Privacy preserving biometric authentication

Master Thesis

Master in *Sciences and Technologies*,
Specialty in *Computer Science*,
Track *Cryptology and Computer Security*.

Author
Bousseaden Anass <anassbousseaden@yahoo.fr>

Supervisor
Zemor gilles <zemor@math.u-bordeaux.fr>

Tutor
Kalpana Singh <kalpana.singh@worldline.com>

December 12, 2023

Declaration of authorship of the document

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of the Master in *Sciences and Technologies*, Specialty in *Mathematics* or *Computer Science*, Track *Cryptology and Computer Security*, is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Date and Signature

Contents

Introduction

In an era where personal devices like smartphones have seamlessly woven into the fabric of our daily lives, the need for secure yet user-friendly authentication methods has reached paramount importance. Biometric authentication, which harnesses unique individual traits such as facial features or palm prints, has emerged as the forefront solution in this landscape. With its inherent convenience and effortless user experience, biometric authentication has eclipsed traditional methods like passwords and PIN codes. This thesis delves into the realm of biometric authentication, particularly focusing on external-facing solutions compatible with deep learning techniques.

Numerous research endeavors have converged on the exploration of biometric authentication. Notable among these is the work by [1] that delves into threshold cryptography, aiming to confidentially enable biometric authentication. However, their solution's impractical reliance on three devices significantly hinders its use. Additionally, adaptations of the fuzzy vault concept, first introduced by [13], have sought to securely store biometric information while facilitating authentication. Nonetheless, these approaches remain vulnerable to brute force attacks, and feasible countermeasures are yet to emerge. An alternative path is illuminated by the Game-Set-MATCH framework [2], which introduces the intriguing concept of external-facing biometric authentication. This approach addresses privacy concerns by circumventing the need for biometric template storage.

This thesis draws inspiration from the Game-Set-MATCH framework [2] and narrows its focus on external biometric authentication solutions. The term "external-facing" refers to authentication mechanisms that do not entail the storage of biometric templates.

The contemporary regulatory landscape, with its growing emphasis on biometric data security, underpins the urgency for confidential biometric authentication methods. The Game-Set-MATCH framework [2] showcases promising implementation outcomes, hinting at its practical viability. However, a crucial limitation arises—this approach is incompatible with deep learning-based algorithms and prohibits the utilization of biometric templates derived from biometric features processed by these algorithms.

Deep learning has propelled biometric authentication to new heights, offering advanced algorithms that translate raw biometric data into Euclidean space vectors, where similarity is gauged through cosine metrics. Enabling the proposed external-facing biometric solution to harmonize with such deep learning-derived vectors can expedite its adoption. This alignment would not only facilitate rapid implementation but also bolster privacy safeguards for users' biometric data.

This study investigates the plausibility of devising an external biometric authentication protocol harmonious with deep learning based algorithm. The primary aim is to provide a cryptographic framework that accommodates external-facing biometric authentication while embracing the potential of deep learning-based techniques without encumbrance.

The thesis unfolds in a logical progression, traversing critical aspects of the research domain. Chapter 1 delves into an in-depth examination of cryptographic tools central to designing a privacy-preserving external-facing biometric authentication protocol. Security and privacy requirements

are precisely defined, setting the foundation for subsequent exploration. In Chapter 2, a comprehensive overview of the Game-Set-MATCH cryptographic protocol is presented, accentuating design decisions and security assumptions. Chapter 4 represents the culmination of this work, wherein the existing protocol is expanded and adapted to seamlessly integrate with deep learning-based methods. Furthermore, a novel variant of the protocol is proposed to rectify an identified error, along with a novel proof of the zero-knowledge proof of knowledge [plaintext-knowledge]. An optimized Paillier encryption under the private-key algorithm is introduced, demonstrating a notable two-fold speed enhancement.

The thesis culminates in a presentation of implementation outcomes. The original Game-Set-MATCH protocol and the extended version proposed in this work are scrutinized and benchmarked. Additionally, the encryption under the private-key algorithm's speed enhancement is quantified, corroborating its efficacy.

By embarking on this journey to harmonize cutting-edge biometric authentication with deep learning paradigms, this study endeavors to carve a pathway toward more secure and technologically integrated authentication solutions.

CHAPTER 1

Preliminaries

In this chapter, we present the various tools in making an external facing privacy preserving authentication protocol.

1.1 Biometric measurements representation and metric

As with biometrics, we need to fix a representation of a biometric measurements along with a metric to compare them.

The paper [[21]] proposes a mapping from face images to a compact euclidean space where distance corresponds to the measure of face similarity. Their system FaceNet uses machine learning methods to produce a vector $\vec{u}$ in $\mathbb{R}^d$ with L2 norm 1. further, for a fix threshold value t two measurements $\vec{u}, \vec{v}$ are considered similar if their L2 distance is less than t : $\|\vec{u} - \vec{v}\| < t$.

We are going to independently refer $\vec{u} \in \mathbb{R}^d$ and **biometric template** as the same thing. We also are going to fix the metric to be the L2 distance and the dimension space to be d.

Cosine similarity

Definition 1.1.1 (Cosine similarity)**.** for $\vec{u}, \vec{v} \in \mathbb{R}^d$, we define the cosine similarity value between them as the value : $\frac{\langle u,v \rangle}{\|u\|\|v\|} \in [-1, 1]$.

This cosine similarity value can be interpreted as a measure of how similar two vectors are.

Geometrically we have that the cosine similarity value is the cosine of the angle θ between the two vectors $\vec{u}, \vec{v}$, hence the name.

Formally, we have : $\cos(\theta) = \frac{\langle u,v \rangle}{\|u\|\|v\|}$.

Cosine similarity for fixed norm vectors

Fix $\vec{u}, \vec{v} \in \mathbb{R}^d$ two vectors of norm $\zeta \in \mathbb{R}$.

What is of interest to us is the relation between the L2 distance and cosine similarity.

In the case of fixed norm we have the following :

$$\|\vec{u} - \vec{v}\|^2 = \langle \vec{u} - \vec{v}, \vec{u} - \vec{v} \rangle$$

$$\|\vec{u} - \vec{v}\|^2 = \langle \vec{u}, \vec{u} \rangle + \langle \vec{v}, \vec{v} \rangle - 2\langle u, v \rangle$$
$$\|\vec{u} - \vec{v}\|^2 = 2\zeta^2 - 2\langle u, v \rangle$$

Furthermore, we have, for some threshold value t :

$$\|\vec{u} - \vec{v}\| < t \iff \frac{\langle u, v \rangle}{\zeta^2} > \frac{t^2}{2\zeta^2} + 1$$

This means that if 2 vectors $\vec{u}, \vec{v}$ have the same norm ζ, checking if their distance is smaller than some threshold value t is the same as checking if their cosine similarity is greater than some value t' (this value being the threshold value for cosine similarity).

This direct mapping will be handy in the design of a biometric authentication scheme based on the L2 distance, as computing the scalar product : $\langle u, v \rangle$ is much more *cryptography-friendly* than directly computing the L2 norm.

1.2 Paillier cryptosystem

The Paillier cryptosystem, introduced by Pascal Paillier in 1999 [19], is an asymmetric encryption scheme based on the computational hardness of the decisional composite residosity assumption. It offers additive homomorphic proprieties enabling computation on encrypted data without decryption. In the following section we delve into the fundamental concepts and features of the Paillier cryptosystem, exploring it's strength and limitation.

Key generation

The KeyGen algorithm takes as input 1^λ and generates a pair of keys $(N, \phi(N))$ for the security parameter λ. N is an integer that is the product of 2 primes.

Let $N = pq$ and $\phi(N) = (p-1)(q-1)$, where p and q are prime numbers, and ϕ is euler totient function. Here, we suppose that $\gcd(\phi(N), N) = 1$.

Encryption map

To understand the Paillier cryptosystem, we need to understand the group structure of the ciphertext space: $\mathbb{Z}_{N^2}^\times$.

The group $\mathbb{Z}_{N^2}^\times$ is a group of order $\phi(N)N$ that is isomorphic to the group product : $\mathbb{Z}_N \times \mathbb{Z}_N^\times$ threw the encryption map $E : \mathbb{Z}_N \times \mathbb{Z}_N^\times \to \mathbb{Z}_{N^2}^\times$ defined as: $E_N(m, r) \mapsto (1+N)^m r^N$.

This, defines the **encryption** algorithm : Given the message $m \in \mathbb{Z}_N$, sample $r \in \mathbb{Z}_N^\times$ and define the cipher text of m as $E_N(m, r)$.

Thanks to the encryption isomorphism, it is easy to see the **additive homomorphic** properties of the encryption scheme: given two random $r_1, r_2 \in \mathbb{Z}_N^\times$ and two messages $m_1, m_2 \in \mathbb{Z}_N$ we have : $E_N(m_1, r_1)E_N(m_2, r_2) = E_N(m_1 + m_2, r_1 r_2)$ which is an encryption of $m_1 + m_2$, we should note that to compute an encryption of $m_1 + m_2$ we only need the encryption of m_1 and the encryption of m_2.

Decryption

Let us describe the **deciphering map** and **deciphering algorithm** .

The decryption map $D_{\phi(N)} : \mathbb{Z}_{N^2}^\times \to \mathbb{Z}_N$, is the map that satisfies:

$$\forall (m,r) \in \mathbb{Z}_N \times \mathbb{Z}_N^\times; D_{\phi(N)}(E_N(m,r)) = m$$

This map is a group morphism.

It is not clear how one would efficiently compute m from the cipher text $c := E_N(m,r)$.

Here there is an efficient algorithm to compute the discrete logarithm, notice that for every $m \in \mathbb{Z}_N : (1+N)^m = (1+Nm) \bmod N^2$, this remark yields an efficient algorithm to extract m from the value $(1+N)^m$, namely compute m as $\frac{(1+N)^m - 1 \bmod N^2}{N}$.

Now that we have this insight, the problem becomes: how do we remove the random hiding factor r^N. For this we can exploit compatibility between $\mathbb{Z}_{N^2}^\times$ and $\mathbb{Z}_N^\times$ through modular reduction and, of course, use the trap-door $\phi(N)$ that acts as a secret key.

Notice that $c = r^N \bmod N$. In particular, given the trap door $\phi(N)$ one can compute, using modular exponentiation in $\mathbb{Z}_N^\times$ the value : $E(m,r)^{-N^{-1} \bmod \phi(N)} = r^{-1} \bmod N$. This comes from the fact that $\mathbb{Z}_N^\times$ is a group of order some divisor of $\phi(N)$ and N is co-prime with $\phi(N)$ by assumption.

Once we have r^{-1}, we can compute $E_N(0,r^{-1})E_N(m,r) = E_N(m,1) = (1+N)^m$ to remove the hiding factor and fall into the case of an easy discrete logarithm.

Security assumptions

The security of this cryptosystem is based on the computational hardness of the Decision Composite Residuosity, in which r acts as a random hiding factor.

Definition 1.2.1 (definition (Decision Composite Residuosity) [19])**.** the **DCR** assumption states that the uniform distribution over the set $\{w^N \bmod N^2 | w \in \mathbb{Z}_N^\times\}$ is computationally indistinguishable from the uniform distribution over $\mathbb{Z}_{N^2}^\times$.

Under this assumption, the Pailler cryptosystem defines an **IND-CPA** homomorphic encryption scheme.

Informally, this means that it is computationally infeasible for an adversary that does not have the trap-door/secret key $\phi(N)$ to tell the difference between two ciphertexts of messages they have chosen.

This can be formalized through the following experience.

$\mathbf{Exp}^{\text{IND-CPA}}_{\text{Paillier},\lambda}(\mathcal{A}_1, \mathcal{A}_2)$

1. Generate a public Key pair $(N, \phi(N))$ for the security parameter λ.
2. $\mathcal{A}_1$ gets the public key N and returns two messages along with an information status (m_0, m_1, s).
3. Uniformly sample $b^\star \in \{0,1\}$.
4. Compute a ciphertext of $m_{b^\star}$: $c^\star = E_N(m_{b^\star}, r)$ for some random value $r \in \mathbb{Z}_N^\times$.
5. Give *challenge* $(s, c^\star)$ to $\mathcal{A}_2$ to obtain the return value b.
6. Experience returns 1 if $b = b^\star$ and 0 otherwise.

We also define the Advantage for the attacker $(\mathcal{A}_1, \mathcal{A}_2)$ as the value :

$$\mathbf{Adv}^{\text{IND-CPA}}_{\text{Paillier},\lambda}(\mathcal{A}_1, \mathcal{A}_2) = \left| \Pr\left[\mathbf{Exp}^{\text{IND-CPA}}_{\text{Paillier},\lambda}(\mathcal{A}_1, \mathcal{A}_2) = 1 \right] - \frac{1}{2} \right|$$

Further, we say that the encryption scheme is secure in the sens of IND-CPA if for every polynomial probabilistic algorithm $(\mathcal{A}_1, \mathcal{A}_2)$ if there exists a λ large enough so that

$$\mathbf{Adv}^{\text{IND-CPA}}_{\text{Paillier},\lambda}(\mathcal{A}_1, \mathcal{A}_2)$$

is negligible.

The Paillier cryptosystem, like any other partially homomorphic encryption scheme, does not provide IND-CCA (indistinguishability under chosen ciphertext attack) security; it is susceptible to chosen ciphertext attacks, where an adversary can create particular cipher texts to gain knowledge.

