Biometrics Cloak: A Security Analysis

Liv d'Aliberti

Computer Science, Princeton University

December 20, 2024

Abstract

Use of biometrics for local device authentication is an increasingly popular option given user convenience, biometric uniqueness, and protocol reliability. However, concern about preserving user privacy has limited use of biometric data for remote client authentication, the process of granting access to a non-local device via sharing a biometric. This note considers a proposed privacy-preserving approach to client-server biometric authentication. The primary objective of this protocol is to maintain protection of biometric data while allowing for secure authentication on a remote server.

1 Introduction

This note provides a security analysis of a privacy-preserving, biometric-based remote client authentication protocol that leverages Fully Homomorphic Encryption (FHE) and random masking. Biometric authentication is increasingly used for secure access due to reliability and convenience [1]. However, the use of biometric data raises critical concerns about privacy and data security, particularly when the biometric is processed remotely. To address these challenges, the proposed protocol, which uses a combination of random masking and FHE, was developed for scenarios where the biometric sample is gathered on a local client device but matched against a biometric template held on a remote server. The primary objective of the protocol is to protect all biometric information during the authentication process while guaranteeing accurate client authentication.

I analyze the protocol's server security against a malicious client. I consider scenarios where an adversary compromises a client device and attempts to exploit the protocol to gain unauthorized access. Specifically, I will focus on preventing the adversary from learning information about the biometric template stored on the server. Additionally, I formally establish the protocol as a secure authentication protocol, both sound and complete.

Future work would need to be done to analyze client privacy against a semi-malicious server: examining the scenario where the server behaves correctly but attempts to infer information about a client's biometric sample. The goal of this work would be to ensure that an honest client's biometric data remains protected throughout the authentication process.

1.1 Biometric Authentication

Use of biometric data for authentication has long been discussed in the literature and is now starting to see more widespread practical application for local authentication [2]. Examples include iPhone authentication via face recognition, computer access through fingerprint readers, iris scans for airport authentication, and voice recognition to access banking information. Additional examples of biometric modalities and use cases can be found in [1, 3].

A biometric is a measurable biological or behavioral characteristic used for automated recognition [4, 5]. While biometrics offer significant advantages for authentication, they also come with inherent risks. On the one hand, a biometric is relatively fixed [6] and uniquely associated [5] with the individual who provides it, making it an effective, convenient, and reliable identifier. On the other hand, if a biometric is compromised and made available to an adversary, the adversary gains permanent access to systems where authentication is tied to that biometric [1]. Unlike passwords, biometrics are non-revocable, meaning that a given biometric cannot be changed or reset.

Given the risks, it is crucial to provide robust protections for biometric data, especially when it is used for authentication processes that occur outside of a local client device, such as on a remote server. Ensuring the privacy of biometric data is essential to maintaining user trust and preventing unauthorized access.

1.2 Remote Client Authentication

Remote client authentication involves validating a client's identity through a security channel before granting access to a remote server [7]. This process ensures that only legitimate clients interact with the server, preventing unauthorized or malicious entities from gaining access. The proposed protocol uses biometric data to validate the remote client and protects the biometric data by leveraging FHE-based security, enabling privacy-preserving authentication. Specifically, the client proves its identity to the server without revealing the biometric sample in plaintext. The server, in turn, ensures that the client cannot manipulate the authentication protocol to gain unauthorized access. The protocol achieves this by performing secure biometric matching under encryption, ensuring the integrity of the authentication process while preserving the confidentiality of the biometric data.

1.3 Fully Homomorphic Encryption

Fully Homomorphic Encryption (FHE) allows for direct computation on encrypted data. While FHE was first theorized in the 1980s, the first concrete scheme was proposed by Craig Gentry's 2009 Stanford PhD thesis [8]. This note focuses on a specific FHE scheme known as CKKS (proposed by Cheon, Kim, Kim, and Song) [9, 10], which was designed for approximate arithmetic. CKKS is particularly well-suited for biometric authentication because it allows operations over real-valued data, which aligns with the nature of biometric data. While integer-based schemes could be adapted for biometric matching, CKKS simplifies the process by enabling computation to be performed natively on real numbers. This makes the protocol more efficient and practical for real-world biometric systems.

