Practical Privacy-Preserving Face Identification based on Function-Hiding Functional Encryption

Alberto Ibarrondo, Hervé Chabanne, Melek Önen

Practical Privacy-Preserving Face Identification based on Function-Hiding Functional Encryption

Alberto Ibarrondo[1,2], Hervé Chabanne[1,3], and Melek Önen[2]

[1] IDEMIA, France
[2] EURECOM, France {ibarrond, onen}@eurecom.fr
[3] Telecom Paris herve.chabanne@telecom-paristech.fr

Abstract. Leveraging on function-hiding Functional Encryption (FE) and inner-product-based matching, this work presents a practical privacy-preserving face identification system with two key novelties: switching functionalities of encryption and key generation algorithms of FE to optimize matching latency while maintaining its security guarantees, and identifying output leakage to later formalize two new attacks based on it with appropriate countermeasures. We validate our scheme in a realistic face matching scenario, attesting its applicability to pseudo real-time one-use face identification scenarios like passenger identification.

Keywords: Biometric Matching · Face Identification · Functional Encryption · Privacy-Preserving Technologies

1 Introduction

The field of Biometrics studies physical and behavioral human characteristics to digitally identify a person. The most commonly used biometric traits are face, iris and fingerprint [16]. Biometrics are used in modern identification systems such as personal (mobile and laptop) authentication, identification for public administration, or border control/passenger identification in the travel industry.

However, biometric data acquisition and processing raises privacy concerns. Since biometric traits cannot be modified or re-issued, its protection is deemed indispensable. Furthermore, data protection regulations enforce strict limitations over usage and storage of biometrics data. While standard cryptography allows secure storage and transmission, secure processing requires advanced cryptographic techniques such as Fully Homomorphic Encryption (FHE)[9], Multiparty Computation (MPC)[19,17] and Functional Encryption (FE) [4].

This work uses FE to protect the biometric matching step with local output decryption. While FE is costly for arbitrary function evaluations, the inner product computed for a matching can be efficiently implemented using FE.

We present a face identification solution built on FE-based private inner product matching, with the key novelties of optimizing matching latency by switching functionalities of encryption and key generation FE algorithms, and identifying two attacks based on inner product output leakage coupled with suitable countermeasures. The paper is outlined as follows. Section 2 describes the

Biometric Matching and FE preliminaries. Section 3 details the proposed solution, architecture and characteristics. Section 4 covers a security analysis. Section 5 comprises implementation and experiments. Section 6 addresses previous work and positions our contribution.

2 Preliminaries

Biometric systems are pattern recognition systems that establish the authenticity of a specific physiological or behavioral user's characteristic. To do so, they scan and compress biometric traits into succinct representations called biometric templates, and perform comparisons between templates.

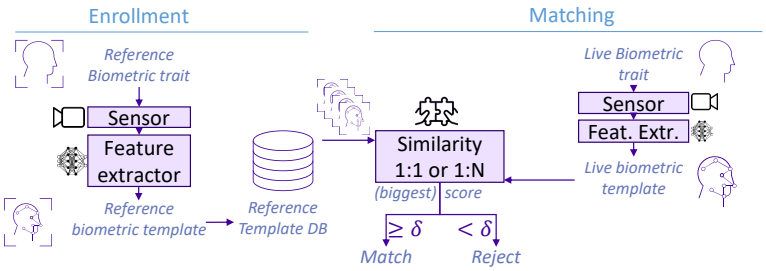

Fig. 1. Diagram of a standard biometric system

Biometric systems present two distinct phases, illustrated in Fig. 1. The *enrollment phase*, where reference templates are acquired and stored in a database, and the *matching phase*, when a live template is captured and matched with the reference templates yielding a positive result if the similarity score is higher than a fixed threshold δ. Depending on the number of reference templates, we can have two scenarios: *Verification* (a.k.a. Authentication) for 1:1 matching, and *Identification* for 1:N matchings. Receiving its input image from a capture sensor, the feature extractor component for face biometrics is nowadays based on Deep Learning models applied to Vision [7,8,2]. The resulting templates are normalized and matched using an inner product as similarity metric.

