Security of a Privacy-Preserving Biometric Authentication Protocol Revisited

Aysajan Abidin[1], Kanta Matsuura[2], Aikaterini Mitrokotsa[1]

[1]Chalmers University of Technology, Gothenburg, Sweden
{aysajan.abidin, aikaterini.mitrokotsa}@chalmers.se
[2]University of Tokyo, Japan
kanta@iis.u-tokyo.ac.jp

Abstract. Biometric authentication establishes the identity of an individual based on biometric templates (*e.g.* fingerprints, retina scans etc.). Although biometric authentication has important advantages and many applications, it also raises serious security and privacy concerns. Here, we investigate a biometric authentication protocol that has been proposed by Bringer *et al.* and adopts a distributed architecture (*i.e.* multiple entities are involved in the authentication process). This protocol was proven to be secure and *privacy-preserving* in the *honest-but-curious* (or *passive*) attack model. We present an attack algorithm that can be employed to mount a number of attacks on the protocol under investigation. We then propose an improved version of the Bringer *et al.* protocol that is secure in the *malicious* (or *active*) insider attack model and has *forward security.*

Key words: Biometrics, privacy-preserving biometric authentication, homomorphic encryption, active attack, forward security.

1 Introduction

Biometric authentication offers important advantages mainly due to the uniqueness of biometric identifiers and other favorable properties since biometrics cannot be lost or forgotten. A biometric authentication system consists of two phases, namely, the *enrollment phase* and the *authentication phase*; and it typically involves two entities: a client and a server. During the enrollment phase, the client provides the server with his biometric data for storage in a database. Then, during the authentication phase, the server authenticates the client if his fresh biometric template matches the one that is stored in the database.

Since the server often has to perform many tasks (*e.g.* retrieving from the database the client's reference biometric template, checking if it matches the fresh template) its role can be divided into several parts. Thus, the execution of the protocol involves different entities where each entity performs a specific task. For instance, a biometric authentication protocol could involve the following entities: a user $\mathcal{U}$, a biometric sensor $\mathcal{S}$, an authentication server $\mathcal{AS}$, a database $\mathcal{DB}$ and a matcher $\mathcal{M}$. This architecture of a biometric authentication system has been proposed by Bringer *et al.* [1]. In this new setup, a biometric authentication system works as follows. Let N be the number of users registered in the authentication system. We denote by $\mathcal{U}_i$ the i-th user where $1 \leq i \leq N$. In the *enrolment phase* the user $\mathcal{U}_i$ registers his biometric data b_i which is then stored in the database $\mathcal{DB}$. In the *authentication phase* a user $\mathcal{U}_i$ first provides a fresh biometric trait b'_i and his identity ID_i to the sensor $\mathcal{S}$, which in turn forwards these data to the authentication server $\mathcal{AS}$. $\mathcal{AS}$ then asks $\mathcal{DB}$ for $\mathcal{U}_i$'s biometric data b_i that is already stored in $\mathcal{DB}$. After getting b_i from $\mathcal{DB}$, $\mathcal{AS}$ sends b_i and b'_i to the matcher $\mathcal{M}$, which checks whether b_i and b'_i match and sends back the result of the comparison to $\mathcal{AS}$, which then makes the decision of whether to grant authentication to the user depending on the matcher's response.

Note that it is assumed that the output of the authentication process denoted as $\mathsf{Out}_{\mathcal{AS}}$ (*i.e.* knowing whether the authentication has been granted or not) is publicly available; something that is quite common in the literature [2,3,4,5,6]. For instance, in case the biometric authentication system is used to restrict access to a building then the event that the door opens corresponds to a successful authentication.

However, biometric authentication has also many serious security and privacy implications. Compromised biometric templates may lead to serious threats to identity, while the inherent irrevocability of biometrics renders this risk even more serious. Furthermore, biometric information may reveal very sensitive and private information such as genetic [7] and medical information [8]. Additional issues of linkability, profiling and tracking of individuals are raised by cross-matching biometric traits. Therefore, privacy-preserving biometric authentication protocols are of utmost importance. Many existing protocols rely on the use of secure multi-party computation techniques including homomorphic encryption [9] and oblivious transfer [10,11].

Contributions and Related Work In this paper, we review a privacy-preserving biometric authentication protocol that has been proposed by Bringer *et al.* [1]. This protocol relies on the Goldwasser-Micali (GM) cryptosystem [12] which is a homomorphic encryption. Bringer *et al.* [1] have shown that their protocol is secure under the assumption that the system entities do not collude and are *honest-but-curious.* Here, we improve upon the original protocol to safeguard it against *malicious* insider attacks.

We first present a generic algorithm that can be employed by an adversary to mount a number of attacks to the protocol under investigation. One of the enablers of the attacks is the bit-by-bit encryption of the biometric data using the GM encryption scheme. Then, we propose an improved protocol that is secure and privacy-preserving in the *malicious* adversarial model. In particular, the improved protocol is secure against *malicious, but non-colluding* insider attacks and has *forward security.* We also compare our protocol with the original protocol.

Some attacks on the protocol under study were presented by Barbosa *et al.* [13] and Simoens *et al.* [14]. Barbosa *et al.* [13] present a simple attack that allows the authentication server $\mathcal{AS}$ to learn some bits of the reference biometric templates due to non-randomisation of the response by the database $\mathcal{DB}$ to the authentication server $\mathcal{AS}$. Simoens *et al.* [14] present possible insider attack ideas and attacks by a single or multiple, colluding malicious entities. In this paper, we extend some of their attack ideas and present a simple yet powerful attack algorithm.

Bringer and Chabanne [15] presented an improvement of the protocol under study, where they replaced the matching algorithm by an error correction procedure using secure sketches and discussed how it can be integrated into the *Private Information Retrieval (PIR)* scheme due to Lipmaa [16]. In their scheme, the database stores encryptions of the biometric templates. However, this scheme is computationally expensive. There are also several biometric authentication protocols proposed by Stoianov [17] that employ the Blum-Goldwasser (BG) [18,19] encryption scheme. But in these protocols, there are three entities, namely a client, a computation server (or database), and an authentication server. There are many other works related to privacy-preserving biometrics. However, to the best of our knowledge, Barbosa *et al.* [13] and Simoens *et al.* [14] are the only ones that study the security of the protocol under investigation.

Outline After giving some definitions and our threat model in Section 2, we present the protocol under study in Section 3. Then, in Section 4, we describe the attack algorithm. Section 5 presents an improvement of the Bringer *et al.* protocol while Section 6 presents its security analysis and compares it with the original protocol. Finally, Section 7 concludes the paper and highlights some future work.

2 Preliminaries

We give notations and definitions of some of the key concepts used throughout the paper. Also, we present a threat model in which we analyse the security and privacy of the biometric authentication protocol under study.

