Open access • Book Chapter • DOI:10.1007/978-3-319-48965-0_20

An Efficient Entity Authentication Protocol with Enhanced Security and Privacy Properties — Source link

Aysajan Abidin, Enrique Argones Rúa, Bart Preneel

Institutions: Katholieke Universiteit Leuven

Published on: 14 Nov 2016 - Cryptology and Network Security

Topics: Authentication protocol, Challenge-Handshake Authentication Protocol, Lightweight Extensible Authentication Protocol, Otway–Rees protocol and Authentication

An efficient entity authentication protocol with enhanced security and privacy properties

Aysajan Abidin, Enrique Argones Rúa, Bart Preneel

KU Leuven, ESAT/COSIC, Belgium and iMinds, Belgium
Firstname.Lastname@esat.kuleuven.be

Abstract. User authentication based on biometrics is getting an increasing attention. However, privacy concerns for biometric data have impeded the adoption of cloud-based services for biometric authentication. This paper proposes an efficient distributed two-factor authentication protocol that is privacy-preserving even in the presence of colluding internal adversaries. One of the authentication factors in our protocol is biometrics, and the other factor can be either knowledge-based or possession-based. The actors involved in our protocol are users, user/client devices with biometric sensors, service provider, and cloud for storing protected biometric templates. Contrary to the existing biometric authentication protocols that offer security only in the *honest-but-curious* adversarial model, our protocol provides enhanced security and privacy properties in the *active* (or *malicious*) adversarial model. Specifically, our protocol offers identity privacy, unlinkability, and user data (i.e., the biometric template data and the second factor) privacy against compromised cloud storage service, and preserves the privacy of the user data even if the cloud storage service colludes with the service provider. Moreover, our protocol only employs lightweight schemes and thus is efficient. The distributed model combined with the security and privacy properties of our protocol paves the way towards a new cloud-based business model for privacy-preserving authentication.

Key words: Biometrics, security, privacy, privacy-preserving authentication.

1 Introduction

As biometric authentication is becoming more popular and ubiquitous, protecting and ensuring the privacy of biometric templates is of utmost importance. Biometrics poses a serious threat to user privacy. Not only does it reveal sensitive information about users such as medical condition, race and ethnicity, but it can also be used for mass surveillance. A number of privacy-preserving authentication protocols involving biometrics have been proposed over the last decade. Most of them, however, are designed to be secure in the *honest-but-curious* (HBC) adversarial model. In this work, we go beyond the HBC model and propose an efficient privacy-preserving biometric authentication protocol with enhanced security and privacy properties in the *malicious* adversary model. Our protocol also utilises an additional short secret, e.g., a password, as a second factor. Privacy of users is protected from two different threats: disclosure of privacy-sensitive data (i.e., biometric templates and other secrets) and disclosure of behavioural information (i.e., user's identity when using an online service) to *malicious* internal adversaries. We employ a distributed model for the protocol participants and categorise them as the user, the client device (i.e., sensor), the service provider, and the cloud storage.

A brief overview of the protocol. It consists of a set of N ($\geqslant 1$) users U, a set of N sensors S (one for each user), a service provider SP, and a cloud storage provider which we just call database DB throughout the paper. During the enrolment phase, the sensor S_i obtains from a user U_i a biometric reference template b_i, a password pw_i (for simplicity, we regard the second factor as a password, but any knowledge-based or possession-based factor could be used instead) and an identity ID_i. It then derives a random bitstring r_i of the same length as b_i from pw_i and ID_i using a key derivation function KDF [1] (i.e., $r_i \leftarrow \mathsf{KDF}(\mathsf{pw}_i, \mathsf{ID}_i)$), computes $b_i \oplus r_i$, and sends $(\mathsf{ID}_i, b_i \oplus r_i)$ to the service provider SP. Since we are using a combination (i.e., XOR) of factors, this has to be taken into consideration when choosing security parameters for these factors (cf. Sect. 3.1). SP then maps the ID_i to an index i (i.e., $i \leftarrow \mathsf{ID}_i$) using a procedure known only to itself and forwards $(i, b_i \oplus r_i)$ to the database DB for storage. SP itself stores (i, ID_i).