The face identification scenario we study on this work leads to two practical considerations. First, high numerical precision is paired with low error rates but privacy-preserving techniques support only integer operations. Secondly, N identities in the DB imply N similarity score computations for a matching, creating a natural bound to N so that an end-to-end identification be performed in pseudo real time, which we set to up to 5s. We exemplify the applicability of this work in a use-case of identification for transport boarding, requiring one-time-per-passenger identification of tens to low hundreds of individuals.

A **Functional Encryption** (FE)[4] scheme is a public-key encryption scheme where a "master" secret key msk is used to derive "functional" secret keys sk_k, allowing decryption for a certain function evaluation $F(k, x)$ on inputs x previously encrypted with public key pk without revealing anything else about

them. Only a handful of functions F efficiently are supported by FE schemes. The inner dot product $\mathbf{x} \cdot \mathbf{y}$ between two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_L^K$ is one of them [1].

We remark that there is no restriction to what the functional secret keys sk_k reveal about the function parameters k. For inner products where $F(k, \mathbf{x}) = \mathbf{x} \cdot \mathbf{y}_k$, sk_k reveal $\mathbf{y}_k$, one of the two biometric templates. We thus resort to *Function-Hiding* inner product encryption (FHIPE) schemes [12], which guarantees that sk_k hide the underlying vectors y_k. These are its four algorithms:

$pp, msk \leftarrow FE.setup(1^\lambda)$	generates public parameters pp and master secret key msk given security parameter λ
$sk_y \leftarrow FE.keygen(msk, \mathbf{y})$	generates functional secret keys sk_y for input y using master secret key msk
$c_x \leftarrow FE.encr(msk, \mathbf{x})$	encrypts message x with master secret key msk into ciphertext c
$z \leftarrow FE.decr(pp, sk_y, c_x)$	evaluates $z = \mathbf{x} \cdot \mathbf{y}$ from ciphertext c_x and functional secret key sk_y using pp

3 Our Solution

Security goals. We begin by establishing the security goals of our solution:

- **Privacy of all templates**. The enrollment phase should store *reference templates* in a privacy preserving manner still allowing inner products. Likewise, extracted *live templates* should support the inner product computation while remaining private for any other use. This is a standard by-design security goal of FE, already covered by the FHIPE scheme[12] of our solution.
- **Protection against inner product leakage**. FE schemes do not treat the inherent leakage of the reference template when computing several inner product operations with it. We formalize the newly identified output leakage below, and develop two practical leakage-based attacks in Sec. 4: *full reference template extraction* and *brute-force impersonation*. Usually overlooked in the secure computation literature, we stress the importance of this leakage in our face identification scenario, where multiple inner products are computed over the same reference template. To protect against them, we establish a carefully selected limit to the total number of identification requests in our solution.

Definition 1 (Inner Product Leakage). *For a single call to $\mathbf{x} \cdot \mathbf{y}$, an inner product of two vectors in $\mathbf{Z}_L^K$, we define function leakage ι as the inverse of the minimum number p of calls required to unequivocally determine an unknown input $\mathbf{y}$ from known inputs $\mathbf{x}_j$ and known outputs $z_j = \mathbf{x}_j \cdot \mathbf{y} \quad \forall j \in \mathbb{N}, j \leq p$. As an extension, for n inner product calls we define accumulated leakage to be $\breve{\iota} = n * \iota$. With $\breve{\iota} \geq 1$, the unknown input $\mathbf{y}$ is revealed.*

Threat model. We consider a semi-honest adversary corrupting the similarity score operation and all steps after that, seeking to obtain as much information as possible from the inputs but preserving their integrity. We consider the adversary to have oracle access to the matching phase, thus being able to submit chosen live biometric samples. Our system is built with trust on the enrollment and the capture modules, for they receive the msk which can decrypt any ciphertext.