The Bleichenbacher [5] attack has shown the practical significance of the IND-CCA security concept in real-world applications. This attack took advantage of the homomorphic properties of RSA encryption and an SSL server that acted as a decryption oracle, allowing it to decrypt any ciphered message.

1.3 Secure two-party computation

Secure two-party computation is a powerful protocol that allows two parties that hold private input to jointly compute a function over their respective private input without revealing anything other than the output of the function to each other.

There are multiple approaches for building 2 party MPC protocols, but here we are going to focus on the Yao's garbled circuit approach [ref].

First, we define the two cryptographic primitives involved in the two-party MPC, namely Garbled Circuit and two-message Oblivious transfer and finally we talk about the security definition of the protocol.

Two-message Oblivious transfer [6]]

Consider the following situation in which there are two parties: Alice and Bob. Alice has two messages μ_0, μ_1 and Bob has a bit b. At the end of the two-message oblivious transfer protocol, Bob should only learn m_b and Alice should not learn anything, essentially making Alice oblivious to which message Bob learns.

Definition 1.3.1 (two message oblivious transfer)**.** A two-message oblivious transfer protocol is a tuple $(\text{Round}_1, \text{Round}_2, \text{Output})$ defined as follows :

- Round_1 is probabilistic polynomial algorithm that, given a bit $b \in \{0,1\}$ outputs a message and secret state (m_1, st)
- Round_2 is a probabilistic polynomial algorithm that, given a pair of bit strings $\mu_0, \mu_1 \in \{0,1\}^k$ and a message m_1, outputs a message m_2
- Output is a polynomial time algorithm that, given a secret state st, a bit b and a message m_2, outputs a bit string $\mu \in \{0,1\}^k$.

Additionally, every algorithm takes as input some security parameter 1^λ that we may omit (they are allowed to run in polynomial time in the security parameter).

To better understand this definition, we describe in a sequence diagram how Alice and Bob would use this protocol to achieve their goal.

Now that we have described this protocol, we shall discuss the correctness and security requirement.

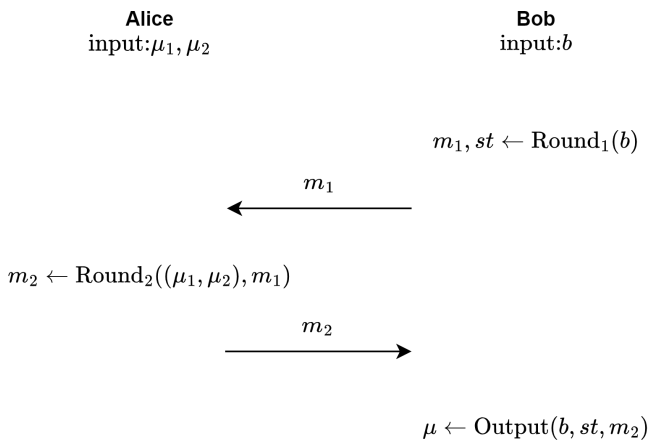

Figure 1.1: Oblivious Transfer

Correctness

Correctness encapsulates the idea that if the protocol is run correctly, Bob learns the message of his choice μ_b.

Definition 1.3.2 (Correctness)**.** Let $(\textbf{Round}_1, \textbf{Round}_2, \textbf{Output})$ be a two-message oblivious transfer protocol, $\lambda \in \mathbb{N}$, $b \in \{0,1\}$, $\mu_0, \mu_1 \in \{0,1\}^k$.

We say that a two-message oblivious transfer protocol is correct if

$$1 - \Pr\left[\mu = \mu_b : (m_1, st) \leftarrow \text{Round}_1(b), m_2 \leftarrow \text{Round}_2(m_1, \mu_0, \mu_1), \mu \leftarrow \text{Output}(m_2, b, st)\right]$$

is negligible.

Receiver Privacy

The receiver privacy encapsulates the idea that throughout the protocol, the sender (Alice) should not learn anything about the choice bit b of the receiver (Bob).

Definition 1.3.3 (Receiver Privacy)**.** Let $(\textbf{Round}_1, \textbf{Round}_2, \textbf{Output})$ be a two-message oblivious transfer protocol and $\mathcal{A}$ be a probabilistic polynomial-time algorithm. We introduce the experience $\text{Exp}_{\lambda}^{RP}(\mathcal{A})$:

1. uniformly sample $b^\star \leftarrow \{0,1\}$
2. $m_1 \leftarrow \text{Round}_1(b^\star, 1^\lambda)$
3. $b \leftarrow \mathcal{A}(m_1, 1^\lambda)$
4. return 1 if $b = b^\star$ 0 otherwise

We say that the two-message oblivious transfer protocol satisfies receiver privacy if there exists λ big enough such that for every polynomial probabilistic time attacker $\mathcal{A}$

$$\text{Adv}_{\lambda}^{RP} = \left|\Pr\left[\textbf{Exp}_{\lambda}^{\text{RP}}(\mathcal{A}) = 1\right] - \frac{1}{2}\right|$$

is negligible.

Sender Privacy

The sender privacy encapsulates the idea that throughout the protocol, the receiver (Bob) should not learn anything other than μ_b .

Definition 1.3.4 (Sender Privacy)**.** Let (**Round**$_1$, **Round**$_2$, **Output**) be a two-message oblivious transfer protocol, and $\mathcal{A}_1, \mathcal{A}_2$, S be probabilistic polynomial-time algorithms. We introduce the experience $\text{Exp}_\lambda^{SP}(\mathcal{A}_1, \mathcal{A}_2, \mathcal{S})$:

1. uniformly sample $b' \leftarrow \{0,1\}$
2. $(m_1, s) \leftarrow \mathcal{A}_1(b', 1^\lambda)$
3. uniformly sample $b^\star \leftarrow \{0,1\}$
4. $m_2 \leftarrow \text{Round}_2((\mu_0, \mu_1), m_1, 1^\lambda)$ if $b^\star = 0$ else $S(\mu_{b^\star}, m_1, 1^\lambda)$
5. $b \leftarrow \mathcal{A}_2(s, m_2, 1^\lambda)$
6. return 1 if $b = b^\star$ else 0

We say that the two-message oblivious transfer protocol satisfies receiver privacy if there exists λ big enough and probabilistic polynomial time simulator S such that for every polynomial probabilistic time attackers $\mathcal{A}_1, \mathcal{A}_2$ and messages μ_0, μ_1

$$\text{Adv}_\lambda^{SP} = \left| \Pr\left[\text{Exp}_\lambda^{SP}(\mathcal{A}_1, \mathcal{A}_2, \mathcal{S}) = 1\right] - \frac{1}{2} \right|$$

is negligible.

Garbled circuits

As the name suggests, a garbled circuit is a cryptographic technique to *garble* boolean circuit.

The first question that we have to address in order to continue is the following : What is a boolean circuit?

What is a boolean circuit ?

There is multiple possible definitions of what is a boolean circuit, the authors of [4] define a boolean circuit as a 6-tuple $f = (n, m, q, A, B, G)$ in which we have

- n : the number of input (bit)
- m : the number of output (bit)
- q : the number of wires
- A : a function to identify each gate's first incoming wire.
- B : a function to identify each gate's second incoming wire.
- G : a function that determines the functionality of each gate.

We also suppose that there is an easy encoding of f as a string.

Garbling schemes

Now that we have established what a circuit is, let us define a garbling scheme, based on [4]. Suppose f a string describing a circuit and $x \in \{0,1\}^n$ an input for that circuit.

A garbling scheme is a five-tuple of algorithms $\mathcal{G}$ = **(Gb, En, De, Ev, ev)** defined as follows:

- **ev**(f, x): returns $f(x) \in \{0,1\}^m$.
- **Gb**$(1^\lambda, f)$: returns a triple of strings (F, e, d) in which F describes a garbled function, e describes an encoding function, d describes a decoding function.
- **En**(e, x) : returns a garbled input X.
- **Ev**(F, X): returns a garbled output Y.
- **De**(d, Y) returns the final output y.

Finally, we (naturally) require the correctness of the scheme, which is that **De**(d,**Ev**(F,**En**(e, x))) = **ev**(f, x).

This is a garbled circuit description that is handy to work with, we simply note here that there is more to these definitions but this would make this description too lengthy which is not the aim here, furthermore, for more details we refer the reader to [4] .

Security requirements

The authors of [4] define two notions of privacy, the first is an indistinguishability-based notion of privacy and the second is a simulation-based notion of privacy.

Here, we describe the simulation-based notion of privacy.

Informally, this definition encapsulates this idea that for a garbling scheme $\mathcal{G}$; the tuple F, X, d does not reveal anything about the input x other than the output $f(x)$.

Consider $\mathcal{A}_1, \mathcal{A}_2$ a polynomial-time probabilistic attacker, $\mathcal{S}$ a polynomial-time probabilistic simulator, $\mathcal{G}$ = **(Gb, En, De, Ev, ev)** a garbling scheme and Φ an information function (for example, we might allow the attacker to know the circuit evaluated, or simply the topology of the circuit).

We introduce the experience $\mathbf{Exp}_\lambda^{\text{privSim}}(\mathcal{A}_1, \mathcal{A}_2, \mathcal{S})$:

- $(f, x), s \leftarrow \mathcal{A}_1(1^\lambda)$
- uniformly sample $b^\star \leftarrow \{0,1\}$
- if $b^\star = 0$ then $(F, e, d) \leftarrow$ **Gb**$(1^\lambda, f); X \leftarrow$ **En**(e, x) otherwise $(F, X, d) \leftarrow \mathcal{S}(1^\lambda, y, \Phi(f))$
- $b \leftarrow \mathcal{A}_2(1^\lambda, (F, X, d), s)$
- return 1 if $b = b^\star$ 0 otherwise

We say that a garbling scheme $\mathcal{G}$ satisfies simulation privacy relative to Φ if for every attacker $\mathcal{A}_1, \mathcal{A}_2$ there is a simulator $\mathcal{S}$ such that

$$\mathbf{Adv}_\lambda^{\text{privSim}}(\mathcal{A}_1, \mathcal{A}_2, \mathcal{S}) = \left| \Pr\left[\mathbf{Exp}_\lambda^{\text{privSim}}(\mathcal{A}_1, \mathcal{A}_2, \mathcal{S}) = 1\right] - \frac{1}{2} \right|$$

is negligible.

Projective garbling scheme

In order to build a secure function evaluation protocol from a garbling scheme, we will require the garbing scheme to be projective.

Informally, a garbling scheme is projective if the encoding e describes a list of tokens: one pair for each input bit so that the encoding function simply selects tokens depending on the input bit.

Formally, on input $\vec{e} = ((X_i^0, X_i^1))_{i\in\{1,...,n\}}, \vec{x} = (x_i)_{i\in\{1,...,n\}}$; $\mathbf{En}(\vec{e}, \vec{x}) = (X_i^{x_i})_{i\in\{1,...,n\}}$.

Secure two party computation protocol

Traditional techniques for secure function evaluation (SFE) and private function evaluation (PFE) incorporate the garbled-circuit approach with oblivious transfer (OT).

In this section we aim at describing a 2 party secure function evaluation protocol from a garbled circuit and a 2 party oblivious transfer protocol in the honest-but-curious setting: The parties adhere to the protocol, and their perspectives do not enable the computation of any unwanted information.

Consider 2 party : Alice and Bob. Alice has the input $\vec{x_A} \in \{0,1\}^{n_A}$ and bob has the input $\vec{x_B} \in \{0,1\}^{n_B}$ and f is circuit known to both parties; furthermore, Alice and Bob wish to learn the output of the function$f(\vec{x_B}, \vec{x_A})$ without disclosing their private input.

To build such a protocol we are going to use two-message oblivious transfer and a projective garbled circuit.

Protocol description

Consider $(\text{Round}_1, \text{Round}_2, \text{Output})$ a two message oblivious transfer and $\mathcal{G}$ = **(Gb, En, De, Ev, ev)** a protective garbled circuit.

The protocol goes as follows :

- Bob computes $(F, \vec{e}, d) \leftarrow \mathbf{Gb}(1^\lambda, f)$
- Bob computes $\vec{X_B} \leftarrow \mathbf{En}(\vec{e}, \vec{x_B})$ (notice that here, Bob only garble his input using the first n_B *tokens*)
- Bob sends $(F, \vec{X_B}, d)$ to Alice.
- for every $i \in \{1, ..., n_A\}$
  - Alice computes $m_1, st \leftarrow \mathbf{Round}_1(x_{n_B+i}, 1^\lambda)$ and sends m_1 to Bob
  - Bob computes $m_2 \leftarrow \mathbf{Round}_2((X^0_{n_B+i}, X^1_{n_B+i}), m_1)$ and sends m_2 to Alice
  - Alice sets $(\vec{X_A})_i = \mathbf{Output}(x_{n_B+i}, st, m_2)$
- Alice computes $\vec{X} = \vec{X_B} \parallel \vec{X_A}$ (the concatenation of the two vectors); $Y \leftarrow \text{Ev}(F, \vec{X})$; $\vec{y} \leftarrow \text{De}(d, Y)$

Security in the honest-but-curious setting

It is easy to see that this protocol is secure in the honest-but-curious setting assuming the security/privacy of garbled circuit and two message oblivious transfer.