Communication Model In our modifications to the protocol under investigation, we assume that there is a secure and authentic channel between the system entities. In particular, we assume that there are shared secret keys between $\mathcal{S}$ and $\mathcal{M}$, $\mathcal{AS}$ and $\mathcal{M}$, $\mathcal{DB}$ and $\mathcal{M}$, that are used to encrypt and authenticate messages sent to $\mathcal{M}$. In addition, $\mathcal{M}$ has a public encryption key to which all other system entities have access, and $\mathcal{S}$ and $\mathcal{DB}$ have a shared secret key that they use to derive a permutation to permute the biometric templates before encrypting them. Since we omit the underlying infrastructure for the public-key primitive (*i.e.* the protocol does not explicitly use certificates), we also assume the authenticity of the matcher's public key. In this paper, we focus on the case where there is only a single $\mathcal{S}$, a single $\mathcal{AS}$, and a single $\mathcal{DB}$ in the system. Therefore, security in the case of multiple entities communicating with each other in parallel is outside the scope of this paper.

Definitions We use the following as a definition of privacy-preserving biometric authentication.

Definition 1 (Privacy-preserving biometric authentication). *We say a biometric authentication protocol is* privacy-preserving *if no probabilistic polynomial-time (PPT) adversary can recover any of the following information, if they are not already known: a fresh biometric b'_i, a stored biometric b_i or the correspondence between the identity ID_i and b_i.*

We also use provably secure message authentication codes (MACs) in our modification to the protocol under study. A MAC scheme MAC consists of a key generation algorithm KeyGen, a tag generation algorithm TAG, and a verification algorithm VRFY. When we say a MAC scheme is ϵ-secure, we refer to the following definition.

Definition 2. *A MAC scheme is called ϵ-secure if no PPT adversary $\mathcal{A}$ can generate a valid message-tag pair, even after making polynomially many tag generation and verification queries, except with probability ϵ.*

Furthermore, when we say secure pseudorandom number generator (PNG) we mean a PNG that satisfies the following definition.

Definition 3. *A PNG is called an ϵ-secure if no PPT distinguisher D can distinguish its output from a randomly chosen bitstring of equal length except for a negligible probability ϵ.*

Lastly, we use symmetric key encryption, denoted by SKE, in our modification. We require SKE to have indistinguishability against ciphertext-only attacks ($\mathsf{IND\text{-}COA}$) (cf. Appendix A). Note that we use Enc (and Dec) to denote the GM encryption (and decryption), and Enc_K (and Dec_K) to denote symmetric key encryption (and decryption) with a key K.

Threat Model In our threat model, we go beyond the *honest-but-curious* (or *passive*) model that is adopted in the original protocol by Bringer *et al.* [1] and extend the adversary model investigated by Simoens *et al.* [14]. Hence, we consider as an adversary $\mathcal{A}$ any passive (or active) internal entity that can violate the protocol specifications and that attempts to recover any of the following information, if they are *not* yet known: the fresh biometric b'_i, the stored template b_i, and/or the correspondence of a user identity ID_i to the stored template b_i. Thus, each of the entities – the user ID_i, the sensor $\mathcal{S}$, the authentication server $\mathcal{AS}$, the database $\mathcal{DB}$, and the matcher $\mathcal{M}$ – may pose threats to privacy of biometric reference, biometric sample and user identity [14].

Assumptions When security and privacy of a biometric authentication system are analysed, there are always certain assumptions that must hold. In our case, we make the following assumptions.

Assumption 1 *We assume that the sensor $\mathcal{S}$ is honest, has not been compromised and captures the biometric templates from alive human users.*

This assumption is important because if the sensor $\mathcal{S}$ is compromised, then the adversary can wait until a legitimate user comes and authenticates himself to the system, and hence easily learns the identity and fresh biometric template of a legitimate user. This is possible because as we mentioned earlier the output of the authentication server to the user is assumed to be publicly known.

A malicious user may attempt to get himself authenticated to the system by a fake identity and a fake biometric template. Also, a series of successful consecutive authentication attempts by the same user identity may also be an indication of a malicious behaviour if there is a specific pattern in the biometric templates used. Therefore, we assume that the system has appropriate measures to limit the number of such trials. This brings us to our next assumption.

Assumption 2 *We assume that the biometric authentication system has a limit on the maximum allowed consecutive failed trials to grant access. This limit does not allow an adversary to create a fake fresh biometric b'_i that is accepted by the matcher $\mathcal{M}$. Also, we assume that the system has a limit on the maximum allowed consecutive successful trials to grant access. This limit helps the system to detect hill climbing attacks; see Simoens et al. [14] for details on this attack.*

Finally, we assume that the system entities are not colluding. We note that this assumption is valid when an adversary has compromised *only* one of the entities. And we believe that this is an important first step towards achieving a protocol secure against *malicious and colluding* insider attacks.

Assumption 3 *We assume that the entities $\mathcal{AS}$, $\mathcal{DB}$, $\mathcal{M}$ may not collude with each other.*

3 The Bringer *et al.* Protocol

Bringer *et al.* [1] have proposed a protocol for privacy-preserving biometric authentication that follows the above described model and involves four entities in the biometric authentication process. According to this protocol the sensor $\mathcal{S}$, the authentication server $\mathcal{AS}$ and the database $\mathcal{DB}$ store the public key pk while the matcher $\mathcal{M}$ stores the secret key sk. $\mathcal{AS}$ also stores the mapping (ID_i, i), for $i = 1, \ldots, N$, where i corresponds to user $\mathcal{U}_i$ and N is the total number of users of the biometric authentication system. Furthermore, $\mathcal{DB}$ stores the reference biometric template b_i. The protocol is based on the GM cryptosystem. We denote by $\mathsf{Enc}(b_i)$ the *bit-by-bit* encryption of the template b_i, i.e. $\mathsf{Enc}(b_{i,1} \ldots b_{i,M}) = \big(\mathsf{Enc}(b_{i,1}), \ldots, \mathsf{Enc}(b_{i,M})\big)$, where M is the bit length of the template.

In the enrolment phase, user $\mathcal{U}_i$ registers (b_i, i) at $\mathcal{DB}$, and (ID_i, i) at the $\mathcal{AS}$. The authentication phase comprises the following phases.

Phase 1 - Communication $\mathcal{U}_i \rightarrow \mathcal{S} \rightarrow \mathcal{AS}$:

- $\mathcal{U}_i$ provides a fresh biometric trait b'_i and his identity ID_i to $\mathcal{S}$.
- Then, $\mathcal{S}$ sends the fresh biometric b'_i encrypted under the public key pk (*i.e.* $\mathsf{Enc}(b'_i)$) as well as the claimed identity ID_i to $\mathcal{AS}$.

Phase 2 - Communication $\mathcal{AS} \leftrightarrow \mathcal{DB}$:

- $\mathcal{AS}$ performs the mapping from ID_i to i and then using a PIR mechanism sends i and requests the corresponding stored biometric template b_i. More precisely, $\mathcal{AS}$ sends to $\mathcal{DB}$ the encrypted value $\mathsf{Enc}(t_j)$, where $1 \leq j \leq N$ and $t_j = 1$, if $j = i$, 0 otherwise.
- $\mathcal{DB}$ computes: $\mathsf{Enc}(b_{i,k}) = \prod_{j=1}^{N} \mathsf{Enc}(t_j)^{b_{j,k}}$ where $1 \leq k \leq M$ and then sends the computed values $\mathsf{Enc}(b_{i,k})$ to $\mathcal{AS}$.