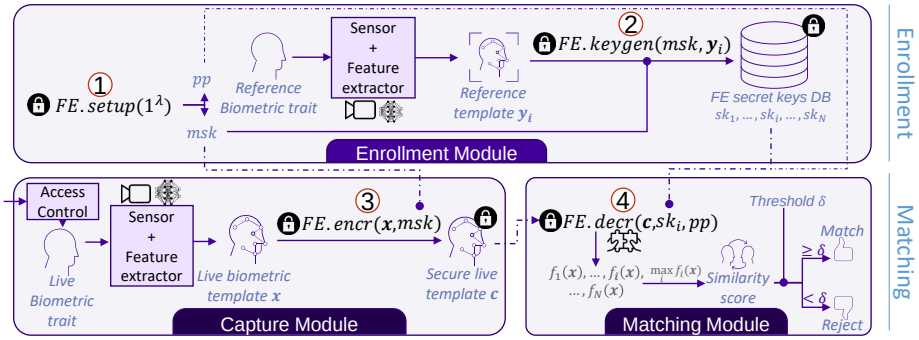

Fig. 2. Architecture of our secure face identification system based on FE

Swapping *FE.encr* with *FE.keygen*. The original FHIPE scheme (Sec. 5.1 of [12]) and posterior works based on it [11] use the function-hiding $FE.keygen$ functionality to protect the live template (step 3 in Fig. 2), keeping $FE.encr$ for the stored templates (step 2 in Fig. 2). We observe that, given the dual nature of the FHIPE scheme, the same security properties hold if we were to swap them. This observation is grounded on remark 3.4.5. of [5]: in the game-based IND-CPA security definition of FHIPE (Fig. 3.10 of [5] or definition 2.1) the adversary and the oracle follow a perfectly equivalent game. To optimize the matching latency we employ the fastest functionality for this phase, which happens to be $FE.encr$ (see Sec. 5), thereby swapping $FE.encr \rightleftharpoons FE.keygen$ with respect to [12,11].

Limiting the number of requests. For templates with K l-bit elements in $\mathbb{Z}_L$ ($L \approx 2^{(l-1)}$), we limit the number N of identification requests of our solution to $N < K$, in order to prevent full reference template extraction due to output leakage (keeping $\breve{\iota} < 1$, detailed in Sec. 4). We enforce this limit via an *access control step* with open instantiation, which could materialize as an agent-controlled checkpoint or a one-time token generated in the enrollment. Furthermore, we add a security margin of 80 bits to hinder brute-force impersonation attacks identified in Sec. 4, leading to a final limit of $N < (K - 80/l)$ requests.

System description. We display our solution in Fig. 2. In the *enrollment phase*, the enrollment module acts as trusted authority to generate msk & pp and protect N ref. templates by converting them into functional keys sk_i. msk is sent to the capture module, and all sk_i along with pp are sent to the matching module. The *matching phase* starts with the access control step. The capture module then gets a live template x and encrypts it into c using msk. Afterwards, the matching module takes sk_i and c, computes their privacy-preserving inner product $z_i = \mathbf{x} \cdot \mathbf{y_i}$, compares the highest score $max(z_i)$ to the threshold δ, and returns a match with the ID/index i of the highest score, or nothing if rejected.

The feature extractor outputs normalized templates $\mathbf{t} \in \mathbb{R}^K_{[-1,1]}$, easily projected into the FHIPE discrete space $\mathbb{Z}_{2^l}$ by scaling with factor 2^l and a truncation to l bits. The subsequent inner product is naturally up-scaled twice:

$$f(\mathbf{x}_{fix}, \mathbf{y}_{fix}) = \lfloor \mathbf{x}_{float} * 2^l \rfloor_l \cdot \lfloor \mathbf{y}_{float} * 2^l \rfloor_l \approx 2^{2l} * f(\mathbf{x}, \mathbf{y})$$

To compare against the threshold $\delta \in [0,1]$ we upscale δ twice: $\delta_{fix} = \delta * 2^{2l}$, obtaining an equivalent comparison. This fixed-point translation imposes a minimum ring size of $2l$ bits to avoid overflows. The approximation impacts the accuracy, since more bits yield more precision, but at the cost of bigger primitives in the FE scheme and thus worse latency. We study this trade-off in Sec. 5.

