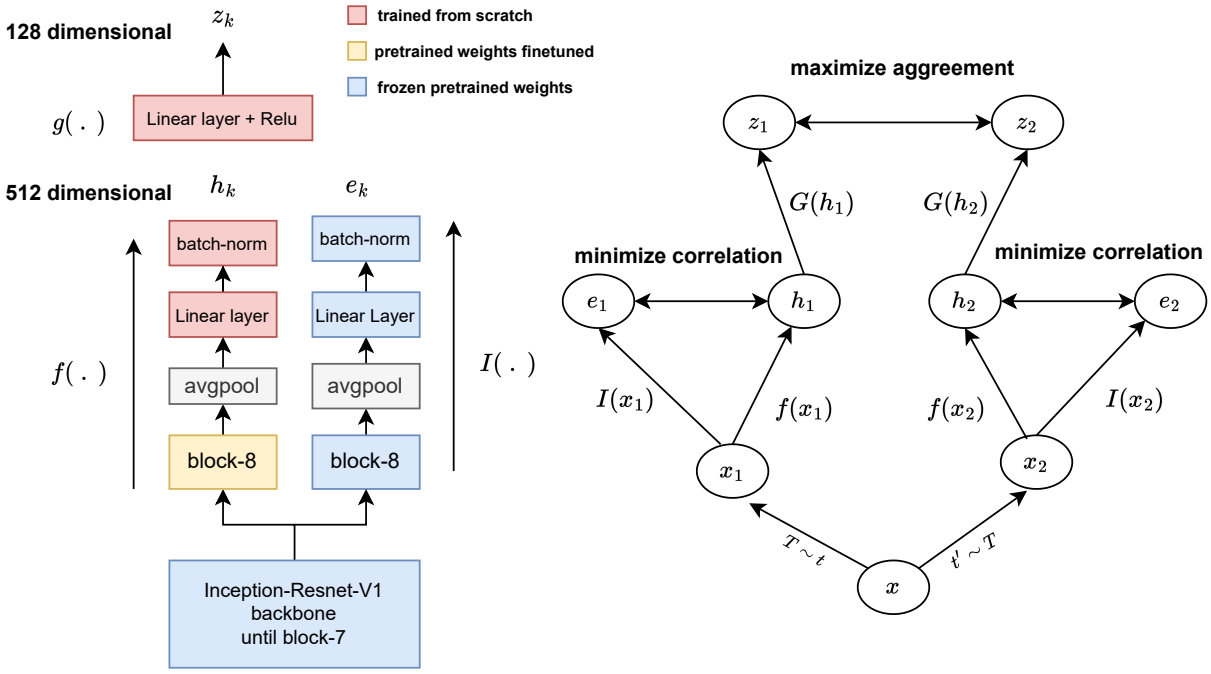

Figure 4.5: Architecture of NgNet, $I(\ .\ )$, $f(\ .\ )$ and $g(\ .\ )$ denote identity mapping, noise vector mapping, and non-linear projection, t and t' are the augmentations of x randomly chosen from a set T

During training, each sample undergoes two data augmentations, incorporating color jitter, horizontal flipping, and Gaussian blur image transformations. We adopt the Inception-ResNet-v1 backbone from Facenet and fine-tune only the last block and a fully-connected layer while freezing the rest of the network. This is done because the majority of task-specific features are generalized within these layers. Both the identity Facenet and NgNet share all layers except the last block and a Linear layer. Additionally, NgNet includes a non-linear projection layer on top of the embedding layer, facilitating its operation within the SimClr framework.

The network is trained on the FFHQ dataset [48] with a larger batch size of 128, which is beneficial in a self-supervised setting. The dimension of the projection is set to 128. Training occurs over 60 epochs with a learning rate of 0.15, a weight decay rate of 0.0001, and a temperature of 0.07, utilizing the Adam optimizer [50].

4.5 Perturbation & Binarization

During octet set creation in the preprocessing phase of the protocol, the correlation between 4-bit pairs is exchanged securely. Although the exact mismatch in the octet is not revealed, it is known from the result to be either (0 or 4-bit mismatch) or (1 or 3-bit mismatch) in the corresponding noise code positions. We adopt one-parameter-defense [81], which perturbs the noise vector y to y', guaranteeing a $(k.\epsilon)$ differential privacy of y using an exponential mechanism [29], where k is the length of the vector and ϵ is the privacy cost. The relative ordering of the elements in y and $y\prime$ remain the same, thereby preserving most of the correlation. We normalize y' (preserved under post-processing [29]) to a unit vector and ϵ is set to 0. The vectors are binarized using CBE-rand (Randomized Circulant Binary Embedding) [82]. The binary embeddings can be obtained by applying a circulant convolution on data point x with r, a random vector independently sampled from the standard normal distribution $N(0, 1)$:

$$b(x) = sign(r \circledast x) = sign(F^{-1}(F(r)\, o\, F(x))) \tag{4.3}$$

where F^{-1} is Inverse Discrete Fourier Transform (IDFT) and F is the Discrete Fourier Transform (DFT). The hamming distance $HD(b\,(x_1), b\,(x_2))$, between two binary codes preserves $1 - cos(\theta)$ [16] in the scale of $[0, 1]$. The generated binary code is of the same dimension as the original vector.

The loss in correlation for the same vector after applying perturbations and CBE-rand is shown in figure 4.6. Ideally, the Hamming distance has to be 0 (same codes); however, because of perturbation and binarization (smaller loss), we observe a correlation loss of ~ 0.10 on average and maximum of ~ 0.25 in the Hamming scale.

4.6 Security & Complexity Analysis

The security of the protocol during data transfer is in line with the security principles outlined in [60]. It operates within a semi-honest security model, which safeguards each party involved in the protocol from an adversary attempting to deduce their input based on the exchanged transcript. Formally, it can

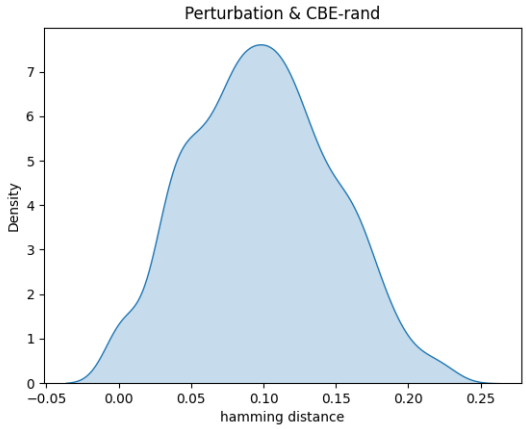

Figure 4.6: Hamming distance distribution between the same output vector after perturbation & binarization

be demonstrated that, from the perspective of a single party, the transcript resulting from the protocol is equally likely to have been generated by any other possible inputs. This property ensures the protocol's security against semi-honest adversaries, even under the assumption of unbounded computational power, thus providing information-theoretic security. To bolster security, one-time padding is applied, and templates are enrolled on the server, effectively extending the framework's protection to both storage and transfer. A formal proof of this security guarantee can be found in the appendix. Regarding the communication cost during the preprocessing phase, it can be calculated as follows: for a noise vector of size n with k repetitions and each element represented by m octets, the cost is $m \times k \times 8 \times (\log n) + 2 \times k \times m$ bits.

For Secure AND operations between two n bit codes, the total communication cost is given by $m \times k \times 8 \times (\log n) + 2 \times k \times m + 4 \times n$ bits. On the other hand, Secure XOR and Secure NOT operations incur no additional communication costs beyond the data size. Lastly, the distribution step involves a cost of n bits. These communication cost estimates provide valuable insights into the resource requirements of the protocol, helping to assess its efficiency and scalability in practice.