Phase 3 - Communication $\mathcal{AS} \leftrightarrow \mathcal{M}$:

- $\mathcal{AS}$ computes $v_k = \mathsf{Enc}(b'_{i,k})\mathsf{Enc}(b_{i,k}) = \mathsf{Enc}(b'_{i,k} \oplus b_{i,k})$, where $1 \leq k \leq M$. Then, $\mathcal{AS}$ permutes v_k and sends $\lambda_k = v_{\pi(k)}$ $(1 \leq k \leq M\ )$ to $\mathcal{M}$.
- $\mathcal{M}$ decrypts the permuted vector λ_k and checks whether the Hamming weight (HW) of the decrypted vector is less than a predefined threshold τ. The result of this control is sent to $\mathcal{AS}$.

Phase 4 - Communication $\mathcal{AS} \rightarrow \mathcal{U}_i$: Finally, $\mathcal{AS}$ accepts or rejects the authentication request ($\mathsf{Out}_{\mathcal{AS}} = 1$ or $\mathsf{Out}_{\mathcal{AS}} = 0$ respectively) depending on the value returned by $\mathcal{M}$.

4 Description of the Attacks

Barbosa *et al.* [13] and Simoens *et al.* [14] presented several attacks on the above protocol when the adversary is a single entity or a combination of multiple entities. In addition, Simoens *et al.* [14] presented a framework for analysing security and privacy of biometric data in biometric authentication systems. In this section, we present a simple yet powerful algorithm (*Algorithm 1*) that can be used as a basis for a number of attacks. The attack algorithm takes a ciphertext as input and returns the corresponding plaintext by querying the matcher. The main enabler of this attack algorithm is the *bit-by-bit encryption* of the communication between the involved parties and the use of Hamming distance as the measure of whether the fresh biometric template matches the stored biometric profile. The algorithm uses as a subroutine the algorithm for the *center search attack*, but it is called only if the condition $\mathsf{HW}(b_i) \leq \tau$ holds; we urge the interested reader to consult Simoens *et al.* [14] for details on the attack

The Attack Idea Upon receiving from $\mathcal{AS}$ a vector λ of ciphertexts, the matcher $\mathcal{M}$ first decrypts λ component-by-component and then compares the Hamming weight of the resulting bitstring with a predefined threshold τ. $\mathcal{M}$ responds YES to $\mathcal{AS}$ if the Hamming weight is less than τ; otherwise, responds NO. Therefore, in order to find b_i from $\lambda := \mathsf{Enc}(b_i) = (\mathsf{Enc}(b_{i1}), \mathsf{Enc}(b_{i2}), \cdots, \mathsf{Enc}(b_{iM}))$, an adversary (say, $\mathcal{AS}$) first finds a bitstring whose Hamming weight is equal to the threshold $\tau + 1$ by repeatedly replacing the components of $(\mathsf{Enc}(0), \cdots, \mathsf{Enc}(0))$ with the corresponding components of λ until it gets rejected by $\mathcal{M}$. By using this bitstring with Hamming weight $\tau + 1$, the adversary is able to recover all bits of b_i one by one, as shown in Algorithm 1.

In the following attacks, we only consider the case when the authentication server $\mathcal{AS}$ (attacks 1 and 2) or the database $\mathcal{DB}$ (attack 3) is compromised, respectively.

Attack 1 - *Compromised* $\mathcal{AS}$: $\mathcal{AS}$ receives from $\mathcal{DB}$ the biometric reference template in encrypted form i.e. $\mathsf{Enc}(b_i) = c_1, \ldots, c_M$. Then, $\mathcal{AS}$ follows Algorithm 1. After executing the Algorithm 1, $\mathcal{AS}$ can successfully deduce all bits of b_i. The worst case complexity of this algorithm is $\max\big(2(\tau + M), 4\tau + M\big)$, where τ is the threshold. We may note here that the complexity of the center search attack is $\max\big(2\tau + M, 4\tau\big)$ [14]. After executing this algorithm $\mathcal{AS}$ has successfully deduced k out of the M bits of b_i, where $M - k = \tau$ are the maximum allowed errors. By following a similar algorithm for the remaining τ bits, it can recover all bits of b_i.

Attack 2 - *Compromised $\mathcal{AS}$*: A variation of the previous attack can be performed if $\mathcal{AS}$ has also at his disposal a valid value $\mathsf{Enc}(b'_i \oplus b_i)$. In this case Algorithm 1 can be executed twice: once for $\lambda = \mathsf{Enc}(b'_i)$ and once for $\lambda = \mathsf{Enc}(b'_i \oplus b_i)$. Thus, $\mathcal{AS}$ will be able to recover b_i and $b'_i \oplus b_i$ and subsequently b'_i.
Attack 3 - *Compromised $\mathcal{DB}$*: A variation of attack 1 can also be performed if $\mathcal{DB}$ is compromised. $\mathcal{DB}$ sets $\lambda = \mathsf{Enc}(t_1), \ldots, \mathsf{Enc}(t_M)$ if $M < N$; otherwise, $\lambda = \mathsf{Enc}(t_1), \ldots, \mathsf{Enc}(t_N), \mathsf{Enc}(0), \cdots, \mathsf{Enc}(0)$. This way, $\mathcal{DB}$ is able to recover t_j's by sending multiple queries to $\mathcal{M}$. Note that in the case of $M \leq N$, if it turns out that $t_j = 0$, for all $j = 1, \ldots, M$, then λ can be chosen to be the encryption of the remaining t_j's. Here we remark that $\mathcal{DB}$ on its own cannot send queries to $\mathcal{M}$ directly. But since $\mathcal{M}$ does not check the integrity of received queries, the adversary can replace $\mathcal{AS}$'s query to $\mathcal{M}$ with his own. In other words, here $\mathcal{DB}$ impersonates $\mathcal{AS}$ to $\mathcal{M}$.

Algorithm 1

```
Input: $\mathsf{Enc}(b_i) = c_1, \cdots, c_M$
Output: $b_i$
Initialise: $b_i = 00 \cdots 0$
For $k = 1$ to $M$:
    Set $\lambda = c_1, \ldots, c_k, \mathsf{Enc}(0), \ldots, \mathsf{Enc}(0)$
    If $\lambda$ is rejected Then
        break
    If $k == M$ Then
        Return centerSearch($b_i$)
Set $k^* = k$ and $b_{i,k^*} = 1$
If $k^* \geq 2$ Then
    For $k = 1$ to $k^* - 1$:
        Set $\lambda = c_1, \ldots, c_{k-1}, \mathsf{Enc}(0), c_{k+1} \ldots, c_{k^*}, \mathsf{Enc}(0), \ldots, \mathsf{Enc}(0)$
            If $\lambda$ is accepted Then
                $b_{i,k} = 1$
For $k = k^* + 1$ to $M$:
    Set $\lambda = c_1, \ldots, c_{k^*-1}, \mathsf{Enc}(0), \ldots, \mathsf{Enc}(0), c_k, \mathsf{Enc}(0), \ldots, \mathsf{Enc}(0)$
    If $\lambda$ is rejected Then
        $b_{i,k} = 1$
Return $b_i$
```

Thus, the Bringer *et al.* protocol is not secure or privacy-preserving in the *malicious insider* attack model. Because of the *bit-by-bit* encryption of the communication between the entities, the above presented attacks are straightforward and easy to mount. Plus, the complexity of the attacks is low. To mitigate the attacks, we next propose some modifications to the original protocol to improve its security and privacy preservation.