Previous works have explored biometric matching using FHE, including [11, 12], and this implementation is most similar to [13]. This note distinguishes itself by both the protocol's random masking methodology and comprehensive security analysis, which considers both client privacy against a semi-malicious adversary and server security against a malicious adversary.

1.4 Note Structure

The rest of the note is structured as follows: Section 2 provides formal definitions and security notation used throughout the analysis, Section 3 describes the privacy-preserving biometric authentication protocol. Section 4 carries out analysis, focusing on server security against a malicious client. Section 5 concludes the note by discussing future work and potential protocol improvements.

2 Preliminaries

I introduce the formal definitions and security notations used within this paper.

Definition 1 (CKKS Homomorphic Encryption Scheme). *A leveled Fully Homomorphic Encryption scheme ε, such as CKKS, that supports approximate arithmetic (addition and multiplication) operations on encrypted real or complex numbers with a controllable precision error. The scheme consists of the following polynomial-time algorithms (KeyGen, Enc, Dec, Eval) with the following properties:*

- $(sk, pk, evk, rlk) \leftarrow KeyGen(1^\lambda, N)$*, where* $\lambda \in \mathbb{N}$ *is the security parameter, and* $N \in \mathbb{N}$ *is the polynomial modulus degree.*
- $\hat{m} \leftarrow Enc(pk, m, \Delta)$*, where* Δ *is a scaling factor to encode floating-point numbers into integers and* $m \in \mathbb{R}$ *is a real plaintext message.*
- $m' \leftarrow Dec(sk, \hat{m}, \Delta)$*, where* $m' \in \mathbb{R}$ *is an approximate plaintext close to original plaintext* m*.*

- $c \leftarrow Eval(\hat{m_1}, \hat{m_2})$ *where the evaluation operation is either an additive or multiplicative operation over* m_1 *and* m_2*.*

Definition 2 (Negligible). *For every constant* C*,* $\epsilon(\lambda) < \lambda^{-C}$ *for all but finitely many* λ*.*

Definition 3 (Secure Authentication Protocol). *An authentication protocol is called a secure authentication protocol if it runs in polynomial time and completeness and soundness hold.*

Definition 4 (Completeness). *An authentication protocol is complete if for an honest client* c *and an honest server* S *given* t *as the biometric template,* b *as a fresh biometric sample from the same user such that distance*$(t, b) \leq \epsilon$*, the protocol succeeds with overwhelming probability (i.e.* S *accepts* C*'s claim) and outputs access token* $\tau = ACCEPT$*.*

Definition 5 (Soundness). *If a malicious client attempts to provide a biometric sample* b' *such that distance* $(t, b') > \epsilon$*, then for any probabilistic polynomial time (PPT) strategy* c^**, the server* S *should output* $\tau = ACCEPT$ *with probability negl*(n)*.*

3 Protocol

I describe the privacy-preserving, biometric-based authentication protocol.

3.1 Protocol Description

The protocol involves two types of parties, client devices $c \in C$, where C is the set of clients, and a centralized remote server S. Each c is tied to individual user $u \in U$, where U is the set of users. The user u would like to access information held on S by authenticating via their biometric data $b \in B$, where B is the set of valid biometric samples.

User u provides a biometric sample b, which the server S evaluates for authentication. Upon successful authentication, the server generates time-limited token τ for the client c to access S for a duration p. After p expires, re-authentication is required. The number of authentication attempts is limited to $x \in \mathbb{N}$; after x failed attempts, any communication from c will be ignored. All communications between c and S is assumed to occur over a secure channel, such as Secure Sockets Layer / Transport Layer Security (SSL/TLS).

Registration
First, c generates (sk_c, pk_c) using FHE scheme ε, and obtains a biometric template t for the associated user u. The biometric template is normalized such that each component

$$t_i = \frac{t_i}{\sum_{i=1}^{n} t_i}, \forall i \in \{1, \cdots, n\}. \tag{1}$$

Then, c encrypts t as $\hat{t} \leftarrow \text{Enc}(pk_c, t)$ and sends tuple $\{pk_c, \hat{t}, id_c\}$, where id_c is a unique identifier tied to device c, to S via trusted channel. S stores $\{pk_c, \hat{t}, id_c\}$ to be used for authentication.