4.7 Evaluation Datasets

In our comprehensive evaluation, we explore the performance of various face recognition models across four distinct settings: Same-Source-Same-Modality (SSSM), Different-Source-Same-Modality (DSSM), NgNet, and Different-Source-Different-Modality (DSDM). To conduct these evaluations, we leverage a selection of publicly available datasets.

In the case of SSSM and NgNet, we utilize three prominent datasets: CelebA [54], CFP [71], and LFW [37]. To construct our evaluation sets, we extract 10,000 face pairs from CelebA's evaluation set and CFP's frontal faces (CFP-FP) for both mated and non-mated comparisons. For LFW, we adhere to the specified verification protocol, which entails using 6,000 pairs for evaluation.

For the DSSM scenario, we employ the SOCOFing dataset [72], which comprises 600 subjects, each with 10 different fingers and 4 impressions, including altered images. To create our evaluation set, we generate

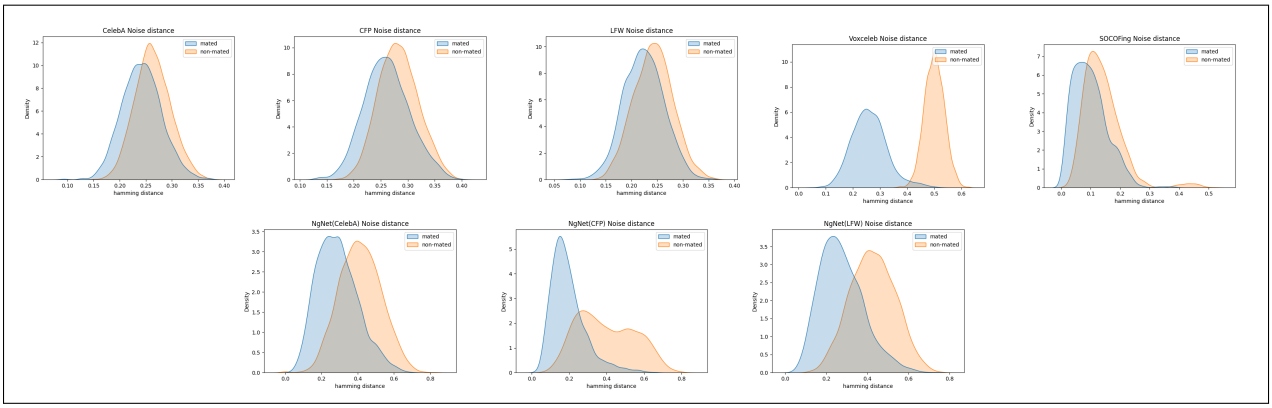

Figure 4.7: Distributions of Hamming distance between noise codes for each dataset

10,000 pairs with the same finger for mated comparisons and different fingers for non-mated comparisons. The index finger is selected for feature vector extraction, while the little finger is employed to introduce noise into the comparisons.

In the context of DSDM, we rely on the FVC2004 DB1A, DB2A, and DB3A datasets [58] and establish correspondence with identities from the Voxceleb1 dataset [62] for utterances, incorporating augmentations of background noise. This allows us to create a comprehensive evaluation framework that spans different sources and modalities for a thorough assessment of face recognition algorithms.

4.8 Empirical Observations on Noise Distributions

Dataset	mated	non-mated
CelebA	0.242	0.263
CFP	0.241	0.282
LFW	0.223	0.262
SOCOFing	0.097	0.139
Voxceleb	0.244	0.409
CelebA - N	0.257	0.413
CFP -N	0.245	0.425
LFW - N	0.194	0.397

Table 4.1: Average noise code Hamming distance for mated vs non-mated pairs, (- N) refers to the distances obtained from NgNet

The noise codes employed in our evaluation serve a crucial purpose in modeling noise ratios and optimizing preprocessing speed. Specifically, our noise codes are designed to achieve an average Hamming distance that

approximates a noise ratio of around 25%. This level of noise ratio has been determined to provide optimal preprocessing speed. However, it's worth noting that noise ratios exceeding 40% may introduce challenges, requiring error correction mechanisms.

In Figure 4.7, we illustrate the distribution of noise codes, showcasing how the noise is distributed across our datasets. Additionally, we present the average Hamming distances between mated and non-mated pairs in Table 4.1. Our findings indicate that, on average, the Hamming distance for mated pairs is below 25%, while for non-mated pairs, it exceeds 25%. This discrepancy is intentional and aligns with our protocol assumptions, as we aim for mated pairs to be around 25% different to facilitate optimal verification without necessitating error corrections.

Here's a summary of our observations in different dataset settings:

- **SSSM (CelebA, CFP, and LFW):** In these datasets, as well as their noise-augmented counterparts (NgNet), and DSDM (Voxceleb), the mean mismatch for mated pairs closely approximates the desired 25% noise ratio.

- **DSSM (SOCOFing):** In the SOCOFing dataset, the mean noise ratio for both mated and non-mated pairs is lower, at around 10% and 14%, respectively. This lower noise ratio can potentially slow down the preprocessing phase. To address this, we apply augmentation techniques such as random Gaussian blur and rotation, as detailed in the supplementary extension, which shifts the distribution to the right.

- **NgNet and DSDM:** In these datasets, the non-mated pairs have an average Hamming distance slightly above 40%. This level of noise may lead to slower computation for non-mated pairs, necessitating error correction mechanisms, which we discuss further in section [fill this].

n conclusion, our noise codes and their distribution play a pivotal role in our evaluation protocol, ensuring that mated pairs have the desired 25% separation for optimal verification while accommodating variations in noise levels across different dataset settings.

4.9 NgNet's Correlational Distribution

In the context of NgNet, it's crucial to ensure that the noise vectors exhibit a property of being uncorrelated with the identity vectors, as described in section 4.4. This property is essential for the effectiveness of the noise modeling.

Dataset	coorelation
CFP	-0.0087
LFW	0.0060
CelebA	-0.0007

Table 4.2: Average correlation between identity and NgNet vectors for each dataset