5 Countermeasure

Now, we propose modifications to the protocol under study to restore its security against the Attacks 1-3 presented in the previous section. Let us first discuss how we can protect the system against the Attack 1. We note that in this case the attacker has $\mathsf{Enc}(b_i)$ and wants to find out what b_i is.

If the matcher $\mathcal{M}$ does *not* directly compute the Hamming weight (HW) of the resulting bit-string from the decryption of the received ciphtertext, we may be able to protect the system against the Attack 1. So, in our modification, $\mathcal{M}$ shares two secret keys K_1 and K_2 with $\mathcal{S}$, a secret key K_3 with $\mathcal{AS}$, and two more secret keys K_4 and K_5 with $\mathcal{DB}$. These keys are used for symmetric key schemes, therefore the length of these keys are *not* as long as the length of the key for the GM encryption. As before, pk and sk are $\mathcal{M}$'s public and secret keys for GM encryption. $\mathcal{S}$ and $\mathcal{DB}$ also share a key $K_{\mathcal{S}\leftrightarrow\mathcal{DB}}$ that is used to derive a permutation π. In addition, $\mathcal{S}$ has a key K that it uses to encrypt the user identity ID_i.

During the *enrollment phase*, $\mathcal{S}$ stores (b_i, i) at $\mathcal{DB}$ and (id_i, i), where $\mathsf{id}_i = \mathsf{Enc}_K(\mathsf{ID}_i)$ (a symmetric key encryption of ID_i with key K), at $\mathcal{AS}$.

The main changes take place in the *authentication phase*.

Phase 1 - Communication $\mathcal{U}_i \to \mathcal{S} \to \mathcal{AS}$:

- $\mathcal{U}_i \to \mathcal{S}$: $\mathcal{U}_i$ provides a fresh biometric trait b'_i and ID_i to $\mathcal{S}$.
- $\mathcal{S} \to \mathcal{AS}$: $\mathcal{S}$ derives a permutation π using the key $K_{\mathcal{S}\leftrightarrow\mathcal{DB}}$ (shared with $\mathcal{DB}$) and permutes b'_i. Then, it generates two random bitstrings S and K'_1 of length M and encrypts $(b'_i)_\pi \oplus S$ with the public key pk (i.e. $a = \mathsf{Enc}((b'_i)_\pi \oplus S))$. In order to achieve *forward security*, K'_1 is generated to replace K_1. $\mathcal{S}$ proceeds to compute $\omega = \mathsf{Enc}_{K_1}(S, K'_1)$, an encryption of S and K'_1 with K_1, and computes $\sigma = \mathsf{TAG}(\omega, K_2)$. Also, $\mathcal{S}$ replaces K_1 with K'_1, which will be used in the next run of the protocol and deletes K_1 permanently. Finally, $\mathcal{S}$ sends a and (ω, σ) along with the encryption of the claimed identity $\mathsf{id}_i = \mathsf{Enc}_K(\mathsf{ID}_i)$ to $\mathcal{AS}$. Note that this encryption of ID_i is done to protect it from an adversary observing the communication from $\mathcal{S}$ to $\mathcal{AS}$.

Phase 2 - Communication $\mathcal{AS} \leftrightarrow \mathcal{DB}$:

- $\mathcal{AS} \to \mathcal{DB}$: $\mathcal{AS}$ extracts the index i from id_i and sends $d_j = \mathsf{Enc}(t_j)$ to $\mathcal{DB}$, for $j = 1, \cdots, N$, where t_j is the same as before.
- $\mathcal{DB} \to \mathcal{AS}$: $\mathcal{DB}$ derives π from $K_{\mathcal{S}\leftrightarrow\mathcal{DB}}$, generates two random bitstrings S' and K'_4 of length M, and computes $c_k = \mathsf{Enc}\big((b_{i,k})_\pi \oplus S'_k\big) = \prod_{j=1}^{N} \mathsf{Enc}(t_j)^{(b_{j,k})_\pi \oplus S'_k}$, where $1 \le k \le M$. $\mathcal{DB}$ then encrypts S' and K'_4 using K_4 to get $\omega' = \mathsf{Enc}_{K_4}(S', K'_4)$, and computes $\sigma' = \mathsf{TAG}(\omega', K_5)$. After that, $\mathcal{DB}$ replaces K_4 with K'_4 (to guarantee *forward security*) and deletes K_4. Finally, $\mathcal{DB}$ sends c, (ω', σ') to $\mathcal{AS}$.

Phase 3 - Communication $\mathcal{AS} \leftrightarrow \mathcal{M}$:

- $\mathcal{AS} \to \mathcal{M}$: $\mathcal{AS}$ computes $\lambda_k = a_k c_k = \mathsf{Enc}\big((b'_{i,k})_\pi \oplus S\big)\mathsf{Enc}\big((b_{i,k})_\pi \oplus S'\big) = \mathsf{Enc}\big((b'_{i,k} \oplus b_{i,k})_\pi \oplus S \oplus S'\big)$, for $1 \le k \le M$, computes $\sigma'' = \mathsf{TAG}(\lambda, K_3)$, and sends (ω, σ), (ω', σ'), and (λ, σ'') to $\mathcal{M}$.

- $\mathcal{M} \rightarrow \mathcal{AS}$: $\mathcal{M}$ first checks the authenticity of ω, ω' and λ by respectively running $\mathsf{VRFY}(\omega, \sigma, K_2)$, $\mathsf{VRFY}(\omega', \sigma', K_5)$, and $\mathsf{VRFY}(\lambda, \sigma'', K_3)$. If any one of them is not authentic, it outputs $\perp$ (*i.e.* aborts the protocol). Otherwise, it proceeds to obtain $S, K_1' \leftarrow \mathsf{Dec}_{K_1}(\omega)$, $S', K_4' \leftarrow \mathsf{Dec}_{K_4}(\omega')$, and $(b_i' \oplus b_i)_\pi \leftarrow \mathsf{Dec}(\lambda) \oplus S \oplus S'$; and replaces K_1 and K_4 with K_1' and K_4', respectively. Lastly, $\mathcal{M}$ checks whether the $\mathsf{HW}\big((b_i' \oplus b_i)_\pi\big) \leq \tau$ and sends the result of this control to $\mathcal{AS}$.

