A Privacy-Preserving Biometric Authentication Protocol Revisited

Aysajan Abidin and Aikaterini Mitrokotsa

Chalmers University of Technology, Gothenburg, Sweden
{aishajia, aikaterini.mitrokotsa}@chalmers.se

Abstract. Biometric authentication establishes the identity of an individual based on biometric templates (i.e. fingerprints, retina scans etc.). Although biometric authentication has important advantages and many applications, it also raises serious security and privacy concerns. In this parer, we investigate a *privacy-preserving 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). We present an attack algorithm that can be employed to mount a number of attacks on the protocol under investigation and propose an improved version of the Bringer *et al.* protocol that combats the presented attacks.

Keywords: Biometrics, privacy-preserving biometric authentication, homomorphic encryption, center search attack.

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. However, biometric authentication has also many serious security and privacy implications. Compromised biometric templates may lead to serious threats to identity (*i.e.* identity theft and fraud), while the inherent irrevocability of biometrics renders this risk even more serious. Biometric information may reveal very sensitive and private information such as genetic and medical information. More precisely, it has been proven that fingerprints may reveal genetic information [9], while retina scans may reveal diseases such as diabetes and strokes [2]. Additional issues of linkability, profiling and tracking of individuals are raised by cross-matching biometric traits. Due to the serious privacy implications that biometric authentication creates, the need for privacy-preserving biometric authentication protocols is of utmost importance. Many existing protocols rely on the use of secure multi-party computation techniques and among others homomorphic encryption schemes.

In this paper, we review a privacy-preserving biometric authentication protocol that has been proposed by Bringer *et al.* [3] and relies on homomorphic encryption and more precisely uses the Goldwasser-Micali cryptosystem [5]. Furthermore, we present an algorithm that can be employed by an adversary to mount a number of attacks to the protocol under investigation. Finally, we propose an improved protocol that can be used to combat these attacks.

The protocol proposed by Bringer *et al.* [3] as well as some other existing ones [1,11] rely on a distributed architecture of the biometric authentication system. This architecture is depicted in Fig. 1, and is composed of the following entities: the user ($\mathcal{U}_i$), where $1 \leq i \leq N$ and N is the number of users, the sensor ($\mathcal{S}$), the authentication server ($\mathcal{AS}$), the database ($\mathcal{DB}$), and the matcher ($\mathcal{M}$).

In the *enrollment phase* the user $\mathcal{U}_i$ registers his/her biometric data b_i which is stored in the database $\mathcal{DB}$. In the *authentication phase* a user U_i, $1 \leq i \leq N$ first provides a fresh biometric trait b'_i and his/her identity ID_i to the sensor $\mathcal{S}$, which in turn forwards these data to the authentication server $\mathcal{AS}$. $\mathcal{AS}$ then asks the database $\mathcal{DB}$ for 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 $\mathcal{M}$, who checks whether b_i and b'_i matches and sends back the result of the comparison to $\mathcal{AS}$. Usually this comparison involves a threshold τ which denotes the maximum allowed difference between b_i and b'_i. Finally, $\mathcal{AS}$ outputs to the user U_i the output of the authentication process $\mathsf{Out}_{\mathcal{AS}} = 1$ or $\mathsf{Out}_{\mathcal{AS}} = 0$, depending on whether U_i is successfully authenticated or not, respectively.

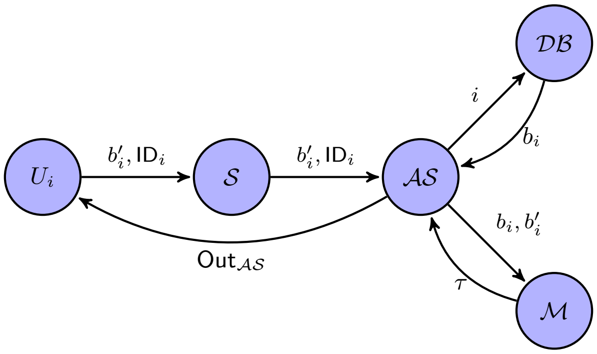


Fig. 1. Schematic description of a distributed biometric authentication system.