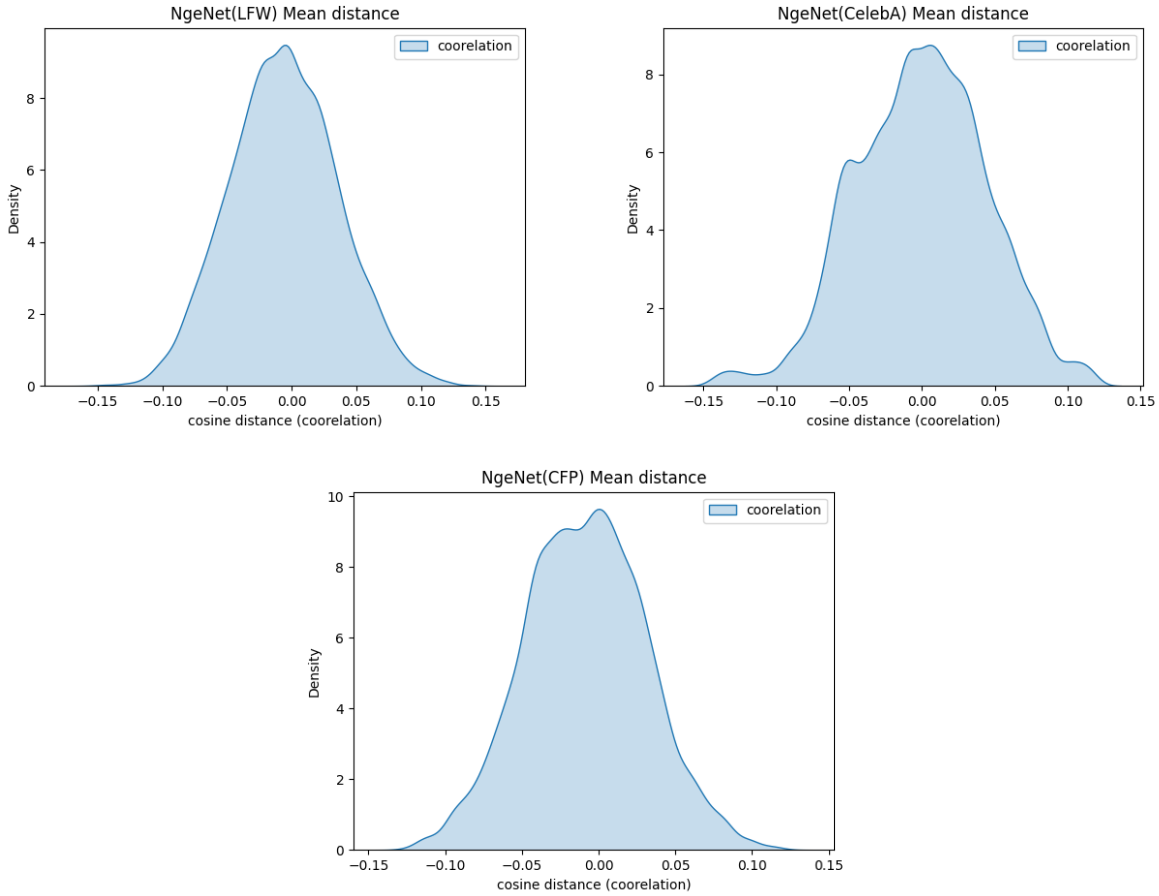

Figure 4.8: Distribution of cosine correlation between NgNet's noise vector and Identity embeddings

From the analysis presented in Table 4.2 and Figure 4.8, it is evident that the correlation of the noise vectors is distributed around 0, and even in the worst-case scenarios, the correlation is close to approximately ± 0.15. This means that the noise vectors are indeed uncorrelated or minimally correlated with the identity vectors, meeting the second property of "noise from the same region."

When we consider the mean correlation across the evaluated datasets (CelebA, CFP, and LFW), which is calculated as $d = (-0.0087 + 0.0060 + -0.0007)/3 = -0.001$, we observe that it is very close to 0. This result is highly acceptable as it indicates a lack of significant correlation between the noise vectors and identity vectors. This lack of correlation ensures that the noise vectors effectively introduce noise into the system without biasing the results based on identity-related information.

In summary, the analysis demonstrates that the noise vectors in NgNet satisfy the requirement of being uncorrelated with identity vectors, supporting the robustness and fairness of the noise modeling process.

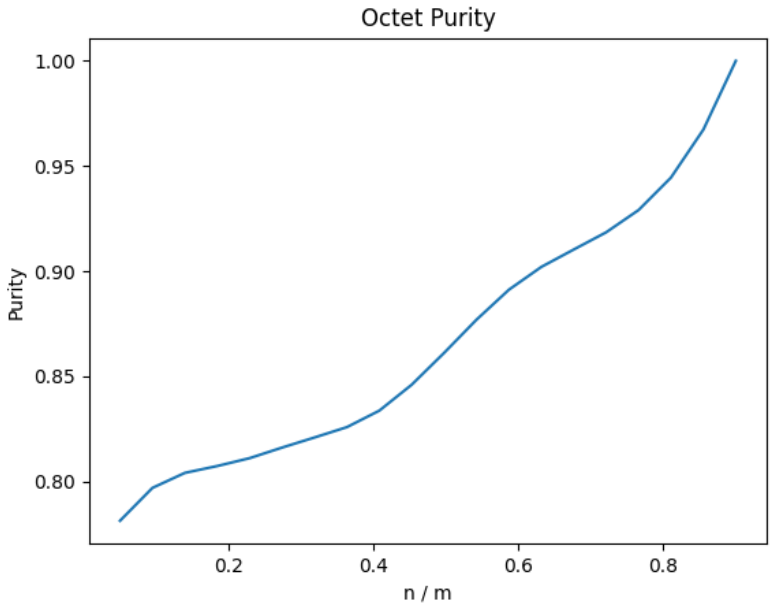

Figure 4.9: Octet purity for parameter ratio n/m

4.10 Protocol's Parameters

The verification protocol involves three key parameters: n, m, and k, which respectively control the preprocessing, distillation, and postprocessing phases. Here's a simplified explanation of their roles and settings:

- n **(Preprocessing Octets):** This parameter determines the number of octets used after preprocessing. We set n to the minimum of $N/4$ and 100, where N represents the size of the noise code.

- m **(Distillation Steps):** m controls the number of steps in the distillation phase. It's set to 90% of the value of n, following a balance between utility and efficiency as suggested in previous work (cite).

- k **(Verification Octets):** k specifies the number of octets needed for the final verification step, and its value depends on the specific requirements of the verification task.

Figure 4.9 demonstrates that when m is set at or above 90% of n, octet purity reaches 100%, indicating that all octets share the same level of mismatch. Deviating from this configuration leads to a gradual decrease in octet purity.

This parameter setup aims to strike a balance between the number of octets used in the protocol and its effectiveness, ensuring efficiency and reliable results.

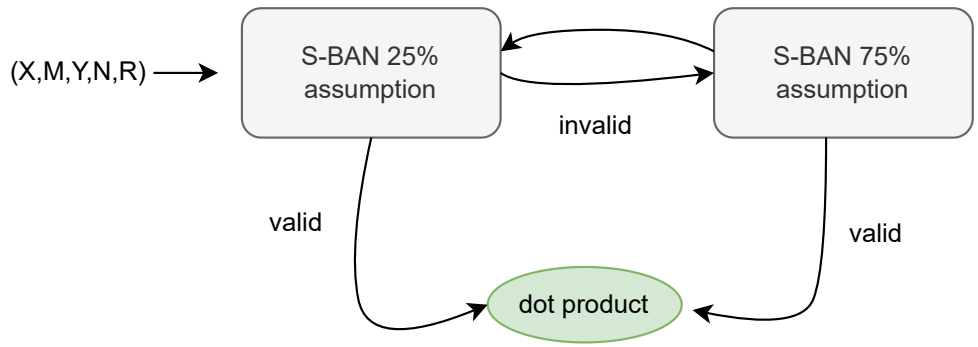

Figure 4.10: Correction Mechanism

4.11 Correction Mechanism

The protocol performs Secure AND computations under the assumption of an octet noise ratio of approximately 25%, corresponding to a 1-bit mismatch. A 3-bit mismatch can cause computation errors by flipping the result, introducing inaccuracies into the dot product computation. While it's possible to construct an octet set with a purity of 1 (no mismatches), it remains uncertain whether the noise ratio is 25% or 75% during verification. To address this uncertainty, we've developed a mechanism to verify the correct Secure AND computation by releasing two resultant shares as $\langle [W_1, S_1], [W_2, S_2] \rangle$. Here:

- $W = W_1 \oplus W_2$ represents the cosine distance.
- $C_1 = W_1 \oplus ra_1$ and $C_2 = W_2 \oplus ra_2$ represent intermediate values.
- $W_e = C_1 \oplus C_2 = S_1 \oplus S_2$ represents the Secure distributed sum of some random distributed vector at each party, effectively protecting octets.

If W_e falls outside the valid range ($W_e < 0$ or $int(W_e) > len(W)$), the distance is marked as invalid. This indicates that some Secure AND computation has flipped.

The size of W_e is set as $4 \times \log(len(W))$ to reliably capture errors. Verification is then repeated with a 75% noise assumption by replacing the AND gate with the NAND gate in the Secure AND scheme of reference [60]. The process alternates between these two assumptions within S-BAN until a valid distance is found, as illustrated in Figure 4.10.