4 Security Analysis

This section first covers the template privacy with FHIPE security, then identifies two novel attacks based on output leakage (Def. 1), proposing countermeasures.

Theorem 1. *Our system preserves the privacy of the live template and the reference templates while allowing the inner product similarity computation.*

Proof. The security of FE sits upon game-based definitions that prove Indistinguishability against Chosen Plaintext Attacks a.k.a. IND-CPA (IND[1,4])). The FHIPE scheme of our solution is proven to hold strong SIM-based security guarantees as per theorem 3.1 of [12], which implies IND-CPA secure in Remark 2.5 of [12]. This directly ensures the privacy of the biometric templates inside ciphertexts and functional secret keys of our solution. □

Protection against output leakage. The inner product function, and by extension all IPE schemes, suffer from output leakage of $\iota = 1/K$, for input vectors of K elements. This leads to a **full reference template extraction attack**, where the attacker launches N chosen identification requests and uses the results to reconstruct a hidden reference template. Indeed the system of N linear equations $\{z_j = \sum_{i=1}^{K} \mathbf{x}_j^{[i]} * \mathbf{y}^{[i]}, j \in \{1,2,...,N\}\}$ for known z_j and $\mathbf{x}_j$ has a unique, non-trivial solution for the K unknown variables $\mathbf{y}^{[i]}$ when all the equations are linearly independent and $N \geq K$. In our system (Sec. 3) we propose the countermeasure of limiting the number N of calls to F to $N < K$, ensuring $\breve{\iota} < 1$. This way, the above system of equations is underdetermined; for $\mathbf{y}^{[i]} \in \mathbb{Z}_L = \{-L, \cdots, -1, 0, 1, \cdots L\}$ there are exactly $(2 * L + 1)^{(K-N)}$ solutions.

In addition, IPE-based biometric matching schemes with up to $N < K$ requests can be subject to a **brute-force impersonation attack**, where a partially extracted ($\breve{\iota} < 1$) reference template $\hat{\mathbf{y}}$ is used to impersonate its owner. The attacker first sets the remaining $K - N$ unknown values of $\hat{\mathbf{y}}$ to arbitrary values and launches several identification requests, so that the FHIPE result $z = \hat{\mathbf{y}} \cdot \mathbf{y}$ might yield $z \geq \delta$, thus matching positive for the identity of $\mathbf{y}$. Beyond this, the attacker could also resort to prior knowledge of the template space (obtainable from feature extractors with similar characteristics) and project the partially extracted template to it, further increasing the chances of a successful impersonation. To thwart these attacks, we set an additional security margin τ to the number N of calls to F in our solution (see Sec. 3), so that $N < (K - \tau)$. Seeking to increase the number of possible solutions of the above system of equations to 2^{80} (80 bits), we fix $\tau \approx 80/l$ for template values of l bits ($L \approx 2^{(l-1)}$).

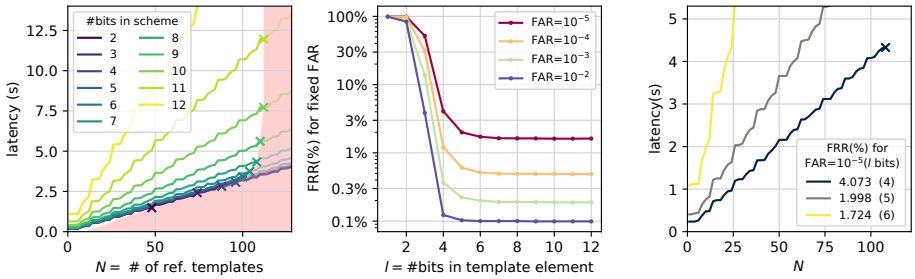

Fig. 3. Experimental results on Latency vs number N of identities (left), on precision vs template element size l (center), and practical trade-off between parameters (right).