During the authentication phase, a user U_i authenticates himself to the service provider as follows. The sensor S_i obtains a fresh biometric template b'_i, the user password pw_i and an identity ID_i from the user. S_i then generates r_i using the same procedure as in the enrolment phase, computes $b'_i \oplus r_i$, and sends $(\mathsf{ID}_i, b'_i \oplus r_i)$ to SP. The service provider SP retrieves i corresponding to ID_i from its own storage and retrieves $b_i \oplus r_i$ from DB by employing a private information retrieval PIR scheme. This scheme allows SP to retrieve $b_i \oplus r_i$ from DB without revealing to DB the value of the retrieved information (under information-theoretic or computational security assumptions, cf. Sect. 3). SP then XORs $b_i \oplus r_i$ and $b'_i \oplus r_i$ to get $b_i \oplus b'_i$, and grants the user U_i access (or simply authenticates the user) if the Hamming weight $\mathsf{HW}(b_i \oplus b'_i) \leqslant \tau$, where τ is a predefined authentication threshold. Note that $\mathsf{HW}(b_i \oplus b'_i) = \mathsf{HD}(b_i, b'_i)$, where HD is the Hamming distance.

This two-factor authentication protocol employs a combination of a private information retrieval scheme and a key derivation function to achieve strong privacy. It preserves the privacy of the biometric templates, password and password-derived key against malicious and colluding service provider and database; and also offers identity privacy and unlinkability against malicious database, due to the database anonymisation and the use of a PIR during authentication.

Applications. Cloud computing provides an interesting set of advantages, such as increased availability and flexibility, reduced risks related to data losses, and reduced costs in terms of technology infrastructure. Due to privacy concerns, however, the adoption of cloud-based services for biometric authentication has been delayed. Our new protocol enables a new secure and privacy-preserving authentication cloud service business model. Services provided by the Database actor in our protocol can be securely transferred to a cloud-based service. This cloud-based Protected Biometrics Database service can scalably provide secure and private storage and retrieval of user's authentication data to different Service Providers.

Remarks. Some design decisions in our system must be explained in more depth, to make clear that they are realistic and fully justified.

First, we choose to work with biometric binary templates instead of other alternative representations based on integers or real numbers. This is justified by the existence of many biometrics based on binary templates, e.g. iris patterns are represented by IrisCodes [2]; or where a binary representation can be derived, e.g. even behavioural biometrics such as online signatures can be represented in binary templates [3]. Furthermore, binary templates can be compared using the Hamming distance, which is very convenient for simple and well-known homomorphic encryption schemes, thus avoiding the need for a specific design of a new cryptosystem. Using other representations requires using much more complex crypto schemes, e.g. computing the Euclidean distance requires a fully homomorphic encryption or coupling additive homomorphic encryption scheme with an oblivious transfer or garbled circuits.

Second, we combine several factors. This may influence usability, since the processes for using the system in both enrollment and verification will take longer and the users will experience usability issues related to both authentication factors. However, the combination of several authentication factors, as demonstrated in this paper, minimizes the risks associated with the use of each of the authentication factors. Specifically, the security of the system is significantly increased, making attacks much more difficult to the typical adversaries, and the privacy concerns posed by the use of biometrics are minimized by using the additional authentication factor for binding the binary biometric information. The proposed solution is simple yet effective.

Related work. Privacy-preserving biometric authentication has attracted a considerable amount of research over the last decade. Many of the existing privacy-preserving biometric authentication protocols are based on secure multi-party computation techniques including oblivious transfer [4, 5] and homomorphic encryption [6, 7], as well as on private information retrieval [8, 9]. For example, Bringer *et al.* [10] proposed a protocol using the Goldwasser-Micali cryptosystem [7]. This protocol by Bringer *et al.* and the subsequent protocols by Barbosa *et al.* [11] and Stoianov [12] all use a distributed entity model. However, all of the these protocols are designed to achieve security in the HBC model, and their security is also critcized [11, 13–16]. To the best of our knowledge, most (if not all) of the protocols using biometrics as single authentication factor presented in the literature are at best secure against HBC adversaries only. Recently, Abidin *et al.* [17] describe a simple attack on the Bringer *et al.* [10] protocol and proposes an improvement to achieve security against malicious but not colluding insider adversaries, utilizing additional secret keys. As in the original Bringer *et al.* protocol, Abidin *et al.* protocol also stores the reference biometric templates in the clear.