Empirical observations indicate that, with a noise code mismatch of up to 40%, no error correction is necessary after averaging the number of iterations over 100 trials at each noise ratio region, as shown in Table 4.3. The average iterations increase to 2.15 and stabilize at 2 in regions above 60%. In the 50% noise region, more than two iterations may be required, as the octet set mismatch changes at each iteration. As a result, non-mated pairs may exhibit more iterations than mated pairs, especially in the 50% noise region, as illustrated in the noise distributions for Voxceleb and NgNet in Section 4.8.

This mechanism ensures robust verification while adapting to varying noise ratios without the need for excessive error correction.

Noise Ratio Region	Avg. Number of iterations
≤ 0.40	1
0.45	1.1
0.48	1.4
0.50	1.65
0.55	2.15
≥ 0.60	2

Table 4.3: Average Number of iterations in correction mechanism at different noise ratio regions

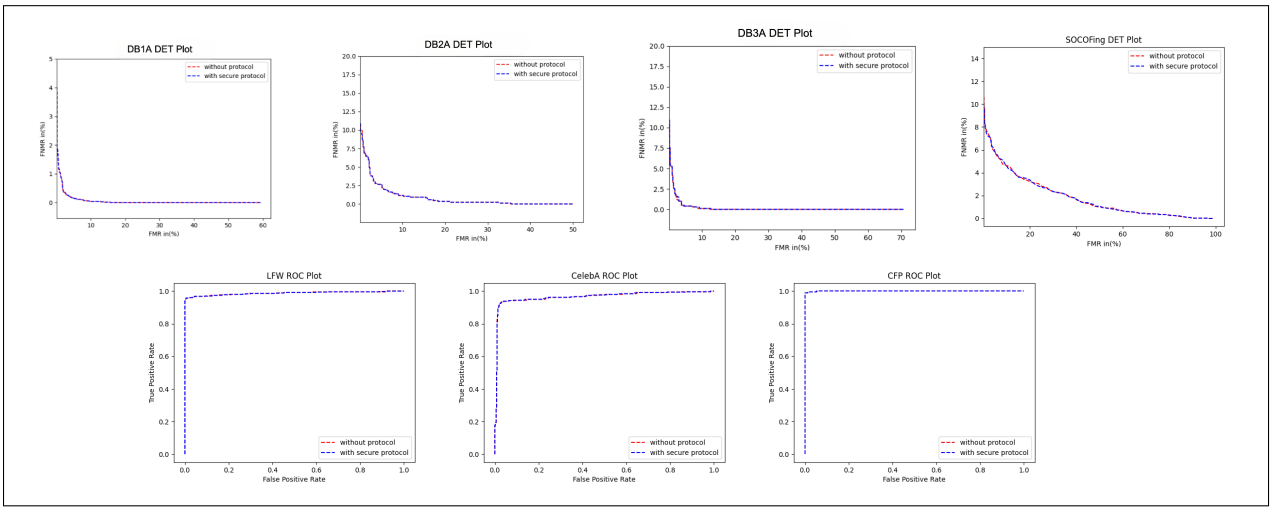

Figure 4.11: DET and ROC plots for each dataset, comparing the deviations after protocol's usage

4.12 Verification Performance

We assess the verification performance of the protocol by comparing the degradation in various verification metrics with and without the protocol's local computation. Specifically, we use False Non-Match Rate (FNMR) @ False Match Rate (FMR) = 0.1%, Equal Error Rate (EER), True Match Rate (TMR) @ False Match Rate (FMR) = 0.1%, and Area Under the Curve (AUC) degradations as performance indicators across different datasets.

In the context of the fingerprint modality, we evaluate the protocol's performance on FVC DB1A, DB2A, and DB3A datasets under the Different-Source-Different-Modality (DSDM) setting and on SOCOFing dataset under the Different-Source-Same-Modality (DSSM) setting. Table 4.4 presents the results, demonstrating low EER and FNMR degradations across the datasets. Specifically, the mean degradation is $d_{eer} = 0.075\%$ and $d_{fnmr} = 0.067\%$.

Upon inspecting the Detection Error Trade-off (DET) curves in Figure 4.11 for each dataset, we observe an almost complete overlap between the local and protocol values. This alignment underscores the effective-

ness of the protocol in maintaining verification performance while preserving privacy and security during computation.

Dataset	FNMR/ FMR=0.1% (local)	FNMR/ FMR=0.1% (protocol)	EER% (local)	EER% (protocol)
DB1A	3.88	4.03	0.30	0.30
DB2A	10.03	9.79	2.65	2.65
DB3A	7.37	7.56	1.19	1.49
SOCOFing	9.73	9.90	3.10	3.10

Table 4.4: Verification Performance on Fingerprint Datasets

Dataset	AUC (local)	AUC (protocol)	TMR/ FMR=0.1% (local)	TMR/ FMR=0.1% (protocol)
CFP	0.999661	0.999653	98.78	98.78
CelebA	0.9660	0.9659	91.16	91.26
LFW	0.9873	0.9872	95.79	95.79

Table 4.5: Verification Performance on Face Datasets

In the context of the face modality, we conduct evaluations using the LFW, CFP, and CelebA datasets under the Same-Source-Same-Modality (SSSM) and NgNet settings. Remarkably, we observe similar degradations in both settings for each dataset.

From the results presented in Table 4.5, it becomes evident that the Area Under the Curve (AUC) and True Match Rate (TMR) degradations are minimal across the datasets, with a mean degradation of $d_{auc} = 0.0051\%$ and $d_{tmr} = 0.05\%$ respectively. The Receiver Operating Characteristic (ROC) curves in Figure 4.11 exhibit a substantial overlap between the local and protocol values.

We attribute these small degradations to the floating-point precision set by the protocol at 8 bits. It's worth noting that higher precision could potentially result in even lower degradations, further underlining the protocol's capability to maintain verification performance while preserving privacy and security during computation.

4.13 Time Cost Analysis

To assess the real-time performance of the protocol, we conducted tests to measure latency and bandwidth during the verification process. The protocol was implemented in Python and run on an Intel Xeon CPU

E5-2640-v4 @ 2.40GHz server with 32GB of RAM. We've provided a summary of the time-bandwidth cost in Table 4.6.

Given that Secure Multi-Party Computation (MPC) protocols typically involve more data transfer, we optimized the protocol by implementing data compression during data transfer and parallelizing secure multiplications (up to 16) to expedite computation.

The results in Table 4.6 illustrate that the time cost exhibits a linear increase as vector sizes grow. Importantly, the bandwidth usage remains below 7MB for practical vector sizes. These findings indicate that the protocol offers real-time performance suitable for practical applications, especially when considering the data compression and parallelization optimizations.

N (size)	Online comparison (s)	Bandwidth (MB)
128	0.771	1.88
192	0.873	2.126
256	0.987	2.395
512	1.455	3.378
1024	2.389	6.602

Table 4.6: Time and memory performance of the protocol with vector size (N)

In addition to the online comparison cost mentioned in Table 4.6, it's important to consider the preprocessing cost, which typically ranges from $200ms$ to $350ms$, depending on the noise ratio regions. The preprocessing includes both offline and online phases. The complete preprocessing (offline) + distillation (D) + post-distillation (PD) cost is minimized at approximately $240ms$ for noise ratios of 25% and 75%, considering a noise code size of 512. Therefore, these noise ratios are particularly desirable, as depicted in Figure 4.12.

The combined cost of D + PD is around $40ms$, except in the 50% noise ratio region, where distillation may need to be repeated multiple times (approximately 1-3 times). This repetition occurs due to octet set cancellation and results in a higher cost in that specific region.