Informally, Alice only learns $F, \vec{X}, d$ due to the security of two message oblivious transfer; furthermore, Alice does not learn anything about $\vec{x_B}$ other than the output y of the function thanks to

the security of projective garbled circuits. From Bob's point of view, thanks to the security of two message oblivious transfer, he does not learn anything about the input bits $\vec{x_A}$ of Alice.

1.4 Digital Signature

In cryptography, a digital signature is a means of ensuring the authenticity of a message by proving that the sender of the message has knowledge of a private key for the corresponding signature scheme. This scheme consists of three algorithms : **KeyGen** which on input 1^λ with λ a security parameter return a pair **(psk, ssk)** the public signing key and the private signing key for the signature scheme; **Sign** which on input m the message to sign and **ssk** the private signing key returns a valid signature σ for the message m. **Verify** which on input the public key **ssk**, a message m and its signature σ return a bit that indicates the result "accepted" (1) or "rejected" (0).

Definition 1.4.1 (Digital Signatures)**.** A digital signature scheme consists of the following three probabilistic polynomial time algorithms (KeyGen, Sign, Verify).

- KeyGen$(1^\lambda) \rightarrow$ psk, ssk
- Sign(ssk, m) $\rightarrow \sigma$
- Verify(psk, m, σ) $\in \{0, 1\}$

Security requirement

The following security properties should be satisfied.

Correctness

$\forall \lambda \in \mathbb{N}, \forall$ psk,ssk $\leftarrow$ KeyGen(1^λ) we have Verify(psk, m, Sign(ssk, m) $= 1$.

Unforgeability

Informally, Unforgeability is the inability of an attacker to forge a signature for any message, not signed by a legitimate signer.

Let KeyGen,Sign,Verify be a digital signature scheme, we introduce the experience $\text{Exp}_\lambda^{\text{Unforgeability}}(\mathcal{A})$

1. psk,ssk $\leftarrow$ KeyGen(1^λ)
2. Initialise the list $L = \{\}$
3. $m^\star, \sigma^\star \leftarrow \mathcal{A}(1^\lambda, \text{psk}, \mathcal{O}^{\text{ssk}})$ ($\mathcal{O}^{\text{ssk}}$ is an oracle that $\mathcal{A}$ can query.)
4. return 1 if Verify(psk, $m^\star, \sigma^\star) = 1$ and $m^\star \notin L$ else 0

Additionally, on query m from $\mathcal{A}$, the oracle $\mathcal{O}^{\text{ssk}}$ is defined as follows :

1. $\sigma \leftarrow$ Sign(ssk, m)
2. add m into L
3. outputs σ

A digital signature scheme is said to be unforgeable if of every probabilistic polynomial-time attacker $\mathcal{A}$ the advantage

$$\text{Adv}_\lambda^{\text{Unforgeability}} := \Pr\left[\text{Exp}_\lambda^{\text{Unforgeability}}(\mathcal{A}) = 1\right]$$

is negligible.

1.5 Packing Methode for Paillier cryptosystem

In this section, we will describe the packing technique used in [2] and discuss its drawbacks and advantages.

The idea behind the packing technique introduced in [2] is that the pallier plain-text space is usually very large due to security requirements. However, in their application the plaintext is only a couple of bits (less than 32 bits), making the encryption scheme not efficient in terms of bandwidth usage.Indeed, a typical paillier modulus size is about 2048 bits making the plain-text space much bigger than the plain-text. Moreover ,in their application, they only need to perform a well-known number of homomorphic operations, which leaves room for more efficient message encoding that would be compatible with these restrictions.

Encoding of vectors

Let N be an RSA modulus, and let $\vec{u}$ and $\vec{w}$ be two vectors of size m.

There are 2 levels of encoding: level 1 and level 2, these two encodings both support the addition operation :

$$\text{Encode}_1(\vec{u}) + \text{Encode}_1(\vec{w}) = \text{Encode}_1(\vec{w} + \vec{u})$$

$$\text{Encode}_2(\vec{u}) + \text{Encode}_2(\vec{w}) = \text{Encode}_2(\vec{w} + \vec{u})$$

Then, the 2 level 1 encoding multiplied together over $\mathbb{Z}$ creates a level 2 encoding :

$$\text{Encode}_1(\vec{u}) \times \text{Encode}_1(\vec{w}) = \text{Encode}_2(\vec{w} \times \vec{u})$$

in which the vector multiplication is the term-to-term multiplication for vectors.

Then when an encoding becomes a level 2 encoding, we cannot perform multiplication anymore; in that sense, the encoding only supports one scalar operation.

This encoding has the advantage of being highly compatible with paillier encyption, indeed, the encoding can be computed homomorphically; that is, given $\text{Enc}(u_1), ..., \text{Enc}(u_n)$ one can effiently compute, without the paillier private key, the value $\text{Enc}(\text{Encode}_1(u_1))$. We note, however, that in order to be able to compute this value, one has to have the grantee that u_i respect the size condition.

For more details on security requirements, parameter selection, and formal proof, we refer the reader to [2].

Performance drawback

This encoding, as discussed previously, helps reduce bandwidth usage in a way that would be compatible with the homomorphic proprieties that the Paillier cryptosystem offers (to some extent). However, we lose when it comes to computation performance.

This depends directly on the packing parameter m : the number of values that we can pack together.

Consider the typical case of N a modulus of 2048 bits , $m = 4$ the packing factor, and values of bit length 16. Then the function Encode_1 allows the packing of 4 values that are less than 2^{16} into an integer less than N.

Now consider the case in which we want to compute some form of the value $\vec{w} \times \vec{u}$ in the encryption domain given some form of $\vec{u}$ ciphered with Paillier cryptosystem. More precisely, given $(\text{Enc}(u_1), .., \text{Enc}(u_4))$ and $w_1, ..., w_4$ or given $\text{Enc}(\text{Encode}_1(\vec{u}))$ and $\text{Encode}_1(\vec{w})$ we either want to compute the vector $(\text{Enc}(u_1 w_1), .., \text{Enc}(w_4 u_4))$ or the value $\text{Enc}(\text{Encode}_2(\vec{w} \times \vec{u}))$.

In this specific case, the naive approach is going to yield better performance results. Indeed, since the u_i are supposed to be small, we are going to have the $m = 4$ modular exponentiation with small integers, while with the encoded version we are going to have about 1 modular exponentation with a very big integer, namely $\text{Encode}_1(\vec{w})$ that has a size comparable to N.

1.6 Zero knowledge proofs

A zero-knowledge proof or zero-knowledge protocol is a method by which one party P (prover) can prove to another party V (verifier) that a given statement is true, while avoiding conveying to V any information beyond the mere fact of the statement's truth.

Definition 1.6.1 (Honest Verifier Zero-knowledge proof)**.** Formally, suppose $\mathcal{L}$ is an NP language. A zero-knowledge protocol is an interaction between two probabilistic polynomial-time algorithms P and V with P trying to convince V that $x \in \mathcal{L}$ and satisfy the following proprieties:

- *Completeness*. If $x \in \mathcal{L}$, w a correct witness, and the protocol is honestly executed, then

$$\Pr[P(w, x) \text{ convince V that } x \in \mathcal{L}] = 1$$

- *Soundness*. if $x \notin \mathcal{L}$, for every probabilistic polynomial-time algorithm algorithm $\hat{P}$,

$$\Pr[\hat{P}(x) \text{ convince V that } x \in \mathcal{L}]$$

  is negligible.

- *Zero-knowledge.* an interaction is considered to be zero-knowledge if for all $x \in \mathcal{L}$ there is a probabilistic polynomial time algorithm which can output a transcript τ' such that τ' is computationally indistinguishable from a real transcript.

Sometimes we need more than just proving that some statement x is in a language $\mathcal{L}$ and we need to verify that the prover actually has a correct witness w.

This is why we introduce a strictly stronger definition, namely the Honest Verifier Zero-Knowledge proof of Knowledge.

In order to ease the notation, we introduce a new binary relation.

We introduce the witness relation R defined as : $\forall w, x; xRw \iff x \in \mathcal{L}$ and w is a correct witness of x .

Definition 1.6.2 (Honest Verifier Zero-Knowledge proof of Knowledge)**.** An honest Verifier Zero-Knowledge proof of knowledge is an honest Verifier Zero-Knowledge proof between a prover P and a verifier V with the additional propriety:

- *Proof of Knowledge*. For every probabilistic polynomial time algorithm algorithm $\hat{P}$, and for all x, there exists a probabilistic polynomial time algorithm E (Extractor) such that :

$$\Pr[\hat{P}(x) \text{ convince V that } x \in \mathcal{L} \text{ and } w \longleftarrow E(\hat{P}, x) \text{ such that } x \not{R} w]$$

  is negligible.

Informally, the proof of knowledge propriety encapsulate the idea that if the prover can convince the verifier that a statement is true, then he can also compute a correct witness for that statement. This definition of "knowledge" pairs well with computational adversaries.

We now present 3 honest verifier zero-knowledge protocol (of knowledge)...

Correct key generation

As discussed in the Paillier cryptosystem section, we make the assumption that the private key and the public key $\phi(N), N$ are coprime. This assumption is necessary for the encryption map to define a bijection. Often times, we need to make sure that this condition is met, and of course revealing the private key would completely impede the encryption scheme.

One natural way to solve this problem would be to resort to zero-knowledge proof.

We present here an honest verifier zero-knowledge proof for the language : $\mathcal{L} = \{N, \gcd(N, \phi(N)) = 1\}$.

furthermore, this proof can be made non-interactive using the Fiat-Shamir heuristic [10][11].

Parameters:

- Public integer parameter m to be fixed (typically $m = 11$ is sufficient)
- The statement N
- The witness $\phi(N)$

Protocol $\Pi_{\text{correct-key}}$:

- V uniformly sample m random values $\rho_1, ..., \rho_m \in \mathbb{Z}_N^\times$ and sends them to P.
- P computes $\sigma_i = \rho_i^{N^{-1} \bmod \phi(N)} \bmod N$ and sends $\sigma_1, ..., \sigma_m$ to V.
- V accepts if and only if for all i : $\sigma_i^N = \rho_i \bmod N$

The main idea behind this proof is that, if N and $\phi(N)$ are not co-prime, there is a good chance that one of the ρ_i is not an N'th root in $\mathbb{Z}_N^\times$ hence the integer m that parameterizes the soundness error probability.

For a formal proof, we refer the reader to [10].

This zero-knowledge proof has to be used in a non-interactive way in practice since a malicious verifier could send well chosen ρ_i in order to get malicious information.

In fact, consider the following scenario: the verifier has ciphertext $c = E_N(m, r) = (1 + N)^m r^N \bmod N^2$. What V can do is compute $c \bmod N = r^N \bmod N$ and then set ρ_1 to be this value, then V receives the associated $\sigma_1 = r \bmod N$ threw the protocol $\Pi_{\text{corrct-key}}$ and can proceed to decipher the ciphertext c.

Using the Fiat-Shamir heuristic, we ensure that $(\rho_i)_i$ are not maliciously generated.

Further discussion

One might think that this proof is not sufficient since it does not assure that N is the product of 2 primes; indeed, if N is the product of 3 primes, for example, then the proof succeeds. However, the authors of [10] argue that it is sufficient since the encryption map defines an isomorphism. Furthermore, in practice, as suggested by **??** we also check that N has no prime factor under a certain bond.

This proof is very efficient to run even on the higher end of security requirement. To give an idea of its efficiency, for a 3072 bits modulus N, it only takes about 50 milliseconds to run the proof on a regular laptop.

Plain Text knowledge

Here we describe an Honest Verifier zero knowledge proof of knowledge protocol that proves the knowledge of plain text under paillier cipher-texts.

This protocol ensures that the prover knows the underlining plaintext of a set of paillier cipher-text.

parameters:

- A paillier public key: N.
- The statement : $\vec{c} = (c_1 = E(m_1, r_1), c_2 = E(m_2, r_2), ..., c_n = E(m_n, r_n)) \in (\mathbb{Z}_{N^2}^{\times})^n$.
- The witness $(m_1, r_1), ..., (m_n, r_n) \in \mathbb{Z}_N \times \mathbb{Z}_N^{\times}$.

protocol $\Pi_{\text{plain-text-knowledge}}$

- **P** uniformly samples $(m, r) \in \mathbb{Z}_N \times \mathbb{Z}_N^{\times}$ and sends $c = E(m, r)$.
- **V** uniformly samples and sends $\vec{e} = (e_1, ..., e_n) \in \mathbb{Z}_N^n$
- **P** sends $M = m + \sum_i e_i m_i \mod N; R = r \prod_i r_i^{e_i} \mod N$
- **V** accepts if and only if $E(M, R) = c \prod c_i^{e_i} \mod N^2$

This version of the protocol is a batch version of the classical plain text knowledge proof [7] presented in [2]. It leverages the linear structure of the original proof to achieve better communication performance, only requiring the prover to send a constant number of messages.

Furthermore, this protocol can be made noninteractive using the Fiat-Shamir heuristic, making the communication complexity constant (independent on n).