Initialization
Now, u provides a fresh biometric sample b of the same size as the template, such that $|b| = |t|$. The sample is normalized such that,

$$b_i = \frac{b_i}{\sum_{i=1}^{n} b_i}, \forall i \in \{1, \cdots, n\}. \tag{2}$$

and it is encrypted, $\hat{b} \leftarrow \text{Enc}(pk_c, b)$. The client c then sends the tuple $\{\hat{b}, id_c\}$ to S via trusted channel.

Matching Computation
Upon receipt of $\{\hat{b}, id_c\}$, S then retrieves the corresponding encrypted template $\hat{t}$ for id_c. The server computes the encrypted squared Euclidean distance between $\hat{b}$ and $\hat{t}$:

$$\hat{d^2} = \sum_{i=1}^{n} (\hat{t_i} - \hat{b_i})^2. \tag{3}$$

Here, the computation occurs under encryption using homomorphic properties of ε. At this stage, the server has the encrypted result $\hat{d^2}$, but does not know its plaintext value. If the value of d^2 was naively sent to c, then c could simply manipulate that unknown d^2 and instead return a small d^2 to indicate low variation from the template. The server cannot guarantee that the value of d^2 is the decrypted version of the d^2 computed via biometric template matching.

Random Masking
To prevent the client from learning the exact value of d^2, the server applies its own secret, a random mask. The server selects μ uniformly at random from a large, discrete set $R = \{0, 1, \cdots, 2^{p(\lambda)} - 1\}$, where $|R|$ grows at least exponentially in λ, i.e. for S selected security parameter λ, $|R| \approx 2^{p(\lambda)}$ for some polynomial $p(\cdot)$. Then, select δ small, but large enough to satisfy $\delta \gg \Delta_{CKKS}/2\epsilon$, where ϵ is the threshold for match acceptance. Then, chooses $q = \mu + \delta$, $r = \mu - \delta$, such that $|q - r| = 2\delta$. The size of δ is important to ensure that $2\delta = |q - r|$ is sufficiently distinguished, even under approximate arithmetic from Δ_{CKKS}. The server encrypts such that $\hat{r} \leftarrow \text{Enc}(pk_c, r), \hat{q} \leftarrow \text{Enc}(pk_c, q)$ and computes,

$$\hat{y} = \hat{q} \cdot \hat{d^2} + \hat{r} \cdot (1 - \hat{d^2}). \tag{4}$$

Intuitively, if d^2 is small (close to zero), $\hat{y}$ should decrypt close to r, and if d^2 is large (exceeds ϵ), then the term involving q dominates, shifting the decrypted result away from r. The server returns $\hat{y}$ to the client via trusted channel.

Decryption and Response
Client c receives back $\hat{y}$ and decrypts, $y \leftarrow \text{Dec}(sk_c, \hat{y})$. Without knowledge of q, r, the decrypted value y appears random to the client - I analyze the randomness within Section 4. The client sends y back for verification over the trusted channel.

Verification
Upon receiving decrypted y, the server verifies the result by checking if $|y - r| \leq \epsilon'$, where ϵ' is the tolerance threshold chosen based on ϵ, q, r, such that $\epsilon' \geq |q - r|\epsilon$ and the approximate precision of the CKKS scheme. By setting ϵ' to be at least this value, I ensure that the server can distinguish a close match from a non-match. Then, server sends token τ, where $\tau \leftarrow$ ACCEPT if $y \approx_\epsilon r$ and $\tau \leftarrow$ REJECT otherwise. S sends τ to c which c can now access S for duration p.

4 Analysis

I first show that our authentication protocol is secure, i.e. it is complete and sound. I then show security under a chosen plaintext attack malicious client.

Theorem 1 (Completeness). The protocol described in Section 3 is complete in accordance with Definition 4.
Proof: Suppose that client c and server S follow the protocol honestly. Assume the client's sample b and stored template t are validly constructed and encrypted. Then, if $d^2 = \sum_{i=1}^{n}(t_i - b_i)^2$ is sufficiently small (i.e. a good biometric match), the protocol ensures that the server will accept.