These insights help us understand the trade-offs in terms of processing time and noise ratios, allowing for informed decisions when configuring the protocol for specific applications.

	setup(s)	online(s)	overall(s)
GC [9]	0.423	0.112	0.535
GMW [33]	0.174	0.014	0.188
S-BAN	0.063	0.020	**0.083**

Table 4.7: Latency comparison with other methods for each MULT operation

The proposed method is exact, meaning that its accuracy is limited only by the precision level specified in the protocol. To assess its efficiency, we conducted a time-cost comparison relative to other popular Secure

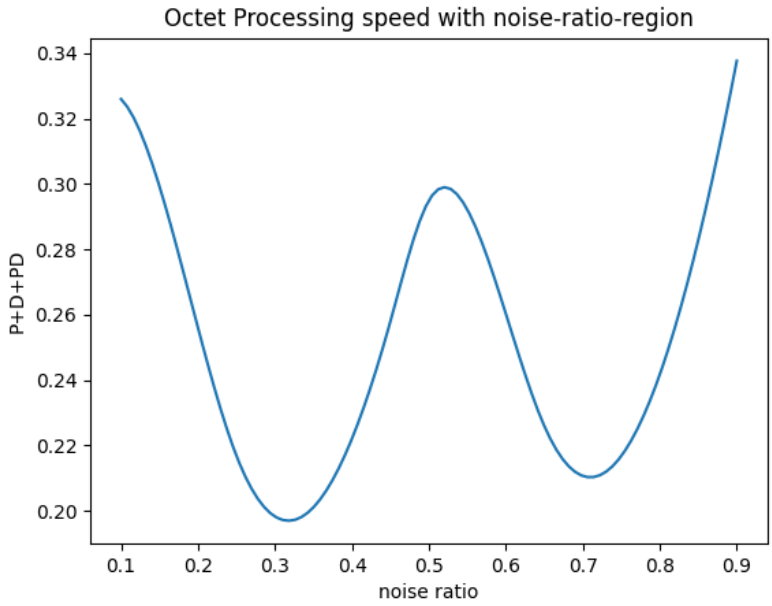

Figure 4.12: Comparing octet processing speed at different noise ratio regions, P+D+PD refers to the time taken for preprocessing + distillation + post-distillation phases

Multi-Party Computation (SMC) boolean circuit evaluators that share the same security assumptions. This analysis was carried out on a single thread. We utilized the scores reported in [34] and leveraged ABY's [25] implementation for a 32-bit precision MULT operation, which is a fundamental component for the dot product.

The results, as detailed in Table 4.7, indicate that our protocol exhibits lower setup and overall time when compared to other SMC boolean circuit evaluators. Notably, the online phase of our protocol is comparable to that of GMW [33], with a slight increase in computation time as additional AND gates are computed for the encrypted inputs.

This comparison highlights the efficiency and effectiveness of our proposed method, positioning it favorably among existing SMC boolean circuit evaluators while maintaining a high level of security and precision.

Chapter 5

Conclusions & Future Work

In this work, we introduced novel verification schemes that incorporate noise through secure two-party computation. The time complexity across various modalities, even for practical vector sizes, remains below $2s$, with a bandwidth requirement of approximately $\sim 3Mb$, making it well-suited for practical applications. Additionally, the degradation across modalities is less than $\sim 0.05\%$ at higher thresholds. Furthermore, the protocol's provision of information-theoretic security ensures a high level of confidence in user authentication. Looking forward to future research, the protocol could be extended by integrating zero-knowledge proofs to enhance defense against malicious adversaries. Moreover, exploring the development of efficient computation schemes that enable operations directly within a single circuit could be a promising avenue for further improvement.

Appendix A

Correctness & Security Proof

We provide the correctness and the security proof of the Secure AND for its Computation phase, algorithm 7, which is used in computing Secure AND, extend it to S-BAN and present its different aspects & results.

Theorem 1. The equality $x \wedge y = z_1 \oplus z_2$ holds when the noise ratio (25%) given by :

$$||r_0 \oplus s_0|| + ||r_1 \oplus s_1|| + ||r_2 \oplus s_2|| + ||r_3 \oplus s_3|| = 1$$

Proof. By expanding the results after applying the AND gate (Eq. 4.2 and 4.3) on the octet we get : $a1 = r2 \oplus r3$, $b1 = r1 \oplus r3$, $c1 = r3$, and $a2 = s2 \oplus s3$, $b2 = s1 \oplus s3$ and $c2 = s3$. The operation $a1 \oplus a2$ yields 1 for either $r2 \oplus s2 = 1$ or $r3 \oplus s3 = 1$, a 1-bit mismatch at position 3 or 4. Similar observation for 1-bit mismatch at position 2 or 4 holds for $b1 \oplus b2$. $c1 \oplus c2$ yields 1 when the mismatch is at position 4.

Let the intermediate variables A, B, C hold the results of the XORs :

$$A = a1 \oplus a2, \;\; B = b1 \oplus b2, \;\; C = c1 \oplus c2 \tag{A.1}$$

We intend to show that result of $z_1 \oplus z_2$ at 25% noise is equal to the $R.H.S$ of :

$$x \wedge y = (x_1 \oplus x_2) \wedge (y_1 \oplus y_2) = (x_1 \wedge y_1) \oplus (x_2 \wedge y_1) \oplus (x_1 \wedge y_2) \oplus (x_2 \wedge y_2) \tag{A.2}$$

Let $M_1 = (x_1 \wedge y_1), M_2 = (x_2 \wedge y_1), M_3 = (x_1 \wedge y_2), M_4 = (x_2 \wedge y_2)$

Eq. A.2 can be reduced to :

$$x \wedge y = M_1 \oplus M_2 \oplus M_3 \oplus M_4 \tag{A.3}$$

After Substituting X_A and X_B in $z1$ and $z2$ we get :

$$\begin{aligned} z_1 \oplus z_2 = ((x_1 \oplus x_2 \oplus A) \wedge (y_1 \oplus y_2 \oplus B)) \\ \oplus (x_1 \wedge (y_1 \oplus y_2 \oplus B)) \\ \oplus (y_1 \wedge (x_1 \oplus x_2 \oplus A)) \\ \oplus (x_2 \wedge (y_1 \oplus y_2 \oplus B)) \\ \oplus (y_2 \wedge (x_1 \oplus x_2 \oplus A)) \oplus C \end{aligned} \tag{A.4}$$

The simplified form of Eq. A.4 is :

$$\begin{aligned} z_1 \oplus z_2 = M_1 \oplus M_2 \oplus M_3 \oplus M_4 \\ \oplus (A \wedge B) \oplus A \wedge (y_1 \oplus y_2) \\ \oplus B \wedge (x_1 \oplus x_2) \oplus M_1 \oplus M_3 \oplus B \wedge (x_1 \oplus x_2) \\ \oplus A \wedge (y_1 \oplus y_2) \oplus M_1 \oplus M_2 \oplus M_2 \\ \oplus M_4 \oplus M_3 \oplus M_4 \oplus C \end{aligned} \tag{A.5}$$

$$\begin{aligned} = M_1 \oplus M_2 \oplus M_3 \oplus M_4 \\ \oplus A \wedge (y_1 \oplus y_2) \oplus B \wedge (x_1 \oplus x_2) \\ \oplus B \wedge (x_1 \oplus x_2) \oplus A \wedge (y_1 \oplus y_2) \\ \oplus C \oplus (A \wedge B) \end{aligned}$$

- First position mismatch : $||r_0 \oplus s_0|| = 1 \implies A = 0, B = 0, C = 0$

$$z1 \oplus z2 = M_1 \oplus M_2 \oplus M_3 \oplus M_4 = x \wedge y \tag{A.6}$$

- Second position mismatch : $||r_1 \oplus s_1|| = 1 \implies A = 0, B = 1, C = 0$

$$\begin{aligned} z1 \oplus z2 = M_1 \oplus M_2 \oplus M_3 \\ \oplus M_4 \oplus (x_1 \oplus x_2) \oplus (x_1 \oplus x_2) \\ = M_1 \oplus M_2 \oplus M_3 \oplus M_4 = x \wedge y \end{aligned} \tag{A.7}$$

- Third position mismatch : $||r_2 \oplus s_2|| = 1 \implies A = 1, B = 0, C = 0$

$$\begin{aligned} z1 \oplus z2 = M_1 \oplus M_2 \oplus M_3 \\ \oplus M_4 \oplus (y_1 \oplus y_2) \oplus (y_1 \oplus y_2) \\ = M_1 \oplus M_2 \oplus M_3 \oplus M_4 = x \wedge y \end{aligned} \tag{A.8}$$

- Fourth position mismatch : $||r_3 \oplus s_3|| = 1 \implies A = 1, B = 1, C = 1$

$$\begin{aligned} z1 \oplus z2 &= M_1 \oplus M_2 \oplus M_3 \oplus M_4 \oplus 1 \oplus 1 \\ &= M_1 \oplus M_2 \oplus M_3 \oplus M_4 = x \wedge y \end{aligned} \tag{A.9}$$

■

Theorem 2. Assuming that the noise ratio is 25%, the Algorithm 7 above is secure.