5 Experiments

We implement our Cython-based solution using the CiFEr[15] library, an ArcFace based[7] feature extractor with templates of size $K = 128$. The experiments were run in an Intel(R) Core(TM) i7-7800X CPU and averaged over 10 runs.

Table 1. Latency (seconds) for single-core FE.decr with template elements of l bits.

l	2	4	6	8	10	12	14	16
FE.decrypt (s)	0.18	0.18	0.19	0.25	0.40	1.08	3.81	14.86

Latency optimization in the matching phase is essential to make our system practical. Using a single core, we measure $FE.setup$ (step 1) to take 0.35s, $FE.keygen$ (step 2) requires 0.19s per key, and $FE.encr$ (step 3) demands 0.082s; thus our proposed swapping reduces the latency of live template protection by 55%. As the only FE operation depending on the template element size, $FE.decr$ latency is recorded in table 1. The feature extractor clocks 36 ± 1ms . We disregard the latency of the access control step, as its instantiation is left open; and the max, the comparison with δ and the secure transmission for being negligible compared to the FE operations. Additionally, we analyze the matching module in the left Fig. 3 based on the number N of identities in our system for a one-time identification scenario. As per Sec. 4, N is limited to strictly less than the template vector length ($N < 128$) to avoid full leakage of the stored templates, and a red area marks the additional security margin to thwart brute-force attacks.

Precision is measured with face identification benchmarks using the Labeled Faces in the Wild (LFW) dataset[10] consisting of 13233 112x112px real face images of famous people. We employ the widespread False Acceptance Rate (FAR) and False Rejection Rate (FRR) as metrics[4]. Typically, robust identification systems enforce $FAR < 10^{-4}$, obtaining a corresponding higher FRR. In the central graph of Fig. 3, we remark that highly compressed templates maintain high precision, with little improvement beyond $l = 6$.

[4] More info in `https://en.wikipedia.org/wiki/Biometrics#Performance`

To close up, the right Fig. 3 presents the best trade-offs in two scenarios:

- **Higher precision**: Optimizing for low FRR, setting $l = 5$ bits per template element to support up to 70 identities, with slower matching of up to $5s$.
- **Many identities**, optimizing for high N (up to 100 identities) by setting $l = 4$ bits, at the cost of $+2\%$ FRR but with faster matching ($\approx 4s$).

6 Previous Work

The study of IPE started off with selective security in [1], already envisioning biometric use-cases, and reaching full security with [6,18]. The function-hiding properties for IPE were introduced in [12], applied to biometric authentication based on Hamming weight ($l = 1$). Further works in function-hiding approaches include [13] and [11]. [3] covers an overhaul of efficient techniques.

The use of FE for privacy-preserving biometrics has also been subject to intense scrutiny, from [20] for biometric authentication using threshold predicate encryption, to the extreme efficiency of [14]. Whereas these works employ Hamming-weight based matchings that do not require approximations (typical from fingerprint or iris), our work tackles the cosine-similarity based matching of face biometrics. [2] covers an exhaustive revision of face recognition, which includes the LFW dataset [10] and the foundations of our feature extractor [7].

Among the most recent works, [11] proposes a useful acceleration trick for the FE scheme of [12], by caching all the repetitive computation depending only of the stored templates, obtaining up to 30% speedups. Much like the original [12], their function-hiding approach uses $FE.encr$ for the stored templates and $FE.keygen$ for the live templates. Our function-hiding solution swaps $FE.encr \rightleftharpoons FE.keygen$ to optimize the latency of the system.

Finally, where all previous works focus on the privacy provided by FE, we identify and address the IPE output leakage, often overlooked and not covered by the security guarantees of FE schemes.