There have been other works combining biometrics with other authentication factors, such as knowledge-based (e.g., passwords) and/or possession-based (e.g., tokens). This multi-factor approach involving biometrics has been a popular approach to remote biometric authentication [18]. For example, in [19] a scheme combining biometrics with a password and a smart card was proposed by Lee *et al.*. Weaknesses of this scheme were identified subsequently in [20], where the authors also propose a flexible remote authentication scheme based on fingerprints and ElGammal cryptosystem. However, this latter scheme was vulnerable to, among others, spoofing attacks as identified by Khan and Zhang [21]. More efficient schemes were also proposed in [22, 23, 18] in the past couple of years, although some of them turned out to have security weaknesses [24]. A common feature among these schemes is that they use smart cards to store authentication information. Hence a drawback of these schemes is that if the smart card is stolen or lost, then either the security is at risk or the user can no longer authenticate himself.

Outline. After giving the necessary background material and our threat model in Sect. 2, we give a detailed presentation of our protocol, paying particular attention to the key derivation function and the private information retrieval scheme in Sect. 3. Next, we analyse its privacy and security in Sect. 4. Finally, Sect. 5 summarises the paper.

2 Background and threat model

This section presents the necessary background material and the security requirements for the cryptographic primitives used in our protocol.

Definition 1. *A function* $\mathsf{negl} : \mathbb{N} \mapsto [0, 1]$ *is said to be negligible if for all positive polynomials* poly *and all sufficiently large* $\lambda \in \mathbb{N}$, *we have* $\mathsf{negl}(\lambda) < 1/\mathsf{poly}(\lambda)$.

2.1 Security and privacy definitions

Definition 2. *Let* Π *be a two-factor authentication protocol. Then* Π *is secure if no probabilistic polynomial time (*PPT*) adversary* $\mathcal{A}$ *can successfully authenticate itself to the verifier as the legitimate user it impersonates, even when given all protocol transcripts and all inputs of the verifier and all provers (i.e., users) with the exception of at least one authentication factor of the user it tries to impersonate.*

Regarding privacy, we consider unlinkability, identity privacy and user data privacy. Let Π be as before in all of the following definitions.
Unlinkability. Intuitively, if the adversary cannot distinguish a user who is authenticating himself from a user who is not, then unlinkability holds. Therefore, we define unlinkability as follows.

Definition 3. *Suppose that any two distinct users* U_{i_0} *and* U_{i_1}, *where* $i_0, i_1 \geqslant 1$, *are given and that* U_{i_β}, $\beta \in \{0, 1\}$, *makes an authentication attempt. Then,* Π *has unlinkability, if any* PPT *adversary* $\mathcal{A}$ *cannot guess* β, *except with a negligible advantage. Here, the adversary's advantage is defined as* $\left|\Pr\{\beta = \beta'\} - 1/2\right|$, *where* β' *is the adversary's guess.*

Identity privacy. If the adversary cannot tell to which ID a given authentication credential belongs, then we say that the identity privacy is preserved. Formally, this is defined as follows.

Definition 4. *Suppose that any identity ID_i and two credentials $c_{i_0} = b_{i_0} \oplus r_{i_0}$ and $c_{i_1} = b_{i_1} \oplus r_{i_1}$, where $i_0, i_1 \geqslant 1$ and c_{i_β}, $\beta \in \{0, 1\}$, belongs to ID_i, are given. Then, Π preserves the identity privacy, if any PPT adversary $\mathcal{A}$ cannot guess β, except with a negligible advantage. Here again, the adversary's advantage is defined as $\big|\Pr\{\beta = \beta'\} - 1/2\big|$, where β' is the adversary's guess.*

User data privacy. If the adversary cannot learn anything about the sensitive user data (i.e., biometric data and the second authentication factor), then we say that the user data privacy is preserved.