Proof. First, we argue that the assumption is valid in the scenario described in the paper [60]. Since the average noise ratio for a mated pair ranges from 0-25%, we argue that adversary. The noise ratio is 25% because we consider an octet. In an octet, there are 4 two bit pairs. This makes only 6 cases possible i.e. no pairs are noisy, 1/2/3/4 pairs are noisy. Early XOR removes the 0, 2, 4 noisy pair cases. We are left with only 1 out of 4 or 3 out of 4 noisy pairs. Since the average noise ratio is 0-25%, majority of the octets in distillation process are correct. This makes the entire protocol to be run with 1 out of 4 noisy pair.

According to the algorithm above, each computation is described as $F(x, y) = z$. We denote the corrupt party using set $\mathcal{C}$. It is easy to see that if both parties are corrupt by same adversary, no protocol can provide any privacy since there is no honest party to protect. Hence, it makes sense to only argue the security in terms of privacy against one of the parties. Further, we consider the privacy against a corrupt P_1. Without the loss of generality, the following argument works equally for P_2 as well. We follow the simulation paradigm to argue the privacy of the computation phase in general. This involves input sharing between two parties, XOR computation, AND computation and output reconstruction. The rest of the phases such as preprocessing, distillation are tackled later in the section.

To argue the privacy of the computation phase, we want to show that the corrupt party P_1 only learns some limited information. We can show that by arguing everything that a corrupt party sees during our protocol execution can be efficiently simulated given only the information available to this corrupt party. Specifically, we will construct a $view_1$ and $view_2$ vectors as the values party P_1 and P_2 see during the protocol execution respectively. Privacy against party P_1 means that there exists a polynomial-time probabilistic simulator $\mathcal{S}$, with input of party P_1 and the output of the protocol, which can output a view that has same distribution as the view of corrupt P_1.

$$\mathcal{S}(\{x, z\}_{P_1 \in \mathcal{C}}) \equiv \{view_1\}_{P_1 \in \mathcal{C}}$$

Similarly privacy against corrupt P_2 can be stated as

$$\mathcal{S}(\{y, z\}_{P_2 \in \mathcal{C}}) \equiv \{view_2\}_{P_2 \in \mathcal{C}}$$

Intuitively, the simulator runs the corrupt party P_1 on it's own input. It simulates the messages from honest parties by uniformly sampling honest party P_2 value which produces the right distribution and it computes honest party P_2's share using z and z_1 to be consistent with it. According to the simulation paradigm, the corrupt party only learns its own input and output bit, using which it can efficiently simulate everything it sees(view).

We define the $view_1$ as a vector which includes values as follows: $\{x, x_1, x_2, a_1, b_1, c_1, x_2 \oplus a_2, y_2 \oplus b_2, z_1, z_2\}$. We also define 2 functions $strip$ and $dress$ as follows:

- $strip(view_1)$ is a function which takes the view of P_1 as input and removes z_2 from it. Hence, $strip(view_1) = \{x, x_1, x_2, a_1, b_1, c_1, x_2 \oplus a_2, y_2 \oplus b_2, z_1\}$
- $dress(view_1')$ is a function which takes a stripped view of P_1 which is of form $\{x, x_1, x_2, a_1, b_1, c_1, x_2 \oplus a_2, y_2 \oplus b_2, z_1\}$ as input and outputs $view_1''$ which is of form $\{x, x_1, x_2, a_1, b_1, c_1, x_2 \oplus a_2, y_2 \oplus b_2, z_1, z_2\}$. In short, $dress$ function tries to fill up the values which $strip$ removed. It does so by first assuming that the output is, say, z and computes what P_2 holds using z and z_1 by $z_2 = z \oplus z_1$. It adds this new z_2 to the $view_1'$ to convert into $view_1''$

Now, we define the simulator $\mathcal{S}$ as follows:

1. Consider the input to the simulator $\mathcal{S}$ as $\{x, z\}$
2. Define a input tuple as (x, y) where $y = 0$ i.e. $(x, 0)$.
3. Run the Computation Phase as described in the general sense.(input sharing, XOR, AND, and output reconstruction with $(x, 0)$ as input bits. Produce $view_1^0$ to be view of party P_1 as defined.
4. Let z be the value received as input. Using z, Output $dress(\{strip(view_1^0)\})$

First, we show that $\{strip(view_1^0\} \equiv \{strip(view_1)\}$. In general, we want to show that $\{strip(view_1^0\} \equiv \{strip(view_1^1)\}$. This can be done using observation that P_1 receives y_1 in input sharing phase, nothing in XOR computation and $x_2 \oplus a_2, y_2 \oplus b_2$ in AND computation. For both cases $y = 0$ and $y = 1$, P_1 distributes the same x and P_2 distributes different y. However, P_1 only sees y_1 which is uniformly distributed in both cases. In multiplication phase, P_1 gets to know a_2 since it already knows x_2. However, a_2 is uniformly distributed based on the fact that a_2 relies on s_0, s_1, s_2, s_3 which are independent of each other. Hence, a_2 is uniformly distributed in both cases. This can also be verified using truth table. Same applies for b_2. P_1 also knows $y_2 \oplus b_2$ which is also uniformly distributed since both y_2 and b_2 are uniformly distributed. For output reconstruction, P_1 only knows z_1 since the view is stripped. This z_1 was

computed prior to output reconstruction phase, hence, can be considered identically distributed for both cases $y = 0, y = 1$.

Now, using correctness of the protocol under the noise assumption in previous theorem, we know that $dress(\{strip(view_1)\}) \equiv \{view_1\}$, This is due to the fact that for a given input say x, y there exists only one output z. Hence, the $dress$ function will have identical distribution as the original $view_1$. This shows that $\mathcal{S}(\{x, z\}) = dress(\{strip(view_1^0)\}) \equiv dress(\{strip(view_1)\}) \equiv \{view_1\}$

This security proof can be further extended to the entire protocol using hybrid argument. Intuitively, we will construct N hybrids where N is the number of secure AND computations such that on j^{th} hybrid, all AND computations before the j^{th} AND computations are replaced by the output of the above simulator and all AND computations after the j^{th} AND computations are same as Hybrid 1 i.e. the view of party P_1. The last hybrid will have all the secure AND computations replaced with their simulator outputs. This proves the security of the entire protocol.

■

Corollary 1. Since, Algorithm 7 is secure, the verification protocol in S-BAN is also secure as it uses only Secure AND , Secure XOR and Secure NOT operations.

We define differential privacy, sensitivity, exponential mechanism, and the algorithm used in perturbing the output vector y before binarization.