L2 norm in $\mathbb{Z}$ and $\mathbb{Z}_N$

Consider the following situation: We have n paillier cipher texts $\text{Enc}[x_1], ..., \text{Enc}[x_n]$, an integer $y < N$ and we wish to prove in zero-knowledge fashion that the L2 norm of $\vec{x} = (x_1, ..., x_n)$ is equal to y over the integers : $\sum_i x_i^2 = y$ ($x_i \in [0, N-1]$).

This situation naturally arises when dealing with ciphered vectors; however ,building such a zero-knowledge proof is challenging, since the equality has to hold over the integers.

L2 norm in $\mathbb{Z}$

The authors of [2] built a zero-knowledge proof for when x_i is much less than N.

The core of their idea is to introduce a prime number $\hat{p}$ in the challenge, once the Prover receives this prime number, he is tasked with proving that $\sum_i x_i^2 = y \mod N\hat{p}$, this effectively forces a cheating prover to guess p in order to dupe the verifier and, of course, can only happen with negligible probability.

This zero-knowledge proof is fast in practice but cannot be made non-interactive as a drawback.

In the following we describe this zero-knowledge proof; for further reading and proof, see [2].

parameters:

- A paillier public key N
- A public integer value $y \in \mathbb{Z}$

- statement : $c_1 = E(x_1, r_n), ..., c_n = E(x_n, r_n) \in \mathbb{Z}_{N^2}^{\times}$
- witness $(x_1, r_1), ..., (x_n, r_n) \in \mathbb{Z}_N \times \mathbb{Z}_N^{\times}$

protocol $\Pi_{\text{L2-norm-}\mathbb{Z}}$:

- V uniformly sample a prime $\hat{p} < N$
- V uniformly sample values : $\alpha \in Z_{2^\lambda}$, $\rho_1, ...\rho_n, \beta, \hat{\beta} \in \mathbb{Z}_N$. Then it sends P the following challenge:
  - Enc$[w_i]$ = Enc$[\alpha x_i + \rho_i]$ for all i
  - Enc$[v]$ = Enc$[\sum_i(-2\alpha\rho_i x_i - \rho_i^2) + \beta]$
  - Enc$[v_p]$ = Enc$[\sum_i(\gamma_i x_i) + d + \hat{\beta}]$ where $\gamma_i := (-2\alpha\rho_i \bmod \hat{p})$, $d := (\sum_i -\rho_i^2) \bmod \hat{p}$
- P deciphers Enc$[w_i]$, Enc$[v]$, Enc$[v_p]$ and sends the following to V :
  - $z := v + \sum w_i^2 \bmod N$
  - $\hat{z} := \hat{v} + \sum w_i^2 \bmod \hat{p}$
- V accept if $z = y\alpha^2 + \beta$ and $\hat{z} = y\alpha^2 + \hat{\beta}$

CHAPTER 2

State of the art

In this chapter, we briefly present the relevant work in privacy preserving biometric authentication. We first talk about the generic technique used in both biometric authentication then we focus on external facing solutions, its advantages and drawbacks.

2.1 Threshold cryptography

One way to guarantee the privacy of biometric templates is to utilize threshold cryptography as a form of protection. The concept is straightforward: attempt to use the various gadgets that we possess (smartwatch, smartphone, etc.) to spread the biometric template across these devices in a manner that would enable privacy-preserving authentication.

BETA : Biometric Enabled Threshold Authentication

The work [1] proposed 3 protocols that would allow biometric template distribution among $t \geq 3$ devices.

The first two protocols enable the use of any distance metric and incorporate powerful techniques such as multi-party computation and fully homomorphic threshold encryption. These protocols are secure in the universal composability framework and can tolerate up to $t - 1$ malicious devices. Despite the utilization of powerful cryptographic techniques and the capability to accommodate any metric, these protocols are not very efficient.

The third protocol attempts to address the issue of efficiency by using cosine similarity as a metric. It employs more efficient tools such as Paillier encryption, NIKZ proofs, and a lightweight garbled circuit. Additionally, it sets $t = 3$ and can handle up to one malicious device.

These three protocols have an awkward requirement: three or more active devices must be present in order to authenticate. Additionally, there is no evidence of successful implementation.

2.2 Fuzzy vault and biometric key exchange

Fuzzy vault

A fuzzy vault is a cryptographic construct designed to bridge the gap between the inherent variability of biometric data and the need for reliable and secure authentication mechanism. It functions as a mathematical algorithm that extracts a stable and reproducible cryptographic key from biometric information, while accommodating the natural variability of such data. By distilling a consistent key from inherently noisy biometric signals, a fuzzy extractor ensure that subsequent authentication processes remain accurate and reliable.

Fuzzy vault authentication protocol

A Fuzzy vault can be used to construct an authentication system, as suggested by [13]. During the setup phase, the user would generate a secret f, locks it with their biometric data in a vault V, and send the server the hashed value of f, $\mathrm{H}(f)$, and the vault V. When the user wishes to authenticate, the server will provide them with the stored vault, and the user can use a new biometric recording to recover the secret and authenticate.

Fuzzy vault drawbacks

This protocol is comparable to the salt and hash techniques used for password authentication. However, it is vulnerable to brute-force attacks, since any user can pretend to be someone else and gain access to the vault. Correlation attacks, as demonstrated by [23], can also be used to exploit this protocol. To counteract these issues, work such as [18, 3] proposes constructions that layers a variety of cryptographic tools to prevent these types of attack, but this has significant security and efficiency drawbacks as a result. Additionally, this technique does not allow for advanced matching metrics, which can affect accuracy performance.

2.3 External facing biometric authentication

In the paper Game-Set-MATCH [2] the authors introduce a new framework for biometric authentication, they go against the standard technique and instead of trying to store the biometric template, they give back this responsibility to the user's device.

We decided to investigate this solution for a few reasons; the results of the paper's implementation demonstrate that this protocol is highly efficient and could be a viable answer to the issue of biometric authentication with strong privacy grantees. Additionally, the tools used in the design of this solution are very promising, incorporating cryptographic techniques such as multiparty computation and homomorphic encryption, making it an ideal topic for a master's degree internship.

CHAPTER 3

Analysis of exteran facing biometric authentication "Game-Set-MATCH" [2]

In this chapter, we present the external-facing biometric authentication scheme proposed in Game-Set-MATCH [2].

We will begin by describing the context in which we will be operating and then progress to the details of the protocol. This chapter aims to give an insight into the cryptographic techniques employed in the development of this solution. We will present a simplified version of the protocol and highlighting the main difference. We will conclude by discussing the implementation results of this protocol and highlight any issues it may have.

3.1 External facing

Parties involved

In this external facing biometric authentication scheme there are 3 parties: the **Device (D)**, **Service Provider (SP)** and the **Terminal (T)**.

We will illustrate a payment situation to illustrate the role each participant will have in this protocol.

Use case: Biometric ATM

The scenario of a biometric ATM, referred to as the **Terminal (T)**, requires a user to enroll with a **Service Provider (SP)** before they can use it. The individual will use their device and biometric data to register with the service provider. This involves the device capturing the user's biometric information and using it to sign up with the service provider. Once registered, the user can then initiate an authentication session with the **ATM (T)** in order to withdraw money. During this session, the **ATM (T)** will take a biometric recording of the user and with the help of the device authenticate the user.

At the time of registration, the user's device takes a biometric recording. Subsequently, when authenticating, the terminal captures the biometric sample, while the device does not take any recording.

Biometric template

We will set the metric to be cosine similarity, and thus each biometric template will be a vector of length n as mentioned in [preliminaries cosine similarity]. Additionally, for ease of use, as stated in [2], each vector component will be a positive integer. Lastly, we demand that each biometric template has an L2 norm of ζ.

We introduce an implicit parameter t that serves as an acceptance threshold for cosine similarity; the vectors $\vec{u}; \vec{w}$ are considered similar if and only if the inner product of the two vectors is greater than or equal to t.

It is worth noting that the requirement for the vector component to be positive is restrictive in practice, and this will be explored further later in the conversation.

3.2 The Protocol [Game-Set-MATCH]

Now that the structure is established, we can explore the protocol's specifics and details. As demonstrated in the biometric ATM example, the protocol consists of two main components: the enrollment process between the User's **Device (D)** and the **Service Provider (SP)**, and the matching between the user's **Device (D)** and the **Terminal (T)**.

High level overview

Enrollment

At the time of enrollment, instead of giving **SP** the biometric template in plain text, the device will encrypt it using Paillier encryption. This is done to protect the privacy of the biometric template, while still allowing homomorphic operations to be done on the template.

This approach presents an issue however: there is no guarantee that the encrypted data is a valid biometric template (e.g. that the L2 norm is ζ).

To address this issue, we will take advantage of zero-knowledge proofs as a practical method of demonstrating that the underlying plain-text is a legitimate biometric template while still maintaining the confidentiality that encryption offers.

Matching

During the matching protocol between the user's device and the terminal, the terminal will take as input the biometric template it recorded from the user attempting to authenticate, $\vec{w}$, and the ciphered biometric template from the user's device used during registration, Enc$[\vec{u}]$.

Then, it will operate homomorphically on the ciphered biometric template to compute an encryption of the cosine similarity between the two vectors.

To conceal $\vec{w}$ from **D**, the terminal will also information theoretically hide this value. Finally, **D** and **T** will run a two-party MPC protocol to determine if the cosine similarity is greater than the threshold, thus indicating a match.

3.3 Enrollment

Parameters

During the enrollment, **D** has input vector $\vec{u} \in \mathbb{N}^n$ such that $\|\vec{u}\| = \zeta$. We suppose that **SP** has public and private signing key pairs (psk,ssk).

The enrollment goes as follows:

- **D** generates a pair of Paillier Encryption keys $(N, \phi(N))$ for security parameter λ.
- **D** computes and sends the ciphered biometric template using paillier encryption : $\vec{c} = (\text{Enc}(u_1, r_1), ..., \text{Enc}(u_n, r_n))$ along with the public key N .
- Execute ZK proofs : $\Pi_{\text{L2-norm-}\mathbb{Z}}(\zeta, \vec{c})$, $\Pi_{\text{plain-text knowledge}}(\vec{c})$ and $\Pi_{\text{correct-key}}(N)$ With **D** as the prover, **SP** as the verifier and $(x_i, r_i)_i$ as the witness.
- If one of the proofs is not correct, **SP** ends the protocol.
- Otherwise: **SP** signs the ciphered biometric template attached with the public key: $(\text{Enc}[\vec{u}], N)$ and sends the signature σ to **D**.

3.4 Matching

Let k be an integer such that for every valid biometric template $\vec{u}, \vec{w}$ we have $\langle \vec{u}, \vec{w} \rangle < 2^k \leq N$.

During the matching part, we assume that **T** has public signing key psk and valid biometric template $\vec{w}$. We also assume that some public threshold value t is known to both parties.

- **D** sends the ciphered biometric template along with the Paillier public key and the associated signature from **SP** to **T** : $((\text{Enc}[\vec{u}], N), \sigma)$.
- **T** upon receiving values $(((\vec{c}, N), \sigma)$ does the following :
  - If the signature is not valid, **T** ends the protocol : $\text{Verify}(psk, (N, [\vec{u}]), \sigma) \stackrel{?}{=} 1$.
  - homomorphically computes the value $\text{Enc}(\langle \vec{u}, \vec{w} \rangle)$ representing the cosine similarity between biometric template $\vec{u}$ and $\vec{w}$.
  - samples $a_1 \in \{0, ..., 2^\lambda\}, a_0 \in \{0, ..., 2^{k+2\lambda}\}$ and $b_0 \in \{0, ..., 2^{k+\lambda}\}$ uniformly at random.
  - Computes and sends to **D** the ciphered-MAC values $A = \text{Enc}[a_1 \langle \vec{u}, \vec{w} \rangle + a_0]$ and $B = \text{Enc}[\langle \vec{u}, \vec{w} \rangle + b_0]$.
- **D** deciphers values A, B to get integer values a, b .
- **D** and **T** run a garbled circuit based multi party computation with (a_1, a_0, b_0) as private input for **T**, (a, b) as private input for **D** and the function f_t as the circuit. At the end of this **T** learns a bit e representing weather or not $\langle \vec{u}, \vec{w} \rangle \geq t$.
- **T** output "success" if $e = 1$ else **T** outputs "failure".

3.5 The circuit to evaluate

Let us describe the high-level functionality of the circuit f_t used during the matching phase between **T** and **D**.

Inputs : (a, b, a_0, b_0, a_1)

Computation:

- set inner_product $= b - b_0$;
- set $a' = a_1 \times$ inner_product $+ a_0$
- return 1 if $(a' = a)$ and (inner_product $\geq t$) otherwise 0

This circuit aims to verify that the inner product between the reference biometric template $\vec{u}$ and the recorded biometric template $\vec{w}$ is greater than some fixed threshold t.

In practice, since this is a circuit we have a choice of bit encoding to do; that is, how do we represent integer. A natural way of doing this is to consider the binary representation of each integer. However, this encoding does not support negative values as a result, so the computation of inner_product can only be positive. This encoding will be relevent in the future discussion.

3.6 MAC values

The circuit verifies a MAC value within it; this is because if we just conceal the value $\langle \vec{u}, \vec{w} \rangle$; that is, during the matching phase we would only transmit the value $b = \langle \vec{u}, \vec{w} \rangle + b_0$ (concealing it with some random value b_0), then we would have no assurance that **D** is actually using the values **T** sent as input to the multiparty computation.