Homomorphic Correctness of Distance Computation: By the properties of the FHE scheme ε, any polynomial operation on encrypted values corresponds to the operation on their plaintext. Thus,

$$\hat{d^2} = \sum_{i=1}^{n}(\hat{t}_i - \hat{b}_i)^2 \implies \text{Dec}(sk_c, \hat{d^2}) = d^2 = \sum_{i=1}^{n}(t_i - b_i)^2. \tag{5}$$

Since the protocol uses CKKS, the decrypted value d^2 is approximate but can be made arbitrarily close to the true value by adjusting the parameters.

Masking Computation: The server computes $\hat{y} = \hat{q} \cdot \hat{d^2} + \hat{r} \cdot (1 - \hat{d^2})$. Decryption then reveals

$$y = q \cdot d^2 + r \cdot (1 - d^2) = r + |q - r| \cdot d^2 = r + 2\delta d^2. \tag{6}$$

I consider the following cases to show completeness:

- **Case 1: Perfect Match** (Idealized Scenario): suppose $d^2 = 0$, then

$$y = r + (q - r) \cdot 0 = r. \tag{7}$$

  The server checks $|y - r| \leq \epsilon'$. Since $y = r$, the server trivially accepts.

- **Case 2: Close match**: Suppose $d^2 \leq \epsilon$, where ϵ is small. Then,

$$y = r + (q - r)d^2 \tag{8}$$

  For $\epsilon' \geq |q - r|\epsilon$ and small d^2, I then have

$$|y - r| = |r + |q - r|d^2 - r| = 2\delta d^2 \leq |q - r|\epsilon. \tag{9}$$

  Given the appropriate $\epsilon' \geq |q - r| \cdot \epsilon$, as described in the verification protocol, the server confirms that a small d^2 gives y close to r, therefore accepting. This takes into consideration, through the choice of ϵ', both potential arithmetic errors inherent to CKKS and the distance threshold ϵ.

- **Case 3: Rejection for Non-Matching Samples**: if $d^2 > \epsilon$, then

$$|y - r| = |q - r|d^2 = 2\delta d^2 > |q - r|\epsilon. \tag{10}$$

  Since q and r have been appropriately chosen such that $(q - r)\epsilon$ is sufficiently large, the difference $|y - r|$ should be easily detectable as exceeding ϵ', and consequently, the server should reject the attempt.

By the correct homomorphic computation of d^2 and the use of random masking values q, r, and an appropriately chosen tolerance ϵ', the protocol should ensure that if the biometric sample is a good match, i.e. $distance(t, b) = d^2 \leq \epsilon$. Thus, the protocol will succeed with overwhelming probability.

From Theorem 1, the following lemma should hold about the provided protocol,

Lemma 1 Suppose client c^* interacts with the protocol using server S, sample b_{c^*}, and that S accepts c^*, by providing $\tau = \text{ACCEPT}$, then it follows that (1) the sample b_{c^*} matches the template t held by S or (2) client c^* knows a good approximation of r for the given authentication run.
Proof: (1) follows directly from Theorem 1. (2) follows from the protocol, where S accepts $c^* \iff |y - r| < \epsilon' \leq |q - r|\epsilon$.

Theorem 2 (Soundness). The protocol in Section 3 is sound in accordance with Definition 5.

Proof: Suppose a malicious client c^* attempts to provide a biometric sample b' such that $d'^2 > \epsilon$, and that the server S accepts with non-negligible probability. Then, by Lemma 1, the malicious client would need to know a good approximation for r for the given authentication attempt.