Defination 1. A mechanism M provides ϵ-differential privacy for any pair of neighboring datasets D and D', and for every set of outcomes O, if M satisfies :

$$P(M(D \in O)) \leq e^{\epsilon} P(M(D' \in O)) \tag{A.10}$$

Algorithm 11 Output Obfuscation

1: Let $y = \langle y_1, y_2....y_k \rangle$ be the output vector

2: Sort y in increasing order : $y_1 \leq y_2 \leq .....y_k$

3: Divide it into k subranges as follows:

4: $R = (-1, (y_1 + y_2)/2), [(y_1 + y_2)/2, (y_2 + y_3)/2)....[(y_k + y_{k-1})/2, 1)$

5: Set a value by randomly selecting from each R_i and assign to y_i position using an exponential mechanism. In our implementation we use $\epsilon = 0$ (equivalent to random sample), formally it is sampled from probability proportional to $e^{(\epsilon u_j^i(D,r)/2\Delta u)}$, where $\Delta u = 2$ for unit vectors and $u_j^i = \frac{1}{|y_i - (i-1)/k + (j-1)p)|}$ [81].

6: return the obfuscated vector y_o

Defination 2. Sensitivity of a query is defined for a query $f : D \rightarrow R$:

$$\Delta S = max_{D,D'}||f(D) - f(D')|| \tag{A.11}$$

Defination 3. The exponential mechanism selects and outputs a score $r \in R$ with probability proportional to $e^{(\epsilon u(D,r)/2\Delta u)}$, where ϵ is the privacy budget, u(D, r) is the utility of a dataset and output pair, and $\Delta u = max_{D,D',r}|u(D, r) - u(D', r)|$ is the sensitivity of the utility score.

Algorithm for modification of a vector y using the above definitions [81]:

It is easy to see that, the Algorithm 11 satisfies $k\epsilon$ Differential privacy (we choose k samplings) i.e when $y'_i + y'_{i+1} = y_i + y_{i+1}$ and $||y - y'||_2 \leq 1$ is interpreted as a neighboring dataset having the same range R :

$$\frac{P(r_j|y)}{P(r_j|y')} = \frac{exp(\epsilon u_{i,y,j}/2\Delta u)/\sum_k exp(\epsilon u_{i,y,k}/2\Delta u)}{exp(\epsilon u_{i,y',j}/2\Delta u)/\sum_k exp(\epsilon u_{i,y',k}/2\Delta u)} \leq exp(\epsilon) \tag{A.12}$$

Related Publications

- SIAN: Secure Iris Authentication using Noise
  Praguna Manvi,Achintya Desai, Kannan Srinathan, Anoop Namboodiri, *International Joint Conference on Biometrics (IJCB)*, 10-13 October 2022, Abu Dhabi, United Arab Emirates
- S-BAN: Secure Biometric Authentication using Noise
  Praguna Manvi,Achintya Desai, Kannan Srinathan, Anoop Namboodiri, *Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP'23)*, December 2023, Ropar, India

Bibliography

[1] ISO/IEC JTC1 SC27 Security Techniques: ISO/IEC 24745:2022. Information Technology - Security Techniques - Biometric Information Protection, 2022.

[2] Palmprint Databases. `http://www.cbsr.ia.ac.cn/english/Palmprint%20Databases.asp`.

[3] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti. A survey on homomorphic encryption schemes: Theory and implementation. *ACM Computing Surveys (Csur)*, 51(4):1–35, 2018.

[4] T. Araki, A. Barak, J. Furukawa, M. Keller, Y. Lindell, K. Ohara, and H. Tsuchida. Generalizing the spdz compiler for other protocols. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, page 880–895, New York, NY, USA, 2018. Association for Computing Machinery.

[5] S. Arar. Fixed-point representation: The q format and addition examples. *All About Circuits*, November 2017.

[6] M. Barni, G. Droandi, R. Lazzeretti, and T. Pignata. Semba: secure multi-biometric authentication. *IET Biometrics*, 8(6):411–421, 2019.

[7] P. Bauspieß, J. Kolberg, D. Demmler, J. Krämer, and C. Busch. Post-quantum secure two-party computation for iris biometric template protection. In *2020 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6, 2020.

[8] D. Beaver. Efficient multiparty protocols using circuit randomization. volume 576, pages 420–432, 08 1991.

[9] M. Bellare, V. T. Hoang, and P. Rogaway. Foundations of garbled circuits. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 784–796, 2012.

[10] M. Blanton and P. Gasti. Secure and efficient protocols for iris and fingerprint identification. In V. Atluri and C. Diaz, editors, *Computer Security – ESORICS 2011*, pages 190–209, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[11] M. Blanton and P. Gasti. Secure and efficient protocols for iris and fingerprint identification. In *Computer Security–ESORICS 2011: 16th European Symposium on Research in Computer Security, Leuven, Belgium, September 12-14, 2011. Proceedings 16*, pages 190–209. Springer, 2011.

[12] D. Bobeldyk and A. Ross. Predicting soft biometric attributes from 30 pixels: A case study in nir ocular images. In *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pages 116–124, 2019.

[13] J. Bringer, H. Chabanne, M. Favre, A. Patey, T. Schneider, and M. Zohner. Gshade: faster privacy-preserving distance computation and biometric identification. In *Proceedings of the 2nd ACM workshop on Information hiding and multimedia security*, pages 187–198, 2014.

[14] J. Bringer, M. Favre, H. Chabanne, and A. Patey. Faster secure computation for biometric identification using filtering. In *2012 5th IAPR International Conference on Biometrics (ICB)*, pages 257–264, 2012.

[15] P. Campisi. Security and privacy in biometrics: towards a holistic approach. In *Security and privacy in biometrics*, pages 1–23. Springer, 2013.

[16] M. S. Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 380–388, 2002.

[17] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *CoRR*, abs/1802.02611, 2018.

[18] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[19] R. Cramer, I. B. Damgård, et al. *Secure multiparty computation*. Cambridge University Press, 2015.

[20] I. Damgard, M. Geisler, and M. Kroigard. Homomorphic encryption and secure comparison. *Int. J. Appl. Cryptol.*, 1(1):22–31, feb 2008.

[21] I. Damgård, V. Pastro, N. Smart, and S. Zakarias. Multiparty computation from somewhat homomorphic encryption. In *Annual Cryptology Conference*, pages 643–662. Springer, 2012.

[22] J. Daugman. Complete discrete 2-d gabor transforms by neural networks for image analysis and compression. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(7):1169–1179, 1988.

[23] J. Daugman. High confidence visual recognition of persons by a test of statistical independence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1148–1161, 1993.

[24] J. Daugman. How iris recognition works. In *The essential guide to image processing*, pages 715–739. Elsevier, 2009.

[25] D. Demmler, T. Schneider, and M. Zohner. Aby-a framework for efficient mixed-protocol secure two-party computation. In *NDSS*, 2015.

[26] B. Desplanques, J. Thienpondt, and K. Demuynck. Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. *arXiv preprint arXiv:2005.07143*, 2020.

[27] W. Diffie and M. E. Hellman. Privacy and authentication: An introduction to cryptography. *Proceedings of the IEEE*, 67(3):397–427, 1979.

[28] W. Du and M. J. Atallah. Secure multi-party computation problems and their applications: A review and open problems. In *Proceedings of the 2001 Workshop on New Security Paradigms*, NSPW '01, page 13–22, New York, NY, USA, 2001. Association for Computing Machinery.

[29] C. Dwork, A. Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[30] J. J. Engelsma, K. Cao, and A. K. Jain. Learning a fixed-length fingerprint representation, 2019.

[31] D. Evans, Y. Huang, J. Katz, and L. Malka. Efficient privacy-preserving biometric identification. In *Proceedings of the 17th conference Network and Distributed System Security Symposium, NDSS*, volume 68, pages 90–98, 2011.

[32] O. Goldreich. *Foundations of Cryptography*, volume 1. Cambridge University Press, 2001.