We assume 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 [8,12,6,7,4]. 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.

2 Preliminaries

We use the following informal definition of privacy-preserving biometrics.

Definition 1 (Privacy-preserving biometric authentication protocol). *We define a protocol as* privacy-preserving *if an adversary is not able to recover any of the following information, if they are not already known: a fresh biometric trait b'_i, a stored biometric template b_i and/or the correspondence between the identity ID_i to the stored template b_i.*

Adversarial model: We consider as an adversary any external or internal entity that is able 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.

Assumption 1 *We assume that the sensor $\mathcal{S}$ is honest and has not been compromised.*

Assumption 2 *We assume that the biometric authentication system has a limit on the maximum allowed 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 $\mathcal{M}$.*

Assumption 3 *We assume that none of the entities $\mathcal{AS}$, $\mathcal{DB}$, $\mathcal{M}$ may collude with each other.*

3 The Bringer *et al.* protocol

Bringer *et al.* [3] have proposed a protocol for privacy-preserving biometric authentication that follows the above described model and involves four entities in the biometric authentication process. Let $(\mathsf{pk}, \mathsf{sk})$ be the public and private key pair for the GM cryptosystem. 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. The authentication server $\mathcal{AS}$ also stores the mapping (ID_i, i)

for $i \in \{1, \ldots, N\}$ where i corresponds to user $\mathcal{U}_i$ and N is the total number of users of the biometric authentication system. Furthermore, the database $\mathcal{DB}$ stores the reference biometric template b_i. We use $\mathsf{Enc}(\cdot)$ (and $\mathsf{Dec}(\cdot)$) to denote the GM encryption (and decryption). $\mathsf{Enc}(b_i)$ denotes the *bit-by-bit* encryption of each bit 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. For the GM encryption function $\mathsf{Enc}(\cdot)$, it holds that $\mathsf{Enc}(a)\mathsf{Enc}(b) = \mathsf{Enc}(a \oplus b)$.

In the enrolment phase, user $\mathcal{U}_i$ registers (b_i, i) at the database $\mathcal{DB}$, and (ID_i, i) at the authentication server $\mathcal{AS}$. The authentication phase can be discriminated into the following phases:

- Phase 1 - Communication $\mathcal{U}_i \to \mathcal{S} \to \mathcal{AS}$: In this phase we discriminate two steps:
  (i) The user $\mathcal{U}_i$ provides a fresh biometric trait b'_i and his identity ID_i to the sensor $\mathcal{S}$.
  (ii) Then, the sensor $\mathcal{S}$ sends the fresh biometric b'_i encrypted with the public key pk (*i.e.* $\mathsf{Enc}(b'_i)$) as well as the claimed identity ID_i of the user $\mathcal{U}_i$ to the authentication server $\mathcal{AS}$.
- Phase 2 - Communication $\mathcal{AS} \leftrightarrow \mathcal{DB}$: This phase can be divided into two steps:
  (i) The authentication server $\mathcal{AS}$ gets i from ID_i and then using a PIR mechanism sends the identity i of user $\mathcal{U}_i$ 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 $j = i$, otherwise $t_j = 0$.
  (ii) The database $\mathcal{DB}$ 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 the authentication server $\mathcal{AS}$.
- Phase 3 - Communication $\mathcal{AS} \leftrightarrow \mathcal{M}$: This phase includes two steps:
  (i) $\mathcal{AS}$ 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 the permuted vector $\lambda_k = v_{\pi(k)}$ $(1 \leq k \leq M\ )$ to $\mathcal{M}$.
  (ii) $\mathcal{M}$ decrypts the permuted vector λ_k and checks whether the Hamming weight (HW) of the decrypted permuted vector is less than a predefined threshold τ. The result of this control is sent to $\mathcal{AS}$.
- Phase 4 - Communication $\mathcal{AS} \to \mathcal{U}_i$: 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 Attacks

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.* [10] for details on the *center search attack*. Before we proceed, let us present our assumptions about what each entity does in a normal situation, and what his/her goals are when/if compromised.