Per Theorem 1, it holds that $d^2 > \epsilon$ results in a rejection. To achieve acceptance, c^* would need to produce a decrypted y that appears to the server as if $|y - r| = |2\delta d^2| < \epsilon'$. This task would require c^* to determine r compared to q, chosen statefully based upon the selection of u uniformly at random. The client's knowledge is limited to $y = r + 2\delta d^2$. Since r is uniform over $R \subset \mathbb{N}$, for a fixed y, to achieve acceptance, c^* would need

$$r \in [y - \epsilon', y + \epsilon']. \tag{11}$$

Since r is chosen to be based upon μ chosen uniform in R, the probability of choosing r within interval 2ϵ is

$$P[r \in [y - \epsilon', y + \epsilon']] \leq \frac{2\epsilon'}{|R|}. \tag{12}$$

By construction $|R|$ was chosen such that $\frac{1}{R}$ is negl(λ). Additionally, since ϵ' scales with ϵ, small, then $2\epsilon'$ can be bounded by a polynomial factor, i.e.

$$\frac{2\epsilon'}{|R|} \leq \frac{poly(\lambda)}{2^{p(\lambda)}} = \text{negl}(\lambda). \tag{13}$$

No PPT adversary can achieve acceptance with non-negligble probability $d^2 > \epsilon$. Thus, the protocol S will output $\tau =$ ACCEPT with probability negl(λ) and the protocol is sound.

Theorem 3 (Polynomial-Time Complexity). The protocol in Section 3 runs in PPT with security parameter λ and input size n, where n is the dimension of the biometric vectors.
Proof: By definition of leveled FHE schemes, ε $(KeyGen, Enc, Dec, Eval)$ all run in polynomial-time dependent upon security parameter λ and size of biometric n. Random mask generation is also, by definition polynomial-time dependent upon uniform random selection from R, sized based upon security parameter λ. Finally, because keys and ciphertexts scale polynomially, communication time is also polynomial in λ and n. Thus, the protocol time complexity is a sum of polynomial computations dependent upon n and λ, and thus, PPT.

4.1 Security Model - Malicious Client

I now suppose that the protocol from Section 3 is carried out in the real world, where a malicious client now attempts to gain unauthorized access by learning information about the biometric template stored on the server. To show that a malicious client is unable to learn any information about the biometric template,

I must show that the protocol is indistinguishability under chosen-plaintext attacks (IND-CPA).

I do not consider the case where the malicious client does not have access to sk, and only has access to pk, as biometric privacy is then dependent solely upon the security of FHE, which has been shown to hold. I instead focus upon a malicious client c^* that attempts to determine information about $\hat{t}$ based upon use of input samples b^*. The goal of the adversary is to learn about $\hat{t}$, as knowledge about $\hat{t}$ would give away biometric information.

Definition 6 (Indistinguishability under Chosen-Plaintext Attack Security Game). *Suppose that A is a PPT adversary with access to client c^* registered with Server S under template $\hat{t}^*$ encrypted under ε, a leveled FHE scheme with security parameter λ. The adversary A has access to both the pk, sk (aka they've aquired device c^*). The indistinguishability under chosen-plaintext attacks (IND-CPA) security game is as follows:*

1. *An adversary A makes $0 \leq s < x \in \mathbb{N}$ many chosen-plaintext queries b^* (per protocol, which limits number of attempts to x). For each query, A submits biometric $\hat{b}^* \leftarrow Enc(pk, b^*)$. The challenger returns $\hat{y}$.*
2. *After making s queries, A submits two distinct plaintext biometric samples b_0 and b_1, where $b_0 \neq b_1$. The challenger then randomly selects $b^* \in \{b_0, b_1\}$ and encrypts $b^* \leftarrow Enc(pk, b^*)$. The challenger then sends y^* to A.*
3. *The adversary then guesses based upon the returned y^*, whether $b_{guess} \in \{b_0, b_1\}$. The adversary wins the game if $P[b_{guess} = b^*] \geq \frac{1}{2} + negl(\lambda)$.*

Definition 7 (IND-CPA Security). The protocol is IND-CPA secure, even with adversarial accsess to pk, sk if no PPT adversary A can win the game with probability significantly greater than $\frac{1}{2}$, i.e.

$$P[b_{guess} = b^*] \leq \frac{1}{2} + \text{negl}(\lambda) \tag{14}$$

Theorem 4 (IND-CPA Security). The protocol in Section 3 is IND-CPA Secure with respect to Definition 7, even when the adversary possesses (pk, sk) of the FHE scheme.
Proof: Consider an adversary and challenger playing the IND-CPA game described in definition 6.