7 Conclusions

This work proposes an efficient, precise and privacy-preserving face identification system based on function-hiding functional encryption. We highlight the inherent leakage of inner product schemes and identify novel reference template extraction and brute force attacks. To counter them, we set a hard limit with a security margin to the number N of identities in the system, adding an access control step to enforce it. In addition, we optimize the matching phase latency by swapping $FE.encr$ and $FE.keygen$ usage, speeding up the live template protection by 55% while maintaining the FE security guarantees. Finally, we implemented this system, showing that 4/5 bits per template element are enough to obtain precise setups that compute matchings against a database of up to 100 identities in pseudo real-time, applicable to passenger identification use-cases.

Acknowledgements. The authors thank Vincent Despiegel for his valuable help towards giving birth to this work. Moreover, we express our gratitude to the willful guidance of Zekeriya Erkin. This work has also been partially supported by the 3IA Côte d'Azur program (reference number ANR19-P3IA-0002).

References

1. Abdalla, M., Bourse, F., De Caro, A., Pointcheval, D.: Simple functional encryption schemes for inner products. In: IACR International Workshop on PKC. pp. 733–751. Springer (2015)
2. Adjabi, I., Ouahabi, A., Benzaoui, A., Taleb-Ahmed, A.: Past, present, and future of face recognition: A review. Electronics **9**(8), 1188 (2020)
3. Barbosa, M., Catalano, D., Soleimanian, A., Warinschi, B.: Efficient function-hiding functional encryption: From inner-products to orthogonality. In: Cryptographers' Track at the RSA Conference. pp. 127–148. Springer (2019)
4. Boneh, D., Sahai, A., Waters, B.: Functional encryption: Definitions and challenges. In: Theory of Cryptography Conference. pp. 253–273. Springer (2011)
5. Bourse, F.: Functional encryption for inner-product evaluations. Ph.D. thesis, PSL Research University (2017)
6. Datta, P., Dutta, R., Mukhopadhyay, S.: Functional encryption for inner product with full function privacy. In: PKC 2016, pp. 164–195. Springer (2016)
7. Deng, J., Guo, J., Niannan, X., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: CVPR (2019)
8. Deng, J., Guo, J., Yuxiang, Z., Yu, J., Kotsia, I., Zafeiriou, S.: Retinaface: Single-stage dense face localisation in the wild. In: arxiv (2019)
9. Gentry, C., et al.: A fully homomorphic encryption scheme, vol. 20. Stanford (2009)
10. Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Tech. Rep. 07-49, University of Massachusetts, Amherst (October 2007)
11. Jeon, S.Y., Lee, M.K.: Acceleration of inner-pairing product operation for secure biometric verification. Sensors **21**(8), 2859 (2021)
12. Kim, S., Lewi, K., Mandal, A., Montgomery, H., Roy, A., Wu, D.J.: Function-hiding inner product encryption is practical. In: SCN18. pp. 544–562. Springer (2018)
13. Kim, S., Kim, J., Seo, J.H.: A new approach to practical function-private inner product encryption. Theoretical Computer Science **783**, 22–40 (2019)
14. Lee, J., Kim, D., Kim, D., Song, Y., Shin, J., Cheon, J.H.: Instant privacy-preserving biometric authentication for hamming distance. IACR Cryptol. ePrint Arch. **2018**, 1214 (2018)
15. project, F.: Cifer: Functional encryption library. `https://github.com/fentec-project/CiFEr` (2021)
16. Sabhanayagam, T., Venkatesan, V.P., Senthamaraikannan, K.: A comprehensive survey on various biometric systems. International Journal of Applied Engineering Research **13**(5), 2276–2297 (2018)
17. Shamir, A.: How to share a secret. Comm. of the ACM **22**(11), 612–613 (1979)
18. Tomida, J., Abe, M., Okamoto, T.: Efficient functional encryption for inner-product values with full-hiding security. In: ICIS. pp. 408–425. Springer (2016)
19. Yao, A.C.C.: How to generate and exchange secrets. In: 27th Annual Symposium on Foundations of Computer Science (sfcs 1986). pp. 162–167. IEEE (1986)
20. Zhou, K., Ren, J.: Passbio: Privacy-preserving user-centric biometric authentication. IEEE Transactions on Info. Forensics and Security **13**(12), 3050–3063 (2018)