- The sensor $\mathcal{S}$ knows b'_i as well as the identity ID_i of the user $\mathcal{U}_i$, and communicates with the authentication server $\mathcal{AS}$; can simulate $\mathcal{AS}$ and wants to find out b_i if compromised. In this paper, however, we assume that the sensor is honest and has not been compromised; cf. Assumption 1.
- The authentication server $\mathcal{AS}$ knows ID_i, communicates with the database $\mathcal{DB}$, the matcher $\mathcal{M}$ and the sensor $\mathcal{U}_i$; can simulate the sensor $\mathcal{S}$ and wants to find out b'_i and/or b_i if compromised.
- The database $\mathcal{DB}$ knows b_i and communicates with the authentication server $\mathcal{AS}$; can query the matcher $\mathcal{M}$ and wants to find out b'_i and/or the identity of $\mathcal{U}_i$ if compromised.
- The matcher $\mathcal{M}$, on its own, cannot mount any attack, unless it is colluding with the other entities. In that case, the matcher can give away the secret keys. We, however, assume that the entities are not allowed to collude with each other.

Algorithm 1

```
Input: $\mathsf{Enc}(b_i) = c_1, \cdots, c_M$
Output: $b_i$
Initialise: $b_i = 00\cdots0$
For $k = 1$ to $M$:
    Set $\lambda = c_1, \ldots, c_k, \mathsf{Enc}(0), \ldots, \mathsf{Enc}(0)$
    Send $\lambda$ to the matcher $\mathcal{M}$
    If $\lambda$ is rejected Then
        break
    EndIf
    If $k == M$ Then
        Return centerSearch($b_i$)
    EndIf
EndFor
Set $k^* = k$
Set $b_{i,k^*} = 1$
If $k^* \geq 2$ Then
    For $k = 1$ to $k^* - 1$:
        Set $\lambda = c_1, \ldots, c_{k-1}, \mathsf{Enc}(0), c_{k+1} \ldots, c_{k^*}, \mathsf{Enc}(0), \ldots, \mathsf{Enc}(0)$
            Send $\lambda$ to the matcher $\mathcal{M}$
            If $\lambda$ is accepted Then
                $b_{i,k} = 1$
            EndIf
    EndFor
EndIf
For $k = k^* + 1$ to $M$:
    Set $\lambda = c_1, \ldots, c_{k^*-1}, \mathsf{Enc}(0), \ldots, \mathsf{Enc}(0), c_k, \mathsf{Enc}(0), \ldots, \mathsf{Enc}(0)$
    Send $\lambda$ to the matcher $\mathcal{M}$
    If $\lambda$ is rejected Then
        $b_{i,k} = 1$
    EndIf
EndFor
Return $b_i$
```

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. So in the case of the former, his goal is to learn b_i; and in the case of the latter, his goal is to learn the identity ID_i of user $\mathcal{U}_i$.

- **Attack 1 - *Compromised* $\mathcal{AS}$:** The authentication server $\mathcal{AS}$ receives from the database $\mathcal{DB}$ 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 $\max\big(2\tau + M, 4\tau\big)$ [10]. 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 τ not recovered bits, he will be able to recover all bits of b_i.
- **Attack 2 - *Compromised* $\mathcal{AS}$:** A variation of the previous attack can be performed if the authentication server $\mathcal{AS}$ 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. $\mathcal{DB}$ by sending multiple queries to $\mathcal{M}$ will be able to recover t_j's. In this case, $\lambda = \mathsf{Enc}(t_1), \ldots, \mathsf{Enc}(t_N), \mathsf{Enc}(0), \ldots, \mathsf{Enc}(0)$ if $M > N$; otherwise, $\lambda = \mathsf{Enc}(t_1), \ldots, \mathsf{Enc}(t_M)$. 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 the authentication server $\mathcal{AS}$'s query to $\mathcal{M}$ with his own. In other words, here $\mathcal{DB}$ impersonates $\mathcal{AS}$ to $\mathcal{M}$.