Phase 4 - Communication $\mathcal{AS} \rightarrow \mathcal{U}_i$: Finally, $\mathcal{AS}$ accepts or rejects the authentication request ($\mathsf{Out}_{\mathcal{AS}} = 1$ or $\mathsf{Out}_{\mathcal{AS}} = 0$ respectively) depending on the value returned by $\mathcal{M}$.

We should note here that the reason for replacing K_1 and K_4 with new independently generated K_1' and K_4', respectively, was to ensure *forward security* and thus to limit the damage in case the keys are compromised. The main question we want to answer now is: *How secure is the improved protocol against the presented attacks?* We address this question in the following section.

6 Security Analysis

Let us assess the security of the modified protocol. Before we proceed, we recall that we aim for security in the *malicious, but non-colluding* model, meaning that any entity can deviate from the protocol specifications but none of the entities may collude with each other. Therefore, we focus on security against *malicious insider* attacks. Since our primary goal is to assure security and privacy of biometric templates and user identity, we do not consider denial of service type of attacks in our analysis.

To begin with, let us analyse case-by-case what may happen when the entities, except for the sensor $\mathcal{S}$ which we assume to be honest and cannot be compromised, are malicious.

- *Attacker* $= \mathcal{AS}$: $\mathcal{AS}$ has knowledge of K_3, so it can send arbitrary queries to $\mathcal{M}$. In addition, it has at its disposal the encrypted user identity $\mathsf{id}_i = \mathsf{Enc}_K(\mathsf{ID}_i)$, encrypted biometric templates $\mathsf{Enc}((b_i')_\pi \oplus S)$ and $\mathsf{Enc}((b_i)_\pi \oplus S')$, $\omega = \mathsf{Enc}_{K_1}(S, K_1')$, $\omega' = \mathsf{Enc}_{K_4}(S', K_4')$, their authentication tags $\sigma = h_{K_2}(\omega)$, $\sigma' = h_{K_5}(\omega')$. He wants to use all this information to gain knowledge of b_i', b_i, and the linkage between ID_i and a biometric template b_i. It may arbitrarily deviate from the protocol specifications, except that it is not allowed to compromise or collude with another protocol entity. Note that $\mathcal{AS}$ can always cause denial of service to legitimate users by providing wrong input to $\mathcal{M}$.
- *Attacker* $= \mathcal{DB}$: $\mathcal{DB}$ has knowledge of all stored biometric templates and of K_4, K_5, S' and π. However, it does not know which b_i is

related to which user identity $\mathcal{U}_i$. It also does not know which user is attempting to authenticate himself to the server $\mathcal{AS}$. Therefore, its goal is to learn which user is trying to authenticate himself and to which user a biometric template belongs. It may also deviate from the protocol specifications, but it cannot collude with other entities.

- *Attacker* = $\mathcal{M}$: $\mathcal{M}$ has the secret keys sk, K_1, K_2, K_3, K_4, and K_5. Its goal is to distinguish whether two authentication attempts are from the same user. Since we assume that communications between the entities cannot be eavesdropped, it cannot use the secret keys to learn b_i and b'_i, unless it colludes with $\mathcal{AS}$.

The modified protocol is secure and preserves the privacy of biometric templates and user identity. In particular, none of the entities $\mathcal{AS}$, $\mathcal{DB}$, and $\mathcal{M}$, all *malicious but non-colluding* and PPT, can link a biometric template to a user identity and a malicious $\mathcal{DB}$ cannot distinguish whether two authentication attempts are from the same user. More precisely, Theorem 1 and Theorem 2 stated in Bringer *et al.* [1] also hold in the proposed modified protocol. We provide their proofs for the modified protocol in Appendix B. Finally, the modified proposed protocol is secure against malicious authentications servers $\mathcal{AS}$ as stated in the following theorem (we provide its proof in Appendix C).

Theorem 1. *If the Assumptions 1-3 hold and if (a) S and S' are generated using ϵ-secure PNGs (cf. Definition 3), (b) the symmetric encryption schemes SKE used between the sensor $\mathcal{S}$ and the matcher $\mathcal{M}$, and between the database $\mathcal{DB}$ and the matcher $\mathcal{M}$, is $\mathsf{IND\text{-}COA}$-secure, and (c) the GM scheme is $\mathsf{IND\text{-}CPA}$-secure. Then, our modified protocol is secure against any* malicious authentication server $\mathcal{AS}$.

Forward Security Informally, *forward security* means that the disclosure of a secret key material does not compromise the secrecy of the exchanged communications from previous rounds. As we briefly mentioned in the previous section, our modified protocol has *forward security.* In particular, the biometric templates exchanged will not be affected by a future disclosure of the secret key used to encrypt them. The original protocol, on the other hand, does not provide forward security. This is because if the matcher $\mathcal{M}$'s secret key is compromised, then all biometric templates exchanged in the past can be learned. But in the modified protocol, the adversary learns the biometric templates in the present round (and onwards) only.

Comparison In comparison with the original protocol, in our modification each protocol entity performs additional cryptographic computations such as, symmetric key encryption/decryption, MAC generation/verification, and generation of pseudo-random numbers. In particular, in the case of $\mathcal{S}$, in the original protocol, $\mathcal{S}$ only computes the encryption of the fresh biometric samples using the GM encryption. But in the modified protocol, in addition to that, $\mathcal{S}$ first generates S, K_1' and then computes $\omega = \mathsf{Enc}_{K_1}(S, K_1')$ and $\mathsf{Enc}_K(\mathsf{ID}_i)$ using a symmetric encryption and computes an authentication tag for ω using a suitable MAC. In the case of $\mathcal{AS}$, the only additional computation done in the modified protocol is the authentication tag generation for λ, *i.e.* $\sigma'' = h_{K_3}(\lambda)$. In the case of $\mathcal{DB}$, in the modified protocol, $\mathcal{DB}$ first generates S', K_4' and then $\omega' = \mathsf{Enc}_{K_4}(S', K_4')$, $\sigma' = \mathsf{TAG}_{K_5}(\omega')$. In the case of $\mathcal{M}$, in the modified protocol, the additional computations done by $\mathcal{M}$ are: $\mathsf{VRFY}(\omega, \sigma, K_2)$, $\mathsf{VRFY}(\omega', \sigma', K_5)$, $\mathsf{VRFY}(\lambda, \sigma'', K_3)$, $\mathsf{Dec}_{K_1}(\omega)$ and $\mathsf{Dec}_{K_4}(\omega')$. Also, it XORs S and S' with $\mathsf{Dec}(\lambda)$. It is evident that in the modified protocol, each system entity performs some additional computations than required in the original protocol. However, as they are symmetric cryptographic operations, these computations are *not* as heavy as those done in the GM encryption.

7 Conclusions

We investigated the security of a *privacy-preserving biometric authentication* protocol proposed by Bringer *et al.* that uses the Goldwasser-Micali cryptosystem in the *malicious* attack model. We presented a simple attack algorithm that can be employed to mount a number of attacks on the system to either obtain the reference biometric template (b_i) or the identity (ID_i) of a user associated with a biometric template (b_i). Furthermore, we proposed an improved version of the Bringer *et al.* [1] protocol and proved its security against *malicious, but non-colluding* insider attacks. As future work, we would like to investigate how to achieve security and privacy against colluding internal adversaries.