To solve this issues we present a challenge to be solved within the garbled circuit that provides an authentication mechanism that is compatible with Paillier encryption and ensures that the value of inner_producer is equivalent to $\langle \vec{u}, \vec{w} \rangle$: the result of the homomorphic computation done by **T**.

The values of a_0, a_1, b_0 are selected in such a way that the inner product $\langle u, w \rangle$ is statistically concealed.

Notice that the values (a_0, a_1, b_0) are not uniformly sampled in $\mathbb{Z}_n$ which could be an option, but would completely impede the efficiency of the protocol.

Indeed, this choice of MAC values would require large integer arithmetic in the circuit, which would make the circuit much bigger and, as a result, much slower.

Furthermore, we argue that there is no need to choose such large values and choosing $a_1 \in \{0, ..., 2^{\lambda}\}, a_0 \in \{0, ..., 2^{k+2\lambda}\}$ and $b_0 \in \{0, ..., 2^{k+\lambda}\}$ suffices.

It can be argued that since $\langle \vec{u}, \vec{w} \rangle < 2^k$, the uniform distribution over the set $\{\langle \vec{u}, \vec{w} \rangle + b_0 | b_0 \in \{0, ..., 2^{k+\lambda}\}\}$ and the uniform distribution over the set $\{1, ..., 2^{\lambda+k}\}$ are indistinguishable from a computational standpoint. The same is true for the sets $\{a_1 \langle \vec{u}, \vec{w} \rangle + a_0 | a_1 \in \{0, ..., 2^{\lambda}\}, a_0 \in \{0, ..., 2^{k+2\lambda}\}\}$ and $\{0, ..., 2^{k+2\lambda}\}$.

This ensures that when the device deciphers A, B to recover integer values a, b it does not learn anything about the inner product $\langle \vec{u}, \vec{w} \rangle$ as it would reveal unwanted information about $\vec{w}$.

3.7 Security

We strive to keep the value of the inner product hidden in our protocol, as it could provide **D** and **T** with unwanted information. However, when a device and the terminal are matched, some information is unavoidably revealed; namely, the terminal and the device learn the predicate $\langle \vec{u}, \vec{w} \rangle > t$.

This information leakage is an integral part of the protocol and cannot be avoided, as it represents biometric authentication. To capture the security requirement in this context, the authors of [2] employ the ideal real-world paradigm and introduce the biometric authentication functionality.

Formally, they show that the protocol is secure against any adversary that can maliciously corrupt the device or jointly corrupt the terminal and authority in a semi-honest manner with no leakage to any party with respect to the biometric authentication functionality. Assuming the security of Paillier Encryption, digital signature, garbled circuit, and two message oblivious transfer.

3.8 Packing Technique

Here, we presented a light version of the protocol that encapsulates all the ideas used by the authors of [2], in order to build an external-facing biometric authentication scheme.

The primary distinction between these protocols is the utilization of the packing technique described in [preliminaries]. A major issue with the protocol discussed in this section is its bandwidth-consuming nature, particularly the matching process which is intended to be carried out frequently.

Indeed, we can observe that if we have a valid biometric template $\vec{u}$ with a bit length shorter than 16, for example, then the efficiency of the communication is about $16/2048$ when we transmit the ciphered version $\text{Enc}(\vec{u}) = (\text{Enc}(u_1), ..., \text{Enc}(u_n))$ with a Paillier modulus N of size 2048. This means that for every 16 bits of information sent, 2048 bits are transmitted.

However, the use of the packing technique has its drawbacks, first, as discussed in Section [preliminaries] the computation cost for the terminal is higher when using the packing and second, this packing method adds a lot of complexity to the protocol and is not flexible, that is; this packing does not translate well to slight change in the protocol specification.

If we want to allow biometric templates to have negative values, we will encounter difficulties when attempting to adapt the packing technique.

3.9 Implementation results

The authors of Game-set-match [2] implement this protocol and report on its efficency, the test were conducted on a single commodity laptop with an i5-8365U processor and 16GB of memory.The Paillier scheme is implemented using a security level analogous to 3072 bit RSA.

Template Length	Bit Length	Enrollment (ms)	Matching (ms)
128	16	2927	87
	24	2892	74
256	16	5008	119
	24	5053	105
512	16	7666	198
	24	9439	195
1024	16	9242	164
	24	17822	266

CHAPTER 4

Contribution

4.1 Negative value support

Motivation

In this section, we will continue to explore external facing biometric authentication scheme; more precisely, we will build upon the protocol presented in the previous chapter and we will try to solve the issue of negative value support. Indeed, remember from the previous chapter that the external facing biometric authentication protocol proposed in [2] does not support biometric template, or should we say, vectors, with negative values.

We are attempting to construct an external biometric authentication system based on the protocol discussed in the preceding chapter that would support a biometric template $\vec{u} \in \mathbb{Z}^n$.

This presents a huge advantage as most machine learning methods produce face embedding in Euclidian space with negative values; that is, in their framework, a biometric template is a vector $\vec{u} \in \mathbb{R}^n$ with norm 1 [21, 16] and two vectors are similar if their cosine similarity is greater than some threshold.

Furthermore, from this enhanced protocol that supports biometric templates $\vec{u} \in \mathbb{Z}^n$, the protocol can be adjusted to support vectors $\vec{u} \in \mathbb{R}^n$ with norm 1 by simply scaling the vectors $\vec{u}$ and the acceptance threshold accordingly.

The goal is to create a cryptographic protocol that is compatible with deep learning based methods in the setting of external facing biometric authentication, while providing strong privacy for users' biometric data, thus enabling compatibility with the most advanced machine learning techniques.

Natural way to achieve negative value support

Consider the protocol presented in the previous chapter, recall that the protocol supports vectors $\vec{u} \in \mathbb{N}^n$, suppose now that we would like to support vectors $\vec{u} \in \mathbb{Z}^n$ how would we go about doing this?

A potential approach to begin with would be to alter the Paillier plaintext representation space, which is currently $\mathbb{Z}_N = \{0, ..., N-1\}$. Instead, a symmetric representation $\mathbb{Z}_N = \{-\frac{N-1}{2}, ..., 0, ..., \frac{N-1}{2}\}$

could be used, allowing for the encryption of negative values while still permitting homomorphic operations.

We will consider this natural strategy and explore the protocol modifications that would allow this representative change to work.

Enrollment

We first consider the enrollment and the implication of this plain-text representative change.

As discussed above, the paillier plaintext space is now $\mathbb{Z}_N = \{-\frac{N-1}{2}, ..., 0, ..., \frac{N-1}{2}\}$.

For a given positive integer parameter n and ζ, we consider a biometric template $\vec{u}$ to be valid if $\vec{u} \in \mathbb{Z}^n$ and $\|\vec{u}\| = \zeta$. The challenge is to ensure compatibility between this representation and the zero-knowledge protocol $\Pi_{\text{L2-norm-}\mathbb{Z}}$.

Fortunately, it is possible to do so as long as the verifier and the prover agree on the same convention.

Consequently, the enrollment process can be adapted to support the enrollment of the negative value vector $\vec{u}$ if the service provider and the device agree on the same convention of plaintext space.

Matching

Now that we have addressed enrollment, we explore how to adapt the matching part of the protocol.

Recall that during the maching protocol between the terminal and the device, the Terminal has input $\vec{w}$ receives $(\vec{c}, N, \sigma)$ from the device and computes the mac values A, B defined as $A = \text{Enc}[a_1\langle\vec{u}, \vec{w}\rangle + a_0]$ and $B = \text{Enc}[\langle\vec{u}, \vec{w}\rangle + b_0]$ with $a_1 \in \{0, ..., 2^\lambda\}, a_0 \in \{0, ..., 2^{k+2\lambda}\}$ and $b_0 \in \{0, ..., 2^{k+\lambda}\}$ uniformly sampled.

The argument that makes everything work is that once the device deciphers the values A, B to get integers a, b; it does not learn any information since a, b are computationally indistinguishable from uniformly sampled integers (for well-chosen k).

Here, the argument still holds; first consider an integer k such that $-2^{k-1} \leq \langle\vec{u}, \vec{w}\rangle \leq 2^{k-1}$ then for uniformly sampled a_1, a_0, b_0 (defined as before) integer $a = a_1\langle\vec{u}, \vec{w}\rangle + a_0$ is computationally indistinguishable from a uniformly sampled $k + 2\lambda$ bit positive integer and $\langle\vec{u}, \vec{w}\rangle + b_0$ is computationally indistinguishable from a uniformly sampled $k + \lambda$ bit positive integer.

This makes the MAC introduced in the previous chapter compatible with negative inner product $\langle\vec{u}, \vec{w}\rangle$.

Finally, the terminal and the device execute a garbled circuit-based MPC protocol to determine if there is a match. This protocol is compatible with the change of presentative space if and only if the garbled circuit supports negative values, meaning that the integers must be signed. To make this possible while keeping a similar circuit complexity, a two's complement bit representation can be used.

We can conclude from this analysis that we can modify the matching component of the protocol discussed in the preceding chapter to enable negative value vectors and achieving our goal.

The protocol

In this section, we will explain the protocol according to the preceding conversation.

Protocol parameters

The protocol has public parameters $(\zeta, n, \lambda, k, t)$ such that:

For every Paillier public key N (for the security parameter λ) and for every biometric template $\vec{u}, \vec{w} \in \mathbb{Z}^n$ such that $\|\vec{u}\| = \|\vec{w}\| = \zeta$ we have $-2^{k-1} \leq \langle \vec{u}, \vec{w} \rangle \leq 2^{k-1}$ and $2^{\lambda+2k} < \frac{N-1}{2}$.

These conditions are simply here to prevent any overflow of the MAC values.

In practice, we can select the parameter k based on ζ. Indeed, we have :

$$\|\vec{u} - \vec{w}\|^2 = \|\vec{u}\|^2 - 2\langle \vec{u}, \vec{w} \rangle + \|\vec{w}\|^2$$

since $0 \leq \|\vec{u} - \vec{w}\|^2 \leq \|\vec{u}\|^2 + \|\vec{w}\|^2$ and $\|\vec{u}\|^2 = \|\vec{w}\|^2 = \zeta^2$ we have :

$$0 \leq \|\vec{u}\|^2 - 2\langle \vec{u}, \vec{w} \rangle + \|\vec{w}\|^2 \leq 2\zeta^2$$

$$-2\zeta^2 \leq \|\vec{u}\|^2 + \|\vec{w}\|^2 - 2\zeta^2 \leq 2\langle \vec{u}, \vec{w} \rangle \leq \|\vec{u}\|^2 + \|\vec{w}\|^2 \leq 2\zeta^2$$

and finally

$$-\zeta^2 \leq \langle \vec{u}, \vec{w} \rangle + \leq \zeta^2$$

fixing k to be $\lceil 2\log_2(\zeta) \rceil$ is in practice optimal.

As a result, the condition $2^{\lambda+2k} < \frac{N-1}{2}$ translates to $\zeta < \sqrt{\frac{N-1}{2^{\lambda+1}}}$ (which in practice is not limiting due to the natural big size of the paillier modulus).

Finally, as before, t represents the acceptance threshold.

Enrollment

Parameters

During the enrollment, **D** has input vector $\vec{u} \in \mathbb{Z}^n$ such that $\|\vec{u}\| = \zeta$. We suppose that **SP** has public and private signing key pairs (psk,ssk).

The enrollment between **D** and **SP** goes as follows :

- **D** generates a pair of Paillier Encryption keys $(N, \phi(N))$ for some security parameter λ.
- **D** computes and sends the ciphered biometric template using paillier encryption : $\vec{c} = (\text{Enc}(u_1, r_1), ..., \text{Enc}(u_n, r_n))$ along with the public key N .
- Execute ZK proofs : $\Pi_{\text{L2-norm-}\mathbb{Z}}(\zeta, \vec{c})$, $\Pi_{\text{plain-text knowledge}}(\vec{c})$ and $\Pi_{\text{correct-key}}(N)$ With **D** as the prover, **SP** as the verifier and $(x_i, r_i)_i$ as the witness with convention that $\mathbb{Z}_N = \{-\frac{N-1}{2}, ..., \frac{N-1}{2}\}$.
- If one of the proofs is not correct, **SP** ends the protocol.
- Otherwise: **SP** signs $(\text{Enc}[\vec{u}], N)$ and sends the signature σ to **D**.

Matching

Parameters

During the matching part, we assume that **T** has public signing key psk, valid biometric template $\vec{w}$. We also assume that some public threshold value t is known to both parties.

The matching between **D** and **T** goes as follows :

- **D** sends the ciphered biometric template along with the Paillier public key and the associated signature from **SP** to **T** : $((\text{Enc}[\vec{u}], N), \sigma)$.
- **T** upon receiving values $((\vec{c}, N), \sigma)$ does the following :

  - If the signature is not valid, ends the protocol : Verify$(psk, (N, [\vec{u}]), \sigma) \stackrel{?}{=} 1$.
  - homomorphically computes the value Enc$(\langle \vec{u}, \vec{w} \rangle)$ representing the cosine similarity between vectors $\vec{u}$ and $\vec{w}$.
  - samples $a_1 \in \{0, ..., 2^{\lambda}\}, a_0 \in \{0, ..., 2^{k+2\lambda}\}$ and $b_0 \in \{0, ..., 2^{k+\lambda}\}$ uniformly at random.
  - Computes and sends to **D** the MAC values $A = \text{Enc}[a_1 \langle \vec{u}, \vec{w} \rangle + a_0]$ and $B = \text{Enc}[\langle \vec{u}, \vec{w} \rangle + b_0]$.