5 Countermeasure

In mitigate the attacks, we let the matcher $\mathcal{M}$ share two secret keys K_1 and K_2 with the sensor $\mathcal{S}$, a secret key K_3 with the authentication server $\mathcal{AS}$, and two more secret keys K_4 and K_5 with the database $\mathcal{DB}$. 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 cryptosystem. As before, pk and sk are $\mathcal{M}$'s public and secret keys for GM encryption. The sensor $\mathcal{S}$ and the database $\mathcal{DB}$ also share a key $K_{\mathcal{S}\leftrightarrow\mathcal{DB}}$ that is used to derive a permutation π. In addition, the sensor $\mathcal{S}$ has a secret key K that he uses to encrypt the user identity ID_i. We use $\mathsf{Enc}_K(\cdot)$ (and $\mathsf{Dec}_K(\cdot)$) to denote symmetric key encryption (and decryption) with key K. By $h_K(\cdot)$ we denote a message authentication tag generation with key K using a provable computationally secure message authentication code (MAC). Then our modification is as shown in Fig. 2.

6 Conclusion

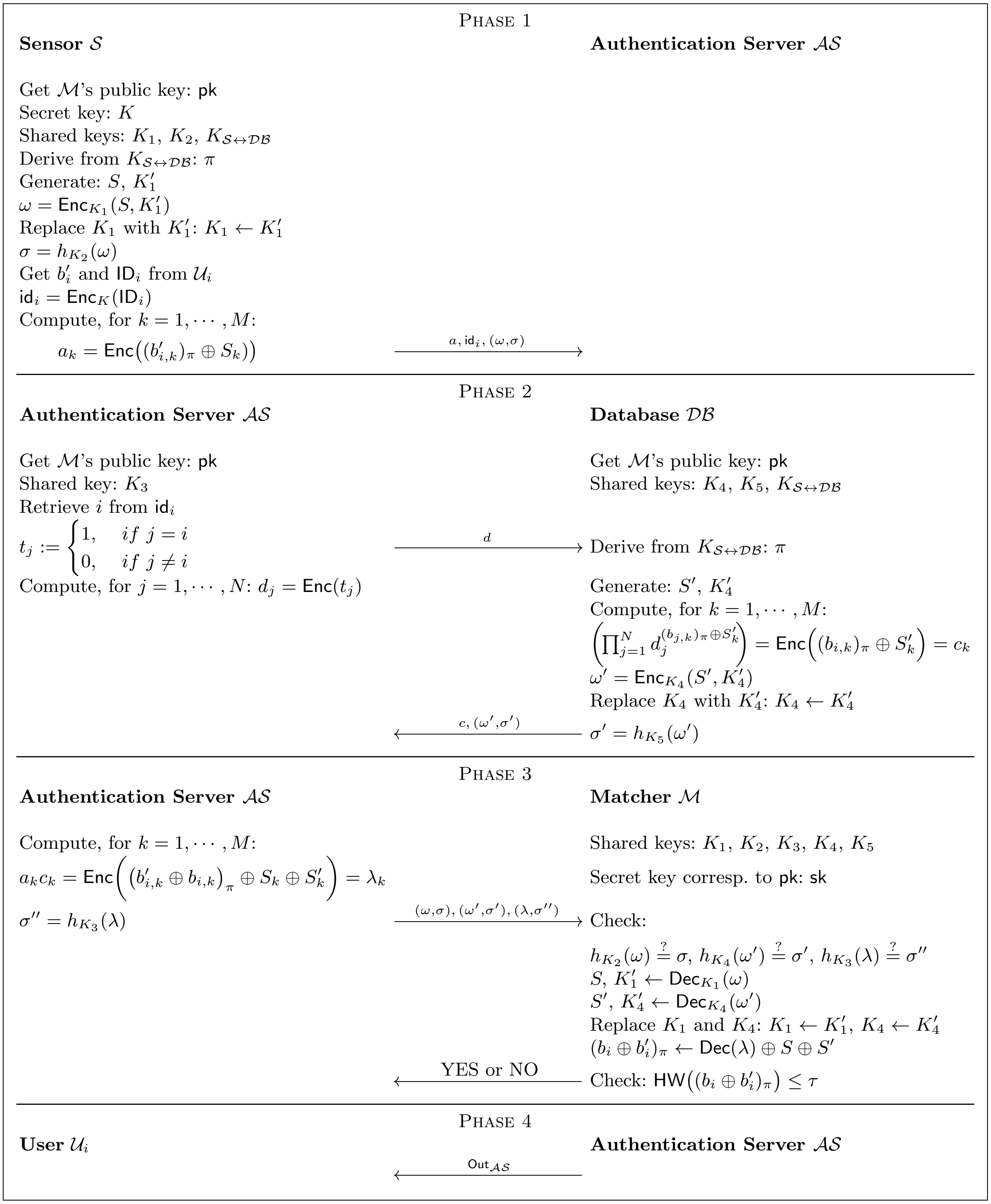
We investigated a *privacy-preserving biometric authentication* protocol proposed that uses the Goldwasser-Micali cryptosystem. We presented a simple attack algorithm that can be employed to mount a number of attacks on the system to either obtain the reference biometric template (b_i) or the identity (ID_i) of a user associated with a biometric template (b_i). Furthermore, we propose an improved version of the protocol that combats the described attacks.