Homomorphic Encryption is IND-CPA Secure: By assumption the underlying FHE scheme ε is IND-CPA secure. Thus, given pk, encryptions of b_0 and b_1 are computationally indistinguishable to any PPT adversary that does not possess additional information about the plaintexts. Thus, without leakage, A cannot distinguish $\hat{b}_0, \hat{b}_1$ based on ciphertexts and homomorphic operations and is FHE IND-CPA secure.

Random Mask Acts as a One-Time Pad: Each authentication uses a fresh, uniformly random masking μ drawn from R of exponential size λ, such that the computed value $y^* = r + 2\delta d_{b^*}^2$ is dependent upon uniformly random r independent of b_0, b_1, and all previous attempts, it acts as a one-time pad on the result $d^2 b^*$.

Reduction to FHE Security: Suppose, for contraction, that A can distinguish between b_0, b_1 with probability greater thant $\frac{1}{2} + \text{negl}(\lambda)$. I then construct a PPT algorithm B that uses A to break the FHE IND-CPA security of the underlying FHE scheme ε.

- Given challenge ciphertext $\hat{b}^*$, B simulates the protocol's server for A. Since B does not know whether b_0 or b_1 was chosen, it masks output exactly as protocol does, picking fresh μ and homomoprhically computing $\hat{y}$, before sending $\hat{y}$ to A.
- A determines whether b_0 or b_1 and returns the answer to B.

Since r as computed by B is uniformly random, then A must be leveraging something about the encrypted $\hat{y}$ correlated to the $d_{b^*}^2$ to determine whether b_0 or b_1 is used. This would break the FHE IND-CPA security of the underlying FHE scheme. This is a contradiction, because the assumption ε is secure. Hence, no such A can exist.

5 Conclusion

This note provides a security analysis of a privacy-preserving, biometric-based remote client authentication protocol. I consider server security and client privacy. Future work would include implementation experimentation and analysis of client privacy against a semi-malicious server.

Alternatively, future work could explore the feasibility and security implications of the server generating and using its own FHE keys to encrypt the computed d^2 value, potentially enabling an even more secure, yet perhaps less computationally efficient protocol.

References

[1] R. Alrawili, A. A. S. AlQahtani, and M. K. Khan, "Comprehensive survey: Biometric user authentication application, evaluation, and discussion," 2024.

[2] A. K. Jain, D. Deb, and J. J. Engelsma, "Biometrics: Trust, but verify," *CoRR*, vol. abs/2105.06625, 2021.

[3] A. Jain, L. Hong, and S. Pankanti, "Biometric identification," *Communications of the ACM*, vol. 43, no. 2, pp. 90–98, 2000.

[4] U. D. of Homeland Security, "Biometrics," *U.S. Department of Homeland Security*.

[5] S. Alwahaishi and J. Zdrálek, "Biometric authentication security: An overview," in *2020 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)*, pp. 87–91, 2020.

[6] A. K. Martin and E. A. Whitley, "Fixing identity? biometrics and the tensions of material practices," *Media, Culture & Society*, vol. 35, no. 1, pp. 52–60, 2013.

[7] T. Weigold, T. Kramp, and M. Baentsch, "Remote client authentication," *IEEE Security & Privacy*, vol. 6, no. 4, pp. 36–43, 2008.

[8] C. Gentry, "Fully homomorphic encryption using ideal lattices," pp. 169–178, 2009.

[9] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *Advances in Cryptology – ASIACRYPT 2017* (T. Takagi and T. Peyrin, eds.), (Cham), pp. 409–437, Springer International Publishing, 2017.

[10] R. Agrawal and A. Joshi, *The CKKS FHE Scheme*, pp. 19–48. Cham: Springer International Publishing, 2023.

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

[12] L. Sperling, N. Ratha, A. Ross, and V. N. Boddeti, "Heft: Homomorphically encrypted fusion of biometric templates," in *2022 IEEE International Joint Conference on Biometrics (IJCB)*, pp. 1–10, 2022.

[13] G. Pradel and C. Mitchell, "Privacy-preserving biometric matching using homomorphic encryption," in *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pp. 494–505, 2021.