- **D** deciphers values A, B to get integer values a, b while taking into account that $\mathbb{Z}_N = \{-\frac{N-1}{2}, ..., \frac{N-1}{2}\}$.
- **D** and **T** run a garbled circuit based multi party computation with (a_1, a_0, b_0) as private input for **T**, (a, b) as private input for **D** and the function f_t as the circuit. At the end of this **T** learns a bit e representing weather or not $\langle \vec{u}, \vec{w} \rangle \geq t$.
- **T** output "success" if $e = 1$ else **T** outputs "failure".

Incompatibility with the packing technique

Recall from [preliminaries] that the packing technique only allows packing of small integer in the setting of paillier encryption. Here we cannot use this technique anymore, as we have changed the paillier representative space. However, we can change the protocol, in order to have packing support while allowing negative values in the biometric template.

Negative value support compatible with the packing technique

To enable the packing technique to be utilized, we will homomorphically apply a clever transformation to the encrypted biometric template. Furthermore, we will adjust the garbled circuit to achieve biometric authentication.

Clever transformation

Suppose $\vec{u}, \vec{w} \in \mathbb{Z}^n$ biometric templates of norm ζ and t some threshold value.
we have the following equivalent inequalities :
$\langle \vec{u}, \vec{w} \rangle \geq t$
$\langle \overrightarrow{u} + \zeta \vec{1}, \overrightarrow{w} + \zeta \vec{1} \rangle \stackrel{?}{\geq} t + \zeta \sum_i (u_i + w_i + \zeta)$
Finally, by denoting $\overrightarrow{U} = (\overrightarrow{u} + \zeta \overrightarrow{1})$ and $\overrightarrow{W} = (\overrightarrow{v} + \zeta \overrightarrow{1})$ the shifted vector, we have :

$$\langle u, v \rangle \stackrel{?}{\geq} t \Leftrightarrow \langle \overrightarrow{U}, \overrightarrow{V} \rangle + (n\zeta^2 - t) \stackrel{?}{\geq} \zeta \sum_i (U_i + V_i)$$

The key here is that the values U_i and W_i are positive integers less than 2ζ, allowing us to use the packing technique on the vectors $\vec{U}$ and $\vec{W}$.

The protocol

The protocol has public parameters $(\zeta, n, \lambda, k, k', t)$ such that:

For every Paillier public key N (for the security parameter λ) and for every biometric template $\vec{u}, \vec{w} \in \mathbb{Z}^n$ such that $\|\vec{u}\| = \|\vec{w}\| = \zeta$ we have $-2^{k-1} \leq \langle \vec{U}, \vec{W} \rangle \leq 2^{k-1}$ and $2^{\lambda+2k} < \frac{N-1}{2}$. We also

declare that these two vectors are similar if the inner product of $\vec{u}$ and $\vec{w}$ is greater than or equal to a certain threshold t, which is used to determine whether a match is made during the matching phase of the protocol.

Enrollment

Parameters

During the enrollment, **D** has input vector $\vec{u} \in \mathbb{Z}^n$ such that $\|\vec{u}\| = \zeta$. We suppose that **SP** has public and private signing key pairs (psk,ssk).

The enrollment between **D** and **SP** goes as follows :

- **D** generates a pair of Paillier Encryption keys $(N, \phi(N))$ for some security parameter λ.
- **D** computes and sends the ciphered biometric template using paillier encryption: $\vec{c} = (\text{Enc}(u_1, r_1), ..., \text{Enc}(u_n, r_n))$ along with the public key N .
- Execute ZK proofs : $\Pi_{\text{L2-norm-}\mathbb{Z}}(\zeta, \vec{c})$, $\Pi_{\text{plain-text knowledge}}(\vec{c})$ and $\Pi_{\text{correct-key}}(N)$ With **D** as the prover, **SP** as the verifier and $(x_i, r_i)_i$ as the witness with the convention that $\mathbb{Z}_N = \{-\frac{N-1}{2}, ..., \frac{N-1}{2}\}$.
- If one of the proofs is not correct, **SP** ends the protocol.
- Otherwise: **SP** does the following :
  - homomorphically apply an isometric to $\vec{u}$ to get the an encryption of $\vec{U}$
  - Computes homorphically an encryption of $\sum_i U_i$
  - signs $(\text{Enc}[\vec{U}], \text{Enc}[\sum_i U_i], N)$ and sends them along with the signature σ to **D**

Matching

During the matching part, we assume that **T** has public signing key psk, valid biometric template $\vec{w}$, and associated shifted vector $\vec{W}$. We also assume that both parties know $\mathcal{T} := (n\zeta^2 - t)$.

The matching between **D** and **T** goes as follows :

- **D** sends $((\text{Enc}[\vec{U}], \text{Enc}[\sum_i U_i], N), \sigma)$ to **T**.
- **T** upon receiving values $(((\vec{c}, s, N), \sigma)$ does the following :
  - If the signature is not valid, ends the protocol : Verify(psk, $(\vec{c}, s, N), \sigma) \stackrel{?}{=} 1$.
  - Homomorphically computes the value $\text{Enc}(\langle \vec{U}, \vec{W} \rangle)$ representing the cosine similarity between vectors $\vec{U}$ and $\vec{W}$.
  - samples $a_1 \in \{0, ..., 2^{\lambda}\}, a_0 \in \{0, ..., 2^{k+2\lambda}\}$ and $b_0 \in \{0, ..., 2^{k+\lambda}\}$ uniformly at random.
  - samples $x_1 \in \{0, ..., 2^{\lambda}\}, x_0 \in \{0, ..., 2^{k'+2\lambda}\}$ and $y_0 \in \{0, ..., 2^{k'+\lambda}\}$ uniformly at random.
  - Computes and sends to **D** the MAC values $A = \text{Enc}[a_1\langle \vec{U}, \vec{W} \rangle + a_0]$, $B = \text{Enc}[\langle \vec{U}, \vec{W} \rangle + b_0]$ $X = \text{Enc}[x_1(\zeta \sum_i U_i + W_i) + x_0]$ and $Y = \text{Enc}[(\zeta \sum_i U_i + W_i) + y_0]$.
- **D** deciphers values A, B, X, Y to get integer values a, b, x, y using the convention $\mathbb{Z}_N = \{-\frac{N-1}{2}, ..., \frac{N-1}{2}\}$.

- **D** and **T** run a garbled circuit-based multiparty computation with $(a_1, a_0, b_0, x_1, x_0, y_0)$ as private input for **T**, (a, b, x, y) as private input for **D** and the function $F_{\mathcal{T}}$ as the circuit. At end of this both **T** and **D** learns a bit e representing weather or not $\langle \vec{u}, \vec{w} \rangle \geq t$.

- **T** output "success" if $e = 1$ else **T** outputs "failure".

The circuit to evaluate

Let us describe the high-level functionality of the circuit $F_{\mathcal{T}}$ used during the matching phase between **T** and **D**.

Inputs : $(a, b, x, y, a_0, b_0, a_1, x_0, y_0, x_1)$
Computation :

- set inner_product $= b - b_0$;

- set $a' = a_1 \times$ inner_product $+ a_0$

- set sum $= y - y_0$

- set $x' = x_1 \times$ sum $+ x_0$

- return 1 if $(a' = a)$ and $(x = x')$ and (inner_product $+ \mathcal{T} \geq$ sum) otherwise 0

This circuit aims to verify that the inner product between the reference biometric template $\vec{u}$ and the recorded biometric template $\vec{w}$ is greater than some fixed threshold t.

We apply a transformation to the vectors, which leads to the need to verify a different inequality in the garbled circuit that uses an additional value. Consequently, we must authenticate, through the MAC mechanism we described, this additional value, thus doubling the number of variables for both parties. Moreover, from the inequality we demonstrate in [clever transformation], we know that the functionality is correct; that is, at the end of the computation, we learn the predicate $\langle \vec{u}, \vec{w} \rangle > t$.

Mac Values

Here, just like in the previous protocols, we verify a Mac within the garbled circuit, acting as an authentication mechanism. Here, we add an additional value, namely the value $S := \zeta \sum_i U_i + W_i$ and, in order to hide the value S, we need to ensure that $S < 2^{k'}$. Since we have $S < 4n\zeta^2$ we can simply fix $k' = \lceil \log_2(4n\zeta^2) \rceil$.

Paillier Packing

This "clever transformation" may appear peculiar, and when examining the protocol, it may appear to be of no use. However, the purpose is to utilize the packing technique in combination with the protocol to obtain improved bandwidth utilization. We won't go into the details as it would make this presentation too long, but it is easy to see that the packing technique can be used to achieve improved communication performances.

Implementation results

We implemented the protocol described in this section. Specifically, our implementation employed the paillier packing technique and as a result is a slight variation of the protocol described in this section.

There are multiple cryptographic tools used in this protocol, making it a challenging project.

We chose C++ for our implementation because it is a high-performance compiled language that provides a wide range of standard libraries to work with, as well as a great number of open source projects in the cryptography domain.

Multi party computation

The use of garbled circuit-based multiparty computation in the protocol turned out to be challenging.

There is limited implementation in this area, and we looked at a variety of open source projects to incorporate into our protocol. Unfortunately, most of them were inadequately documented, making it difficult to get started.

We had to take into account the requirement for large integer support (more than 64 bits) and compatibility with C++, which narrowed down the potential candidate.

After a comprehensive examination, and with the help of the benchmarking and documentation work from MP-SPDZ [14], we concluded that TinyGarble [22], TinyGarble2 [12], ABY2.0[20], Obliv-C [25] and EMP Toolkit [24] were the most suitable projects for our protocol, given our limitations.

We eventually selected the TinyGarble[22] project because it was both efficient and had the most comprehensive documentation; moreover, it was capable of performing big integer arithmetic but would require designing a circuit in Verilog and then synthesizing it using either Yosys or Synopsys Design as a downside.

Digital signature

For digital signature we used the OpenSSL implementation for both hashing and the signature. The signature that we used is RSASSA-PKCS1-v1_5 as specified in RFC 3447.

Paillier Cryptosystem

There are a few open source libraries that implement the Paillier cryptosystem, such as the Intel Paillier Cryptosystem Library, which is optimized for x86 Intel processors.

We decided to create our own implementation for a couple of reasons. Primarily, it provides us with more flexibility, which is essential since we are using a packing technique and a variety of custom zero-knowledge proofs. Additionally, most libraries do not have the optimized version of the encryption algorithm under the private key presented in [Encryption under the private key optimization], making them slower than our implementation for our use case.

We employ the open source GNU Multiple Precision Arithmetic Library with a C++ wrapper that provides operator overload for large integer calculations. Additionally, since the protocol is conducted in a network environment, we must serialize the large integers in order to transmit them. To accomplish this, we have developed our own serialization protocol based on the Boost C++ serialization library.

Test setting

We assess the performance of both the enrollment and matching protocols in a local network environment.

We conducted our evaluations on a single laptop with an Intel i5-10310U processor running at 1.70GHz and 8GB of RAM. We employed a Paillier modulus of 2048 bits, which is the minimum key size recommended by NIST.

We simply note that we use the L2 norm zero-knowledge proof introduced in the paper Game-Set-MATCH [2] even if we believe that their proof is not complete as discussed in a future section. We investigated templates of various lengths, such as 128, 256, 512, and 1024, with different bit lengths, including 16 and 32.

Results

Template Length	Bit Length	Enrollment (ms)	Matching (ms)
128	16	2,338	174
	32	1,964	237
256	16	4,068	225
	32	4,064	200
512	16	7,666	198
	32	7,730	231
1024	16	17,170	240
	32	15,341	290

The enrollment time increases in proportion to the template length n. As anticipated, the enrollment is the most expensive part of the protocol, since it is meant to be done only once, while the matching is designed to be done multiple times. The time for the matching indicates that this protocol can provide a satisfactory user experience, since it takes less than 300 ms to complete.

Future work

We built upon the work of Game-Set-MATCH [2] to create an authentication protocol that would enable the use of negative value biometric templates. We initially constructed a basic version of the protocol and then extended it to incorporate the packing technique. We then implemented this version of the protocol, providing implementation results.

We did not demonstrate security with respect to the biometric authentication functionality introduced in Game-Set-MATCH [2] and leave this for future work. This protocol is quite analogous and the demonstration will be nearly identical.

4.2 Encryption under the private key optimization

When it comes to cryptography, there are great benefits in making algorithms faster, as they are used so frequently.

One of the primary algorithms used in the Paillier cryptosystem is the encryption map E_N.

The encryption map does not necessitate the possession of the private key, yet, frequently ciphering is done with the knowledge of the private key.

In this particular instance, encryption can be accelerated due to the Chinese Remainder Theorem as the knowledge of the private key implies knowledge of the factorization of N.

In this instance, the most costly part of encryption is to compute the value of $r^N \bmod N^2$ when r is randomly sampled over the set $\mathbb{Z}_N$.

We present an algorithm that is optimized for this particular situation and is two times faster than the best known algorithm.