7 Acknowledgements

We would like to thank Serge Vaudenay and Julien Bringer for their useful comments and feedback.

References

1. M. Barbosa, T. Brouard, S. Cauchie, and S. M. Sousa. Secure Biometric Authentication with Improved Accuracy. In *Proceedings of the 13th Australasian conference on Information Security and Privacy*, ACISP '08, pages 21–36, Berlin, Heidelberg, 2008. Springer-Verlag.
2. J. Bolling. A window to your health. *Jacksonville Medicine, Special Issue: Retinal Diseases*, 51, 2000.
3. J. Bringer, H. Chabanne, M. Izabachène, D. Pointcheval, Q. Tang, and S. Zimmer. An application of the Goldwasser-Micali cryptosystem to biometric authentication. In *Proceedings of the 12th Australasian conference on Information security and privacy*, ACISP'07, pages 96–106, Berlin, Heidelberg, 2007. Springer-Verlag.
4. H. Gilbert, M. J. B. Robshaw, and H. Sibert. Active attack against HB+: a provably secure lightweight authentication protocol. *Electronic Letters*, 41:1169–1170, October 2005.
5. S. Goldwasser and S. Micali. Probabilistic encryption & how to play mental poker keeping secret all partial information. In *Proceedings of the fourteenth annual ACM symposium on Theory of computing*, STOC '82, pages 365–377, New York, NY, USA, 1982. ACM.
6. J. Hermans, A. Pashalidis, F. Vercauteren, and B. Preneel. A new RFID privacy model. In *Proceedings of the 16th European conference on Research in computer security*, ESORICS'11, pages 568–587, Berlin, Heidelberg, 2011. Springer-Verlag.
7. A. Juels and S. A. Weis. Authenticating pervasive devices with human protocols. In *Proceedings of the 25th annual international conference on Advances in Cryptology*, CRYPTO'05, pages 293–308, Berlin, Heidelberg, 2005. Springer-Verlag.
8. K. Ouafi and S. Vaudenay. Strong Privacy for RFID Systems from Plaintext-Aware Encryption. In J. Pieprzyk, A.-R. Sadeghi, and M. Manulis, editors, *Cryptology and Network Security*, volume 7712 of *Lecture Notes in Computer Science*, pages 247–262. Springer Berlin Heidelberg, 2012.
9. L. Penrose. Dermatoglyphic topology. *Nature*, 205:544–546, February 1965.
10. K. Simoens, J. Bringer, H. Chabanne, and S. Seys. A framework for analyzing template security and privacy in biometric authentication systems. *IEEE Transactions on Information Forensics and Security*, 7(2):833–841, 2012.
11. C. Soutar, D. Roberge, A., R. Gilroy, and B. Kumar. Biometric encryption using image processing. In *Proceedings of SPIE*, volume 3314 of *Optical Security and Counterfeit Deterrence Techniques II*, pages 178–188, 1998.
12. S. Vaudenay. On Privacy Models for RFID. In K. Kurosawa, editor, *Proceedings of the 13th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT 2007)*, volume 4833 of *Lecture Notes in Computer Science*, pages 68–87. Springer, 2007.