[33] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game, or a completeness theorem for protocols with honest majority. In *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*, pages 307–328. 2019.

[34] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game, or a completeness theorem for protocols with honest majority. In *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*, pages 307–328. 2019.

[35] M. Gomez-Barrero, E. Maiorana, J. Galbally, P. Campisi, and J. Fierrez. Multi-biometric template protection based on homomorphic encryption. *Pattern Recognition*, 67:149–163, 2017.

[36] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[37] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[38] J. Illingworth and J. Kittler. A survey of the hough transform. *Computer Vision, Graphics, and Image Processing*, 44(1):87–116, 1988.

[39] A. Jain, S. Prabhakar, L. Hong, and S. Pankanti. Fingercode: a filterbank for fingerprint representation and matching. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, volume 2, pages 187–193 Vol. 2, 1999.

[40] A. Jain, A. Ross, and S. Prabhakar. An introduction to biometric recognition. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14:4 – 20, 02 2004.

[41] A. K. Jain, S. C. Dass, and K. Nandakumar. Soft biometric traits for personal recognition systems. In D. Zhang and A. K. Jain, editors, *Biometric Authentication*, pages 731–738, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[42] A. K. Jain, D. Deb, and J. J. Engelsma. Biometrics: Trust, but verify. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 4(3):303–323, 2021.

[43] A. K. Jain, P. Flynn, and A. A. Ross. *Handbook of biometrics*. Springer Science & Business Media, 2007.

[44] A. K. Jain, S. Prabhakar, L. Hong, and S. Pankanti. Fingercode: a filterbank for fingerprint representation and matching. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, volume 2, pages 187–193. IEEE, 1999.

[45] A. K. Jain, S. Prabhakar, L. Hong, and S. Pankanti. Filterbank-based fingerprint matching. *IEEE transactions on Image Processing*, 9(5):846–859, 2000.

[46] A. K. Jain, A. Ross, and S. Prabhakar. An introduction to biometric recognition. *IEEE Transactions on circuits and systems for video technology*, 14(1):4–20, 2004.

[47] A. K. Jain, A. Ross, and U. Uludag. Biometric template security: Challenges and solutions. In *2005 13th European signal processing conference*, pages 1–4. IEEE, 2005.

[48] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks, 2019.

[49] P. Kaur, N. Kumar, and M. Singh. Biometric cryptosystems: a comprehensive survey. *Multimedia Tools and Applications*, 82(11):16635–16690, 2023.

[50] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[51] A. Kumar. IIT Delhi Iris Database (Version 1.0) [online] available: https://www4.comp.polyu.edu.hk/ csajaykr/.

[52] A. Kumar and A. Passi. Comparison and combination of iris matchers for reliable personal authentication. *Pattern Recognition*, 43(3):1016–1026, 2010.

[53] Y.-h. Li, M. Savvides, and T. Chen. Investigating useful and distinguishing features around the eyelash region. In *2008 37th IEEE Applied Imagery Pattern Recognition Workshop*, pages 1–6. IEEE, 2008.

[54] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[55] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C. Chang, M. G. Yong, J. Lee, W. Chang, W. Hua, M. Georg, and M. Grundmann. Mediapipe: A framework for building perception pipelines. *CoRR*, abs/1906.08172, 2019.

[56] X. Luo, J. Tian, and Y. Wu. A minutiae matching algorithm in fingerprint verification. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, volume 4, pages 833–836. IEEE, 2000.

[57] Y. Luo, S.-c. S. Cheung, T. Pignata, R. Lazzeretti, and M. Barni. An efficient protocol for private iris-code matching by means of garbled circuits. In *2012 19th IEEE International Conference on Image Processing*, pages 2653–2656, 2012.

[58] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain. Fvc2004: Third fingerprint verification competition. In D. Zhang and A. K. Jain, editors, *Biometric Authentication*, pages 1–7, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[59] Manisha and N. Kumar. Cancelable biometrics: A comprehensive survey. *Artificial Intelligence Review*, 53:3403–3446, 2020.

[60] P. Manvi, A. Desai, K. Srinathan, and A. Namboodiri. Sian: Secure iris authentication using noise. In *2022 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–9, 2022.

[61] S. Minaee, A. Abdolrashidi, H. Su, M. Bennamoun, and D. Zhang. Biometrics recognition using deep learning: A survey. *Artificial Intelligence Review*, pages 1–49, 2023.

[62] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman. Voxceleb: Large-scale speaker verification in the wild. *Computer Science and Language*, 2019.

[63] C. A. of Sciences. Gait Databases, Year.

[64] L. O'Gorman. Comparing passwords, tokens, and biometrics for user authentication. *Proceedings of the IEEE*, 91(12):2021–2040, 2003.

[65] M. Osadchy, B. Pinkas, A. Jarrous, and B. Moskovich. Scifi - a system for secure face identification. In *2010 IEEE Symposium on Security and Privacy*, pages 239–254, 2010.

[66] V. M. Patel, N. K. Ratha, and R. Chellappa. Cancelable biometrics: A review. *IEEE signal processing magazine*, 32(5):54–65, 2015.

[67] H. Proença and L. A. Alexandre. Ubiris: A noisy iris image database. In *Proceed. of ICIAP 2005 - Intern. Confer. on Image Analysis and Processing*, volume 1, pages 970–977, 2005.

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

[69] A.-R. Sadeghi, T. Schneider, and I. Wehrenberg. Efficient privacy-preserving face recognition. In *Information, Security and Cryptology–ICISC 2009: 12th International Conference, Seoul, Korea, December 2-4, 2009, Revised Selected Papers 12*, pages 229–244. Springer, 2010.

[70] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015.

[71] S. Sengupta, J.-C. Chen, C. Castillo, V. M. Patel, R. Chellappa, and D. W. Jacobs. Frontal to profile face verification in the wild. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9, 2016.

[72] Y. I. Shehu, A. Ruiz-Garcia, V. Palade, and A. E. James. Sokoto coventry fingerprint dataset. *CoRR*, abs/1807.10609, 2018.

[73] T. Tan. Casia Iris Image Dataset V3 [online] available: http://biometrics.idealtest.org.

[74] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991.

[75] P. Tuyls, B. Skoric, and T. Keevenaar. On private biometrics, secure key storage and anti-counterfeiting, 2007.

[76] M. University. MMU Iris Image Database [online] available: http://pesona.mmu.edu.my/ ccteo/.

[77] A. Utzhanova. Fingerprint technology and sustainable development. *European Journal of Sustainable Development*, 5(4):325–325, 2016.

[78] J. Wayman, A. Jain, D. Maltoni, and D. Maio. *An Introduction to Biometric Authentication Systems*, pages 1–20. Springer London, London, 2005.

[79] R. Wildes, J. Asmuth, G. Green, S. Hsu, R. Kolczynski, J. Matey, and S. McBride. A system for automated iris recognition. In *Proceedings of 1994 IEEE Workshop on Applications of Computer Vision*, pages 121–128, 1994.

[80] J. Yan, A. Blackwell, R. Anderson, and A. Grant. Password memorability and security: empirical results. *IEEE Security Privacy*, 2(5):25–31, 2004.

[81] D. Ye, S. Shen, T. Zhu, B. Liu, and W. Zhou. One parameter defense—defending against data inference attacks via differential privacy. *IEEE Transactions on Information Forensics and Security*, 17:1466–1480, 2022.

[82] F. Yu, S. Kumar, Y. Gong, and S.-F. Chang. Circulant binary embedding. In *International conference on machine learning*, pages 946–954. PMLR, 2014.

[83] C. Zhao, S. Zhao, M. Zhao, Z. Chen, C.-Z. Gao, H. Li, and Y.-a. Tan. Secure multi-party computation: theory, practice and applications. *Information Sciences*, 476:357–372, 2019.