8 Acknowledgements

We would like to thank Serge Vaudenay for suggesting the attack algorithm and Julien Bringer for the helpful discussions. This work was partially supported by the FP7-STREP project "BEAT: Biometric Evaluation and Testing", grant number: 284989.

References

1. Bringer, J., Chabanne, H., Izabachène, M., Pointcheval, D., Tang, Q., Zimmer, S.: An application of the Goldwasser-Micali cryptosystem to biometric authentication. In: ACISP'07. LNCS (2007) 96–106

2. Ouafi, K., Vaudenay, S.: Strong Privacy for RFID Systems from Plaintext-Aware Encryption. In: CANS'12. Volume 7712 of LNCS. Springer (2012) 247–262
3. Vaudenay, S.: On Privacy Models for RFID. In: ASIACRYPT'07. Volume 4833 of LNCS., Springer (2007) 68–87
4. Hermans, J., Pashalidis, A., Vercauteren, F., Preneel, B.: A new RFID privacy model. In: ESORICS'11. LNCS (2011) 568–587
5. Juels, A., Weis, S.A.: Authenticating pervasive devices with human protocols. In: Proceedings of the 25th annual international conference on Advances in Cryptology. CRYPTO'05 (2005) 293–308
6. Gilbert, H., Robshaw, M.J.B., Sibert, H.: Active attack against HB+: a provably secure lightweight authentication protocol. Electronic Letters **41** (2005) 1169–1170
7. Penrose, L.: Dermatoglyphic topology. Nature **205** (1965) 544–546
8. Bolling, J.: A window to your health. Jacksonville Medicine, Special Issue: Retinal Diseases **51** (2000)
9. Rivest, R.L., Adleman, L., Dertouzos, M.L.: On data banks and privacy homomorphisms. In: Foundations of Secure Computation. Academic Press (1978) 165–179
10. Rabin, M.O.: How to exchange secrets with oblivious transfer. Technical Report TR-81, Aiken Computation Lab, Harvard University (1981)
11. Even, S., Goldreich, O., Lempel, A.: A randomized protocol for signing contracts. Commun. ACM **28**(6) (1985) 637–647
12. Goldwasser, S., Micali, S.: Probabilistic encryption & how to play mental poker keeping secret all partial information. In: Proceedings of ACM symposium on Theory of computing. STOC '82 (1982) 365–377
13. Barbosa, M., Brouard, T., Cauchie, S., Sousa, S.M.: Secure Biometric Authentication with Improved Accuracy. In: ACISP'08. LNCS (2008) 21–36
14. Simoens, K., Bringer, J., Chabanne, H., Seys, S.: A framework for analyzing template security and privacy in biometric authentication systems. IEEE Transactions on Information Forensics and Security **7**(2) (2012) 833–841
15. Bringer, J., Chabanne, H.: An authentication protocol with encrypted biometric data. In: AFRICACRYPT'08. LNCS (2008) 109–124
16. Lipmaa, H.: An oblivious transfer protocol with log-squared communication. In: ISC. Volume 3650 of LNCS. (2005) 314–328
17. Stoianov, A.: Cryptographically secure biometrics. SPIE 7667, Biometric Technology for Human Identification VII **76670C** (2010) 76670C–76670C–12
18. Blum, M., Goldwasser, S.: An efficient probabilistic public key encryption scheme which hides all partial information. In: CRYPTO'84. (1985) 289–299
19. Menezes, A., van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press (1996)

A Appendix

$\mathsf{Exp}^{\mathsf{IND\text{-}COA}}_{\mathsf{SKE},\mathcal{A}}$ is the $\mathsf{IND\text{-}COA}$ game against an SKE scheme defined as follows.

```
$\mathsf{Exp}^{\mathsf{IND\text{-}COA}}_{\mathsf{SKE},\mathcal{A}}$: $K$          $\leftarrow \mathsf{KeyGen}(1^\ell)$
                 $m_0, m_1$                  $\leftarrow A_1(1^\ell)$
                 $c$                         $\leftarrow \mathsf{Enc}(m_\beta, K),\ \beta \xleftarrow{R} \{0,1\}$
                 $\beta'$                    $\leftarrow A_2(m_0, m_1, c)$
Return $1$ if $\beta' = \beta$, $0$ otherwise
```

The adversary's advantage in this game is defined as $\mathsf{Adv}^{\mathsf{IND\text{-}COA}}_{\mathsf{SKE},\mathcal{A}} = \left|2\Pr\left(\mathsf{Exp}^{\mathsf{IND\text{-}COA}}_{\mathsf{SKE},\mathcal{A}} = 1\right) - 1\right|$. A SKE scheme is said to be $\mathsf{IND\text{-}COA}$-secure, if $\forall$ PPT adversary $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{IND\text{-}COA}}_{\mathsf{SKE},\mathcal{A}} \leq \mathsf{negl}(\ell)$, where (and below) $\mathsf{negl}(\ell) : \mathbf{N} \mapsto [0,1]$ is a negligible function meaning that for all positive polynomials P and all sufficiently large $\ell \in \mathbf{N}$, we have $\mathsf{negl}(\ell) < 1/P(\ell)$.

$\mathsf{Exp}^{\mathsf{IND\text{-}CPA}}_{\mathsf{GM},\mathcal{A}}$ is the $\mathsf{IND\text{-}CPA}$ game against the GM encryption and is defined as in the previous game, but now the adversary has access to the public key. This scheme is said to be $\mathsf{IND\text{-}CPA}$ secure if $\forall$ PPT adversary $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{GM},\mathcal{A}} = \left|2\Pr\left(\mathsf{Exp}^{\mathsf{IND\text{-}CPA}}_{\mathsf{GM},\mathcal{A}} = 1\right) - 1\right| \leq \mathsf{negl}(\ell)$.

B Appendix

Here we prove the Theorem 1 and 2 in Bringer *et al.* [1] in the case of our improved protocol.

Theorem 2. *For any ID_{i_0} and two biometric templates b'_{i_0}, b'_{i_1}, where i_0, $i_1 \geq 1$ and b'_{i_0} is the biometric template related to ID_{i_0}, any of the malicious, but not colluding $\mathcal{AS}$, $\mathcal{DB}$, and $\mathcal{M}$ can only distinguish between $(\mathit{ID}_{i_0}, b'_{i_0})$ and $(\mathit{ID}_{i_0}, b'_{i_1})$ with a negligible advantage.*

Proof. Since $\mathcal{DB}$ and $\mathcal{M}$ have no access to user identities, their advantage is 0 in distinguishing between $(\mathsf{ID}_{i_0}, b'_{i_0})$ and $(\mathsf{ID}_{i_0}, b'_{i_1})$.