Phase 1

Sensor $\mathcal{S}$ | **Authentication Server $\mathcal{AS}$**

Get $\mathcal{M}$'s public key: pk
Secret key: K
Shared keys: K_1, K_2, $K_{\mathcal{S}\leftrightarrow\mathcal{DB}}$
Derive from $K_{\mathcal{S}\leftrightarrow\mathcal{DB}}$: π
Generate: S, K_1'
$\omega = \mathsf{Enc}_{K_1}(S, K_1')$
Replace K_1 with K_1': $K_1 \leftarrow K_1'$
$\sigma = h_{K_2}(\omega)$
Get b_i' and ID_i from $\mathcal{U}_i$
$\mathsf{id}_i = \mathsf{Enc}_K(\mathsf{ID}_i)$
Compute, for $k = 1, \cdots, M$:

$$a_k = \mathsf{Enc}\big((b_{i,k}')_\pi \oplus S_k\big) \xrightarrow{a,\, \mathsf{id}_i,\, (\omega,\sigma)}$$

Phase 2

Authentication Server $\mathcal{AS}$ | **Database $\mathcal{DB}$**

Get $\mathcal{M}$'s public key: pk
Shared key: K_3
Retrieve i from id_i

$$t_j := \begin{cases} 1, & if\ j = i \\ 0, & if\ j \neq i \end{cases}$$

Compute, for $j = 1, \cdots, N$: $d_j = \mathsf{Enc}(t_j)$

Get $\mathcal{M}$'s public key: pk
Shared keys: K_4, K_5, $K_{\mathcal{S}\leftrightarrow\mathcal{DB}}$

$\xrightarrow{d}$ Derive from $K_{\mathcal{S}\leftrightarrow\mathcal{DB}}$: π

Generate: S', K_4'
Compute, for $k = 1, \cdots, M$:

$$\left(\prod_{j=1}^{N} d_j^{(b_{j,k})_\pi \oplus S_k'}\right) = \mathsf{Enc}\Big((b_{i,k})_\pi \oplus S_k'\Big) = c_k$$

$\omega' = \mathsf{Enc}_{K_4}(S', K_4')$
Replace K_4 with K_4': $K_4 \leftarrow K_4'$

$\xleftarrow{c,\, (\omega',\sigma')}$ $\sigma' = h_{K_5}(\omega')$

Phase 3

Authentication Server $\mathcal{AS}$ | **Matcher $\mathcal{M}$**

Compute, for $k = 1, \cdots, M$:

$$a_k c_k = \mathsf{Enc}\left(\left(b_{i,k}' \oplus b_{i,k}\right)_\pi \oplus S_k \oplus S_k'\right) = \lambda_k$$

$\sigma'' = h_{K_3}(\lambda)$ $\xrightarrow{(\omega,\sigma),\, (\omega',\sigma'),\, (\lambda,\sigma'')}$ Check:

Shared keys: K_1, K_2, K_3, K_4, K_5
Secret key corresp. to pk: sk

$h_{K_2}(\omega) \overset{?}{=} \sigma$, $h_{K_4}(\omega') \overset{?}{=} \sigma'$, $h_{K_3}(\lambda) \overset{?}{=} \sigma''$
S, $K_1' \leftarrow \mathsf{Dec}_{K_1}(\omega)$
S', $K_4' \leftarrow \mathsf{Dec}_{K_4}(\omega')$
Replace K_1 and K_4: $K_1 \leftarrow K_1'$, $K_4 \leftarrow K_4'$
$(b_i \oplus b_i')_\pi \leftarrow \mathsf{Dec}(\lambda) \oplus S \oplus S'$

$\xleftarrow{\text{YES or NO}}$ Check: $\mathsf{HW}\big((b_i \oplus b_i')_\pi\big) \leq \tau$

Phase 4

User $\mathcal{U}_i$ | **Authentication Server $\mathcal{AS}$**

$\xleftarrow{\mathsf{Out}_{\mathcal{AS}}}$

Fig. 2. The modified Bringer *et al.* [3] protocol.