The algorithm

We present the classical algorithm used to compute a uniformly sampled element in the set $\{r^N \bmod N^2 | r \in \mathbb{Z}_N\}$ when the factorization of N is known.

We then present an improved version of the algorithm that increases its speed by a factor of two under the same assumptions and without any security drawbacks.

We demonstrate its correctness and compare it with the traditional algorithm.

Finally, to demonstrate its effectiveness, we benchmark our solution.

Traditional algorithm [19]

The traditional approach involves the following steps:

1. uniformly sample $(r_p, r_q) \leftarrow \mathbb{Z}_p \times \mathbb{Z}_q$
2. using modular exponentiation compute $R_p = r_p^N \bmod p^2$ and $R_q = r_q^N \bmod q^2$.
3. compute $R = R_p U + R_q V \bmod N^2$ the Chinese Reminder theorem value.

This is a classical use of the chinese remeinder theorem, levereging the knowledge of the factorisation of N to speed up the modular exponentiation.

Our proposal algorithm

Now consider the algorithm.

1. uniformly sample $(r_p, r_q) \leftarrow \mathbb{Z}_p \times \mathbb{Z}_q$
2. compute $R_p = r_p^p \bmod p^2$ and $R_p = r_q^q \bmod q^2$
3. compute $R = R_p U + R_q V \bmod N^2$ the Chinese Reminder Theorem value

We argue that this algorithm achieves the desired functionality, that is, the algorithm outputs a uniformly sampled element in the set $\{r^N \bmod N^2 | r \in \mathbb{Z}_N\}$.

Let us first prove the correctness and show that the output distribution is uniform across the set $\{r^N \bmod N^2 | r \in \mathbb{Z}_N\}$. Consider $R := r^N \bmod N^2$, we show that there is a unique r_p, r_q such that the output of the algorithm is R. It is evident that, due to the Chinese remainder theorem, it is enough to demonstrate the existence of an integer r_p for which $r_p^p \equiv r^N \pmod{p^2}$ (resp r_q).

It is known that p and $p-1$ are relatively prime, so there exist u and v such that $up+v(p-1) = 1$. Consequently, $r_p = r^{nu} \bmod p^2$ is a valid solution.

This is because $r_p^p = r^{nup} = r_p^{n(1-v(p-1))} = r^n \bmod p^2$.

A similar argument can be made when considering $\bmod q$, which leads to the conclusion.

Performance comparison

When assessing the performance of these two algorithms, the only distinction is the exponent in the modular exponentiation. The traditional algorithm necessitates two exponentiations to the power of N, while the optimized algorithm necessitates two modular exponentiations, one to the power of p and the other to the power of q. Since the time complexity of modular exponentiation is proportional to the size of the exponent and N is twice as large as p and q, it can be inferred that the optimized algorithm is roughly twice as fast.

This is further confirmed by the implementation result and a more thorow analysis would lead the same conclusion.

Implementation results

We compare the two implementations of the algorithm and analyze their respective time efficiency.

We implemented the 2 algorithm in C++ using the The GNU Multiple Precision Arithmetic Library.

bit length	time (ms)
2048 (classical)	5.346
2048 (optimized)	2.682
3072 (classical)	15.077
3072 (optimized)	7.879

4.3 Plain Text knowledge

The authors of Game-Set-MATCH [2] proposed an Honest Verifier zero-knowledge proof of knowledge that demonstrates the knowledge of plain text in the context of paillier encryption.

In this section, we provide a novel proof of this result.

Plain Text knowledge protocol

This protocol ensures that the prover knows the underlining plaintext of a set of paillier ciphertext.

Parameters:

- A paillier public key: N.
- The statement : $\vec{c} = (c_1 = E(m_1, r_1), c_2 = E(m_2, r_2), ..., c_n = E(m_n, r_n)) \in (\mathbb{Z}_{N^2}^{\times})^n$.
- The witness $(m_1, r_1), ..., (m_n, r_n) \in \mathbb{Z}_N \times \mathbb{Z}_N^{\times}$.

Protocol $\Pi_{\text{plain-text-knowledge}}$

- **P** uniformly samples $(m, r) \in \mathbb{Z}_N \times \mathbb{Z}_N^{\times}$ and sends $c = E(m, r)$.
- **V** uniformly samples and sends $\vec{e} = (e_1, ..., e_n) \in \mathbb{Z}_N^n$
- **P** sends $M = m + \sum_i e_i m_i \bmod N; R = r \prod_i r_i^{e_i} \bmod N$
- **V** accepts if and only if $E(M, R) = c \prod c_i^{e_i} \bmod N^2$

This version of the protocol is a batch version of the classical plain text knowledge proof [7] presented in [2]. It leverages the linear structure of the original proof to achieve better communication performance, only requiring the prover to send a constant number of messages.

Furthermore, this protocol can be made noninteractive using the Fiat-Shamir heuristic, making the communication complexity constant (independent on n).

Proof

Completeness is clear from the description of the protocol, we prove soundness and zero-knowledge.

Soundness

We first need the following lemma :

Lemma 4.3.1 (Probability that a random matrix of size n is non singular in $\mathbb{Z}_N$ [17])**.** If $N = pq$ an RSA modulus with $p > q$, the probability for a random matrix of size n to be non-singular is $\prod_{i=0}^{n-1} \frac{p^n - p^i}{p^n} \prod_{i=0}^{n-1} \frac{q^n - q^i}{q^n} \geq e^{-\frac{2n}{q-1}}$.

Suppose that the prover can correctly answer $n+1$ challenges for commitment $c = E(m, r)$.

For $j \in \{0, 1, ..., n\}$ we write these challenges $\vec{e_j}$ and the associated answers M_j, R_j.

We now show that he can recover $\vec{r} = (r_1, ..., r_n)$ the randomness used in the encryption of the statement $\vec{c} = (c_1, ..., c_n)$ (in polynomial time).

Consider the matrix $E = \begin{bmatrix} \vec{e_0} - \vec{e_1} \\ \vdots \\ \vec{e_0} - \vec{e_n} \end{bmatrix}$, which is a random matrix in $\mathbb{Z}_N$. According to the previous Lemma, this matrix is nonsingular with overwhelming probability.

As a result, we can consider the unique vector $\vec{v}$ such that $E\vec{v}^T = (1, 0, ..., 0)$, integers $(k_1, ..., k_n)$ such that we have: $\langle \vec{v}, \vec{e_0} - \vec{e_1} \rangle - k_1 N = 1$ and for $j \in \{2, ..., n\}$ $\langle \vec{v}, \vec{e_0} - \vec{e_j} \rangle - k_j N = 0$.

we have the following that holds :

$$r_1 \prod_{j=1}^{n} r_j^{k_j N} = \prod_{j=1}^{n} (R_0/R_j)^{v_i} \mod N$$

Since the prover can compute both $(k_1, ..., k_n)$ and r_i^N (which is just the value $c_i \bmod N$) he is able to compute $r_1 = \prod_{j=1}^{n} (R_0/R_j)^{v_i} c_j^{-k_j N} \mod N$. Finally, he can recover r_1 and then decipher c_1 to recover m_1.

Proceeding in the same way, he can recover the other witnesses.

We can conclude that the probability of a cheating prover who is unaware of $(m_i, r_i)_i$ being able to correctly answer a randomly chosen challenge $\vec{e}$ is negligible (if N has large prime factors, which is the case in practice).

Zero Knowledge

Notice that a real transcript is a tuple: $(c, \vec{e}, M, R)$ with $(\vec{e}, M, R)$ uniformly sampled in $\mathbb{Z}_N^n \times \mathbb{Z}_N \times \mathbb{Z}_N^\times$ such that the following equation holds : $E(M, R) = c \prod c_i^{e_i} \mod N^2$.

We now describe the simulator's strategy to output a transcript that would be statistically indistinguishable from a real transcript.

The simulator first uniformly sample $M, R \in \mathbb{Z}_N \times \mathbb{Z}_N^\times$, $\vec{e} \in \mathbb{Z}_N^n$ and sets $c = E(M, R) \prod_i c_i^{-e_i} \bmod N^2$. Finally the simulator outputs $(c, \vec{e}, M, R)$.

This transcript is statistically indistinguishable from a real transcript, since the distribution of $(\vec{e}, M, R)$ is identical to the distribution of a real transcript and the equation $E(M,R) = c\prod c_i^{e_i} \mod N^2$ holds by construction.

4.4 L2 norm in $\mathbb{Z}$ and $\mathbb{Z}_N$

Consider the following situation: We have n paillier cipher texts Enc$[x_1], ...,$ Enc$[x_n]$, an integer $y < N$ and we wish to prove in zero-knowledge fashion that the L2 norm of $\vec{x} = (x_1, ..., x_n)$ is equal to y over the integers : $\sum_i x_i^2 = y$ $(x_i \in [0, N-1])$.

This situation naturally arises when dealing with ciphered vectors; however ,building such a zero-knowledge proof is challenging, since the equality has to hold over the integers.

L2 norm in $\mathbb{Z}$

The authors of [2] propose a zero-knowledge proof that achieves this.

The core of their idea is to introduce a prime number $\hat{p}$ in the challenge, once the Prover receives this prime number, he is tasked with proving that $\sum_i x_i^2 = y \mod N\hat{p}$, this effectively forces a cheating prover to guess $\hat{p}$ in order to dupe the verifier and, of course, can only happen with negligible probability.

This zero-knowledge proof is fast in practice but cannot be made non-interactive as a drawback.

In the following we describe this zero-knowledge proof; for further reading and proof, see [2].

parameters:

- A paillier public key N
- A public integer value $y \in \mathbb{Z}$
- statement : $c_1 = E(x_1, r_n), ..., c_n = E(x_n, r_n) \in \mathbb{Z}_{N^2}^{\times}$
- witness $(x_1, r_1), ..., (x_n, r_n) \in \mathbb{Z}_N \times \mathbb{Z}_N^{\times}$

protocol $\Pi_{\text{L2-norm-}\mathbb{Z}}$:

- V uniformly sample a prime $\hat{p} < N$
- V uniformly sample values : $\alpha \in Z_{2^\lambda}$, $\rho_1, ...\rho_n, \beta, \hat{\beta} \in \mathbb{Z}_N$. Then it sends P the following challenge:
  - Enc$[w_i]$ = Enc$[\alpha x_i + \rho_i]$ for all i
  - Enc$[v]$ = Enc$[\sum_i(-2\alpha\rho_i x_i - \rho_i^2) + \beta]$
  - Enc$[v_p]$ = Enc$[\sum_i(\gamma_i x_i) + d + \hat{\beta}]$ where $\gamma_i := (-2\alpha\rho_i \mod \hat{p})$, $d := (\sum_i -\rho_i^2) \mod \hat{p}$
- P deciphers Enc$[w_i]$, Enc$[v]$, Enc$[v_p]$ and sends the following to V :
  - $z := v + \sum w_i^2 \mod N$
  - $\hat{z} := \hat{v} + \sum w_i^2 \mod \hat{p}$
- V accept if $z = y\alpha^2 + \beta$ and $\hat{z} = y\alpha^2 + \hat{\beta}$

Flaw in the soundness argument

Here, we explore what we believe to be a flaw in the soundness argument of this zero-knowledge proof. Remember that soundness is the property of a zero-knowledge proof that ensures that a dishonest prover cannot convince a verifier of something that is false.

Let us first review the soundness argument given in *Game-Set-MATCH*[2].

The argument goes as follows.

Suppose a cheating prover has the statement $c_1, ..., c_n$ and witness $(x_i, r_i)_i$ with $x_i \in [0, N-1]$ such that $\sum_i x_i \neq y$. Then there exists an integer $e \neq 0$ such that $\sum_i x_i^2 + e = y$.

The authors then argue that if the prover is able to compute $\hat{z}$ such that the verification $\hat{z} = y\alpha^2 + \hat{\beta}$ succeeds, then we have :

$$\hat{z} = \alpha^2 \sum_i x_i^2 + \hat{\beta} + e\alpha^2 \bmod \hat{p}$$

If we note $\gamma = \alpha^2 \sum_i x_i^2 + \hat{\beta}$ the equation simply becomes:

$$\hat{z} = \gamma + e\alpha^2 \bmod \hat{p}$$

The argument goes on to say that the prover can calculate γ by using the encrypted values of w_i and v_p. Furthermore, he can compute e and has the value of $\hat{z}$, which enables him to obtain α (as it is assumed to be less than $\hat{p}$). Finally, with the equation $z = y\alpha^2 + \beta$, he can recover β. Since this is only possible with a negligible probability, it must be that $e = 0 \bmod \hat{p}$. The flaw in the argument is that the prover is able to compute the values γ. This is only guaranteed when the computation of w_i does not overflow, that is, $\alpha x_i + \rho_i = w_i$ over the integers, if this condition is not met, then it is not so clear how the prover would go about computing the values of γ and the soundness argument does not hold anymore. However, one can argue that, if the integers overflow, that is, the prover is not able to recover $\alpha x_i + \rho_i$ from w_i, then he does not have enough *information* to be able to compute the value $\hat{z}$. However, this argument is heuristic at best and does not constitute a proof.

Counter-example

Consider integers ($x_i \in [0, N-1]$) such that $\sum_i x_i^2 = y$.