In the case of $\mathcal{AS}$, it has access to $\mathsf{id}_{i_0} = \mathsf{Enc}_K(\mathsf{ID}_{i_0})$, where $\mathsf{Enc}_K(\cdot)$ is a symmetric encryption with the sensor $\mathcal{S}$'s key K. However, even if $\mathcal{AS}$ knows ID_{i_0}, it cannot distinguish between $(\mathsf{ID}_{i_0}, b'_{i_0})$ and $(\mathsf{ID}_{i_0}, b'_{i_1})$, except with a negligible probability, as we see below.

Suppose that $\mathcal{AS}$ has a non-negligible advantage in distinguishing between $(\mathsf{ID}_{i_0}, b'_{i_0})$ and $(\mathsf{ID}_{i_0}, b'_{i_1})$. Then, we can construct an adversary $\mathcal{A}$, consisting of algorithms A_1 and A_2, such that $\mathcal{A}$'s advantage in the following game is non-negligible, contradicting the $\mathsf{IND\text{-}CPA}$-security of GM cryptosystem:

$$\boxed{\begin{array}{lll}
\mathsf{Exp}^{\mathsf{IND\text{-}CPA}}_{\mathsf{GM},\mathcal{A}}: & \mathsf{pk} = (n, x), \mathsf{sk} = (p, q) & \leftarrow \mathsf{KeyGen}(1^{\ell}) \\
& m_{i_0} = m'_{i_0} \oplus S, m_{i_1} = m'_{i_1} \oplus S, m'_{i_0} \neq m'_{i_1} & \leftarrow A_1(1^{\ell}, \mathsf{pk}) \\
& c & \leftarrow \mathsf{Enc}(m_{i_\alpha}),\ \alpha \xleftarrow{R} \{0,1\} \\
& \alpha' = \mathsf{guess}_{\mathcal{AS}} & \leftarrow A_2\big(\mathcal{AS}(m_{i_0}, m_{i_1}, c, \mathsf{pk})\big) \\
\multicolumn{3}{l}{\mathsf{Return}\ 1 \text{ if } \beta' = \beta,\ 0 \text{ otherwise}}
\end{array}}$$

In the experiment, A_2 simulates the biometric authentication protocol by letting pk be $\mathcal{M}$'s public key and storing m'_{i_0} and m'_{i_1} in $\mathcal{DB}$. A_2 then asks $\mathcal{AS}$ to guess β from $c = \mathsf{Enc}(m_{i_\beta}) = \mathsf{Enc}(m'_{i_\beta} \oplus S)$ and returns β as the guess for α. So, $\mathcal{A}$ wins if $\mathcal{AS}$ wins in his guess. Thus, $\mathcal{AS}$ can only distinguish between $(\mathsf{ID}_{i_0}, b'_{i_0})$ and $(\mathsf{ID}_{i_0}, b'_{i_1})$ with negligible probability. $\square$

The next theorem shows that a malicious database $\mathcal{DB}$ cannot distinguish whether two authentication attempts are from the same user.

Theorem 3. *For any two users $\mathcal{U}_{i_0}$ and $\mathcal{U}_{i_1}$, where i_0, $i_1 \geq 1$, if $\mathcal{U}_{i_\beta}$ where $\beta \in \{0, 1\}$ makes an authentication attempt, then the malicious $\mathcal{DB}$ can only guess β with a negligible advantage. Here, the adversary's advantage is defined as $\left|\Pr\{\beta = \beta'\} - 1/2\right|$, where β' is $\mathcal{DB}$'s guess.*

Proof (of Theorem 3). $\mathcal{DB}$ guesses β from $\mathsf{Enc}(t_j)$, for $j = 1, \cdots, N$, where $t_j = 1$ when $j = i_\beta$ ($\beta \in \{0, 1\}$), otherwise $t_j = 0$. The proof is similar to that of Theorem 2 in Bringer *et al.* [1].

Suppose that $\mathcal{DB}$ can guess β with non-negligible advantage. Then, we can construct a PPT adversary $\mathcal{A}$, consisting of A_1 and A_2, that uses $\mathcal{DB}$ as a blackbox to win in the following game with non-negligible advantage; contradicting the $\mathsf{IND\text{-}CPA}$-security of GM cryptosystem:

$$\boxed{\begin{array}{lll}
\mathsf{Exp}^{\mathsf{IND\text{-}CPA}}_{\mathsf{GM},\mathcal{A}}\text{: } \mathsf{pk} = (n, x), \mathsf{sk} = (p, q) & \leftarrow & \mathsf{KeyGen}(1^\ell) \\
\qquad m_0 = 0, m_1 = 1 & \leftarrow & A_1(1^\ell, \mathsf{pk}) \\
\qquad c & \leftarrow & \mathsf{Enc}(m_\alpha),\ \alpha \xleftarrow{R} \{0, 1\} \\
\qquad \alpha' = \mathsf{guess}_{\mathcal{DB}} & \leftarrow & A_2(\mathcal{DB}(\mathsf{Enc}(t_j), \mathsf{pk})),\ j = 1, \cdots, N \\
\text{Return } 1 \text{ if } \beta' = \beta,\ 0 \text{ otherwise} & &
\end{array}}$$

where $\mathsf{Enc}(t_{i_1}) = c$, $\mathsf{Enc}(t_{i_0}) = y^2xc, y \xleftarrow{R} \mathbb{Z}_n^\star$, $t_j = 0$, $\forall j \neq i_0$, i_1. Note that if $c = \mathsf{Enc}(m_0)$, then y^2xc is not a quadratic residue $\mod n$, so $\mathcal{DB}$'s guess, which is 0, and α coincide. Similarly, if $c = \mathsf{Enc}(m_1)$, then y^2xc is a quadratic residue $\mod n$, so $\mathcal{DB}$'s guess, which is 1, and α coincide. Hence, $\mathcal{DB}$'s advantage of guessing β correctly should be negligible. □

C Appendix

Here, we present the proof of Theorem 1. Let $\mathcal{A}$ be any PPT adversary, consisting of two algorithms A_1 and A_2. Let us consider the following game against the modified biometric authentication protocol Π. Let KeyGen be an algorithm that generates both symmetric and asymmetric keys needed in the protocol, upon 1^ℓ (a string of 1s of length ℓ) as an input. As usual, ℓ is a security parameter.

$$\boxed{\begin{array}{lll}
\mathsf{Exp}^{\mathsf{biometric\text{-}privacy}}_{\Pi,\mathcal{A}}\text{: } (\mathsf{pk}, \mathsf{sk}), K, K_{\mathcal{S}\leftrightarrow\mathcal{DB}}, K_1, \cdots, K_5 & \leftarrow & \mathsf{KeyGen}(1^\ell) \\
\qquad S, S' & \leftarrow & \mathsf{PNG}(s),\ s \xleftarrow{R} \{0, 1\}^{r>0} \\
\qquad a,\ (\omega, \sigma),\ c,\ (\omega', \sigma') & \leftarrow & \Pi(\mathsf{pk}, K, K_{\mathcal{S}} \leftrightarrow \mathcal{DB}, K_1, K_2, K_4, K_5) \\
\qquad \gamma_0 = (\lambda^{(0)}, \sigma_0''),\ \gamma_1 = (\lambda^{(1)}, \sigma_1'') & \leftarrow & A_1(a, c, K_3, (\omega, \sigma), (\omega', \sigma')) \\
\qquad \beta & \xleftarrow{R} & \{0, 1\} \\
\qquad \mathsf{Out}_{\mathcal{M}} & \leftarrow & \mathcal{M}(\gamma_\beta, (\omega, \sigma), (\omega', \sigma'), \mathsf{sk}, K_1, \cdots, K_5) \\
\qquad \beta' & \leftarrow & A_2(\gamma_0, \gamma_1, \mathsf{Out}_{\mathcal{M}}) \\
\text{Return } (\beta' = \beta, \mathsf{Out}_{\mathcal{M}}) & &
\end{array}}$$