Now consider the statement $c_1, ..., c_n$ defined as $c_i = \text{Enc}(N - x_i, r_i)$ for some randomness r_i. First, we note that $\sum_i (N - x_i)^2 = y \bmod N$ enables the prover to compute z on any challenge from the verifier. Furthermore, we have $\sum_i (N - x_i)^2 \neq y$ such that the statement is false (y is always less than N).

However, running the proof (in an honest manner) with this statement generates an accepting proof; that is, the prover with statement $c_1, ..., c_n$ and witness $(N - x_1, r_1), ..., (N - x_n, r_n)$ is able to convince an honest verifier that $\sum_i (N - x_i)^2 = y$.

The key here is that the values w_i are now equal to $\rho_i - \alpha x_i$ over the integers with overwhelming probability. In the same way, the value v_p now equals $\hat{\beta} - \sum_i (\gamma_i x_i) + d$ over the integers with overwhelming probability.

Finally, the value of $\hat{v} + \sum w_i^2$ is equal to $y\alpha^2 + \hat{\beta}$ (modulo $\hat{p}$) allowing the cheating prover to generate a false proof for the statement $c_1, ..., c_n$.

4.5 L2 norm in $\mathbb{Z}_N$

Let us present a variant of this proof that is more natural to build.

Namely an L2 norm proof mod N : Given Enc$[x_1], ..., Enc[x_n]$, an integer $y < N$, we wish to prove in zero knowledge fashion that the L2 norm of $\vec{x} = (x_1, ..., x_n)$ is equal to y over $\mathbb{Z}_N$ $\sum_i x_i^2 = y \bmod N$.

To build this proof, we are going to take inspiration from paillier multiplication proof [8] and proof of n-th root [8].

We present the protocol and further provide formal proof for completeness, soundness, and zero knowledge.

parameters:

1. A paillier public key N
2. A public integer y $\in \mathbb{Z}_n$
3. statement : $c_1 = E(x_1, r_n), ..., c_n = E(x_n, r_n) \in \mathbb{Z}_{N^2}^{\times}$
4. witness $(x_1, r_1), ..., (x_n, r_n) \in \mathbb{Z}_N \times \mathbb{Z}_N^{\times}$

protocol $\Pi_{\text{L2-norm-}\mathbb{Z}_n}$:

- P uniformly samples n random values $t_0, ..., t_n \in Z_n$ and computes $t = t_1x_1 + t_2x_2 + ... + t_nx_n$.
- P commits to cipher-texts values $T_1, T_2, ..., T_n, T =$ Enc$[t_1, \tau_1]$, Enc$[t_2, \tau_2]$, ..., Enc$[t_n, \tau_n]$, Enc$[t, \tau]$
- V sends challenge $e \in Z_n$
- P computes $(z_i, \zeta_i) = (t_i + ex_i \bmod N, \tau_i r_i^e \bmod N)$ $\zeta = \tau^{-1} \prod_i r_i^{e_i} \bmod N$ and sends $((z_1, \zeta_1), ..., (z_2, \zeta_2), \zeta)$
- V accepts if and only if the following equality hold Enc$[z_i, \zeta_i] = T_i c_i^e \mod N^2$ and Enc$[ey, \zeta]T = c_1^{z_1} c_2^{z_2} ... c_n^{z_n} \mod N^2$

This protocol can be seen as a batch version of the multiplication proof.

Completeness

Completeness is clear from the protocol description.

Soundness

Suppose the prover can answer two challenges e, e' for commitment $T_1, T_2, ..., T_n, T$. We note the associated response $(z_i, \zeta_i)_i, \zeta$ and $(z'_i, \zeta'_i)_i, \zeta'$.

This means that the verification equation holds for e, e' :

$$\forall i \in \{1, ..., n\}; \text{Enc}[z_i, \zeta_i] = T_i c_i^e \mod N^2$$

$$\forall i \in \{1, ..., n\}; \text{Enc}[z'_i, \zeta'_i] = T_i c_i^{e'} \mod N^2$$

$$\text{Enc}[ey, \zeta]T = c_1^{z_1} c_2^{z_2} ... c_n^{z_n} \mod N^2$$

$$\text{Enc}[e'y, \zeta']T = c_1^{z'_1} c_2^{z'_2} ... c_n^{z'_n} \mod N^2$$

it follows :

$$\forall i \in \{1, ..., n\}; \text{Enc}[z_i, \zeta_i]\text{Enc}[z'_i, \zeta'_i]^{-1} = c_i^{e-e'} \mod N^2$$

$$\text{Enc}[ye, \zeta]\text{Enc}[ye', \zeta']^{-1} = c_1^{z_1 - z'_1} c_2^{z_2 - z'_2} ... c_n^{z_n - z'_n} \mod N^2$$

this means that :

$$z_i - z'_i = (e - e')x_i \bmod N$$
$$y(e - e') = \sum_i (z_i - z'_i)x_i \mod N$$

And we finally with overwhelming probability :

$$y = \sum_i x_i^2 \mod N$$

In conclusion, if the prover can answer 2 challenges for some commitment, then $y = \sum_i x_i^2 \mod N$, which means that the prover can cheat with probability at most negligible probability.

Zero Knowledge

A transcript is a tuple $T_1, T_2, ..., T_n, T, e, z_1, \zeta_1, ..., z_2, \zeta_2, \zeta$ such that the following equations hold :

$$\forall i \in \{1, ..., n\}; \text{Enc}[z_i, \zeta_i] = T_i c_i^e \mod N^2$$
$$\text{Enc}[ey, \zeta]T = c_1^{z_1} c_2^{z_2} ... c_n^{z_n} \mod N^2$$

The simulator strategy on input e is to sample uniformly $z_i, \zeta_i \in \mathbb{Z}_N \times \mathbb{Z}_N^\times$, $\zeta \in \mathbb{Z}_N^\times$ then set :

$$\forall i \in \{1, ..., n\}; T_i = \text{Enc}[z_i, \zeta_i] c_i^{-e} \mod N^2$$
$$T = c_1^{z_1} c_2^{z_2} ... c_n^{z_n} \text{Enc}[ey, \zeta]^{-1} \mod N^2$$

Further discussion

In practice, we can use this zero-knowledge proof together with a range proof[15] to have an L2 norm proof over the integers. Furthermore, if the zero-knowledge proof presented in Game-Set-MATCH [2] is indeed flawed, then to the best of our knowledge there is no L2 norm over the integer proof that does not use a range proof.

The use of range proof introduces performance issues because even the most efficient range proofs [15] have complexity $\mathcal{O}(n\lambda)$, which in practice will make these kinds of proof λ times slower than the proof in Game-Set-MATCH [2]. The Fiat-Shamir heuristic can be employed to make this proof non-interactive, which is a positive aspect.

CHAPTER 5

Conclusion

In a world increasingly reliant on personal devices, biometric authentication has emerged as a paramount means of ensuring security and user convenience. This thesis has undertaken a comprehensive exploration of the landscape of biometric authentication, with a particular focus on external-facing solutions compatible with deep learning methodologies.

The review of existing literature highlights both the promise and limitations of current biometric authentication approaches. While solutions like Game-Set-MATCH offer a glimpse into a secure, privacy-preserving future, their impracticalities and incompatibility with deep learning algorithms underscore the need for further innovation.

As biometric data security regulations tighten, the imperative for a robust yet adaptable authentication method becomes ever clearer. Our research has taken a significant step towards bridging this gap. By proposing a cryptographic framework that can harmoniously interface with deep learning-based techniques, we seek to pave the way for a new era in biometric authentication—one that is both secure and flexible.

Our thesis has outlined the critical elements of this research journey, from a foundational exploration of cryptographic tools and security requirements to a detailed analysis of the Game-Set-MATCH protocol. We've presented an extended version of the protocol, addressing its limitations and optimizing cryptographic processes for greater efficiency.

Through rigorous implementation and benchmarking, we've demonstrated the feasibility and viability of our proposed approach. This work not only contributes to the ongoing discourse in biometric authentication but also opens avenues for further research and innovation in this dynamic field.

In conclusion, this thesis represents a stride towards secure, adaptable, and privacy-preserving external-facing biometric authentication. By aligning with the power of deep learning, we aim to provide users with a seamless yet robust means of safeguarding their digital identities. As technology continues to evolve, so too must our security measures, and we believe that the integration of biometrics and deep learning is a promising step in that direction.

Bibliography

[1] Shashank Agrawal, Saikrishna Badrinarayanan, Payman Mohassel, Pratyay Mukherjee, and Sikhar Patranabis. Beta: Biometric enabled threshold authentication. Cryptology ePrint Archive, Paper 2020/679, 2020. https://eprint.iacr.org/2020/679.

[2] Shashank Agrawal, Saikrishna Badrinarayanan, Pratyay Mukherjee, and Peter Rindal. Game-set-match: Using mobile devices for seamless external-facing biometric matching. Cryptology ePrint Archive, Paper 2020/1363, 2020. https://eprint.iacr.org/2020/1363.

[3] Pia Bauspieß, Tjerand Silde, Matej Poljuha, Alexandre Tullot, Anamaria Costache, Christian Rathgeb, Jascha Kolberg, and Christoph Busch. Brake: Biometric resilient authenticated key exchange. Cryptology ePrint Archive, Paper 2022/1408, 2022. https://eprint.iacr.org/2022/1408.

[4] Mihir Bellare, Viet Tung Hoang, and Phillip Rogaway. Foundations of garbled circuits. Cryptology ePrint Archive, Paper 2012/265, 2012. https://eprint.iacr.org/2012/265.

[5] Daniel Bleichenbacher. Chosen ciphertext attacks against protocols based on the rsa encryption standard pkcs #1. In Hugo Krawczyk, editor, *Advances in Cryptology — CRYPTO '98*, pages 1–12, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

[6] Tung Chou and Claudio Orlandi. The simplest protocol for oblivious transfer. Cryptology ePrint Archive, Paper 2015/267, 2015. https://eprint.iacr.org/2015/267.

[7] Ronald Cramer, Ivan Damgård, and Jesper Buus Nielsen. Multiparty computation from threshold homomorphic encryption. Cryptology ePrint Archive, Paper 2000/055, 2000. https://eprint.iacr.org/2000/055.

[8] Ivan Damgård and Mads Jurik. A generalisation, a simplification and some applications of paillier's probabilistic public-key system. In Kwangjo Kim, editor, *Public Key Cryptography*, pages 119–136, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[9] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam D. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *CoRR*, abs/cs/0602007, 2006.

[10] Sharon Goldberg, Leonid Reyzin, Omar Sagga, and Foteini Baldimtsi. Efficient noninteractive certification of rsa moduli and beyond. Cryptology ePrint Archive, Paper 2018/057, 2018. https://eprint.iacr.org/2018/057.

[11] Iftach Haitner, Yehuda Lindell, Ariel Nof, and Samuel Ranellucci. Fast secure multiparty ecdsa with practical distributed key generation and applications to cryptocurrency custody. Cryptology ePrint Archive, Paper 2018/987, 2018. https://eprint.iacr.org/2018/987.

[12] Siam Hussain, Baiyu Li, Farinaz Koushanfar, and Rosario Cammarota. Tinygarble2: Smart, efficient, and scalable yao's garble circuit. Cryptology ePrint Archive, Paper 2020/1181, 2020. https://eprint.iacr.org/2020/1181.

[13] A. Juels and M. Sudan. A fuzzy vault scheme. In *Proceedings IEEE International Symposium on Information Theory,*, pages 408–, 2002.

[14] Marcel Keller. MP-SPDZ: A versatile framework for multi-party computation. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020.

[15] Yehuda Lindell. Fast secure two-party ecdsa signing. Cryptology ePrint Archive, Paper 2017/552, 2017. https://eprint.iacr.org/2017/552.

[16] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition, 2018.

[17] Kenneth Maples. Singularity of random matrices over finite fields, 2013.

[18] Ivan De Oliveira Nunes, Peter Rindal, and Maliheh Shirvanian. Oblivious extractors and improved security in biometric-based authentication systems. Cryptology ePrint Archive, Paper 2022/1030, 2022. https://eprint.iacr.org/2022/1030.

[19] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Jacques Stern, editor, *Advances in Cryptology — EUROCRYPT '99*, pages 223–238, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[20] Arpita Patra, Thomas Schneider, Ajith Suresh, and Hossein Yalame. Aby2.0: Improved mixed-protocol secure two-party computation. Cryptology ePrint Archive, Paper 2020/1225, 2020. https://eprint.iacr.org/2020/1225.

[21] Florian Schroff, Dmitry Kalenichenko, and James 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.

[22] Ebrahim M. Songhori, Siam U. Hussain, Ahmad-Reza Sadeghi, Thomas Schneider, and Farinaz Koushanfar. Tinygarble: Highly compressed and scalable sequential garbled circuits. In *2015 IEEE Symposium on Security and Privacy*, pages 411–428, 2015.

[23] Benjamin Tams. Decodability attack against the fuzzy commitment scheme with public feature transforms, 2014.

[24] Xiao Wang, Alex J. Malozemoff, and Jonathan Katz. EMP-toolkit: Efficient MultiParty computation toolkit. https://github.com/emp-toolkit, 2016.

[25] Samee Zahur and David Evans. Obliv-c: A language for extensible data-oblivious computation. Cryptology ePrint Archive, Paper 2015/1153, 2015. https://eprint.iacr.org/2015/1153.