The adversary's advantage $\mathsf{Adv}_{\Pi,\mathcal{A}}^{\text{biometric-privacy}}$ at the end of this game is defined as $\mathsf{Adv}_{\Pi,\mathcal{A}}^{\text{biometric-privacy}} = |\Pr\{\beta' = \beta\} - 1/2|$, where $\beta \in \{0,1\}$ is $\mathcal{M}$'s choice and β' is the adversary's guess for β. We say that the biometric authentication protocol is secure against malicious $\mathcal{AS}$, if $\mathsf{Adv}_{\Pi,\mathcal{A}}^{\text{biometric-privacy}} \leq \mathsf{negl}(\ell)$.

Note that as stated in Assumption 2, we assume that the adversary does not have access to an acceptable biometric template, because otherwise the adversary can easily produce two challenges so that it wins the above experiment with non-negligible advantage.

Proof (of Theorem 1). Case 1. $\mathsf{HW}\big(\mathsf{Dec}(\lambda^{(\beta)}) \oplus S \oplus S'\big) \leq \tau$, for $\forall\beta \in \{0,1\}$. In this case, $\mathcal{M}$'s output always be the same (i.e., $\mathsf{Out}_{\mathcal{M}} = \mathsf{YES}$.) Hence, the adversary's advantage in this case is 0.

Case 2. $\mathsf{HW}\big(\mathsf{Dec}(\lambda^{(\beta)}) \oplus S \oplus S'\big) > \tau$, for $\forall\beta \in \{0,1\}$. Also in this case, $\mathcal{M}$'s output always be the same (i.e., $\mathsf{Out}_{\mathcal{M}} = \mathsf{NO}$). Hence, the adversary's advantage in this case is 0.

Case 3. $\mathsf{HW}\big(\mathsf{Dec}(\lambda^{(\beta)}) \oplus S \oplus S'\big) \leq \tau$ and $\mathsf{HW}\big(\mathsf{Dec}(\lambda^{(1-\beta)}) \oplus S \oplus S'\big) > \tau$. Suppose that a PPT adversary $\mathcal{A}$ has a non-negligible advantage δ of winning the game $\mathsf{Exp}_{\Pi,\mathcal{A}}^{\text{biometric-privacy}}$. Then we can construct a PPT adversary $\bar{\mathcal{A}}$ that wins in $\mathsf{Exp}_{\mathsf{SKE},\bar{\mathcal{A}}}^{\mathsf{IND\text{-}COA}}$ and/or $\mathsf{Exp}_{\mathsf{GM},\bar{\mathcal{A}}}^{\mathsf{IND\text{-}CPA}}$ with advantage δ. The construction of such an adversary $\bar{\mathcal{A}}$, for example in the case of $\mathsf{Exp}_{\mathsf{SKE},\bar{\mathcal{A}}}^{\mathsf{IND\text{-}COA}}$, may proceed as follows:

$$
\boxed{
\begin{array}{lll}
\mathsf{Exp}_{\mathsf{SKE},\bar{\mathcal{A}}}^{\mathsf{IND\text{-}COA}}: & K' & \leftarrow \mathsf{KeyGen}(1^{\ell}) \\
 & \begin{cases} m_0 = (m_{00}, m_{01}), & \mathsf{HW}(m_{00}) \leq \tau \ \&\ \mathsf{HW}(\neg m_{00}) > \tau \\ m_1 = (m_{10}, m_{11}), & \mathsf{HW}(m_{10}) > \tau \ \&\ \mathsf{HW}(\neg m_{10}) \leq \tau \end{cases} & \leftarrow \bar{\mathcal{A}}(1^{\ell}) \\
 & c & \leftarrow \mathsf{Enc}_{K'}(m_{\alpha}),\ \alpha \xleftarrow{R} \{0,1\} \\
 & \alpha' = \begin{cases} \mathsf{guess}_{\mathcal{A}}, & if\ \mathsf{Out}_{\mathcal{M}} = \mathsf{YES}, \\ 1 - \mathsf{guess}_{\mathcal{A}}, & if\ \mathsf{Out}_{\mathcal{M}} = \mathsf{NO}. \end{cases} & \leftarrow \bar{\mathcal{A}}(\mathcal{A}(m_0, m_1, c)) \\
\multicolumn{3}{l}{\mathsf{Return}\ 1 \text{ if } \alpha' = \alpha,\ 0 \text{ otherwise}}
\end{array}
}
$$

where $|m_{00}| = |m_{10}| = |S|$ and $|m_{01}| = |m_{11}| = |K_1|$. $\bar{\mathcal{A}}$ then simulates the biometric authentication protocol and replaces, without loss of generality, the symmetric key encryption scheme between the sensor $\mathcal{S}$ and the matcher $\mathcal{M}$. More precisely, $\bar{\mathcal{A}}$ replaces ω with the challenge ciphertext c and ω' with an encryption of a bitstring of all zeros. $\bar{\mathcal{A}}$ then runs A_1 to obtain $\gamma_0 = (\mathsf{Enc}_{\mathsf{GM}}(0), \sigma_0'')$ and $\gamma_1 = (\mathsf{Enc}_{\mathsf{GM}}(1), \sigma_1'')$, where 0 and 1 respectively stand for bitstrings of all zeros and ones. If $\mathsf{Out}_{\mathcal{M}} = \mathsf{YES}$, $\bar{\mathcal{A}}$ outputs $\mathcal{A}$'s guess β' as α'; if $\mathsf{Out}_{\mathcal{M}} = \mathsf{NO}$, $\bar{\mathcal{A}}$ outputs $1 - \beta'$ as α'. This is because, when $\mathsf{Out}_{\mathcal{M}} = \mathsf{YES}$, $\beta' = 0$ would indicate that $\mathsf{HW}(m_{\alpha 0}) \leq \tau$, and $\beta' = 1$ would indicate that $\mathsf{HW}(1 \oplus m_{\alpha 0}) > \tau$. And similarly, when $\mathsf{Out}_{\mathcal{M}} = \mathsf{NO}$, $\beta' = 0$ would indicate that $\mathsf{HW}(m_{\alpha 0}) > \tau$, and $\beta' = 1$ would indicate that $\mathsf{HW}(1 \oplus m_{\alpha 0}) \leq \tau$. Hence, if $\mathcal{A}$ wins, so does $\bar{\mathcal{A}}$. □