Definition 5. *We say that Π preserves the privacy of the user data, i.e., the biometric templates (both fresh and reference), the password and/or the password-derived key, if no PPT adversary $\mathcal{A}$ can gain more information on the user data than what is allowed by the protocol transcripts, except with a negligible probability.*

2.2 Key derivation function

A key derivation function (KDF) is a (deterministic) function that can be used to derive keys for cryptographic applications using a secret input data, such as passwords. We require that the KDF satisfies the following security definition [1].

Definition 6. *A key derivation function KDF is said to be secure with respect to a source of input with sufficient min-entropy γ if no probabilistic polynomial time (PPT) attacker $\mathcal{A}$ can distinguish its output from a random output of equal length, except with a negligible probability $\mathsf{negl}(\gamma)$.*

2.3 Private information retrieval

A PIR scheme allows a user to retrieve a value from a database without revealing to the database which value is retrieved. For example, using a PIR scheme a user can retrieve the i-th bit (or the i-th block) from a database of N-bits (or a database of N blocks) without revealing the value of i to the database. We require that the PIR scheme satisfies the following definition.

Definition 7. *Suppose that the database contains an N data blocks with blocklength ℓ bits each, with both $N, \ell \geqslant 1$ (i.e., the database contains $x_1 x_2 \cdots x_N$, where the length of x_i is ℓ, for $i = 1, \cdots, N$.) And let PIR be the private information retrieval scheme employed to retrieve the i-th block from the database (i.e., $x_i \leftarrow \mathsf{PIR}(i)$). Then the database should* *not have any information about the value of i and x_i. If the database is assumed to be computationally unbounded, the* PIR *is called information-theoretic* PIR*; otherwise, it is called computational* PIR*.*

2.4 Threat model

In the typical security analysis, adversaries are divided into two main categories: (i) *honest-but-curious* (HBC) adversaries, and (ii) *malicious* adversaries. In the HBC adversarial model, corrupted parties follow the protocol specification. However, the adversary may obtain the internal state of all corrupted parties (i.e., transcript of all received messages) and may attempt to use this information to recover sensitive data (e.g., biometric templates) that should remain private. In the malicious adversarial model, the corrupted parties may arbitrarily deviate from the protocol specification in order to break the security and privacy of the protected data. Since external adversaries cannot obtain more information than the internal ones, we consider exclusively *malicious* internal adversaries that may arbitrarily deviate from the protocol specification.

To be privacy-preserving, a biometric authentication protocol should satisfy not only the security requirement (cf. Def. 2), but also the following privacy requirements:

1. ***Biometric reference privacy:*** An adversary $\mathcal{A}$ should not be able to recover the stored reference biometric template (cf. Def. 5).
2. ***Biometric sample privacy:*** $\mathcal{A}$ should not be able to recover the fresh biometric sample (cf. Def. 5).
3. ***Password privacy:*** $\mathcal{A}$ should not be able to recover the password or the key derived from the password (cf. Def. 5).
4. ***Identity privacy:*** $\mathcal{A}$ should not be able to link a database entry to a user identity ID. Note that the protocol does not require ID to be personally identifiable information, and so this privacy requirement only concerns whether a database entry is associated to a specific user ID employed in the protocol (cf. Def. 4).
5. ***Unlinkability:*** $\mathcal{A}$ should not be able to link an authentication attempt to a user (cf. Def. 3).

In this paper, we only consider adversaries that attempt to violate these privacy requirements and skip denial-of-service type of attacks.

Depending on the attack scenario (i.e., depending on which protocol entity is compromised or malicious), the privacy requirements change accordingly. For instance, the service provider SP always knows which user is authenticating himself so the unlinkability and identity privacy are not relevant, although the latter can be achieved using anonymous IDs. Therefore, whenever the SP is compromised, either colluding or not colluding with the DB, we only focus on the biometric samples and the password (or password-derived key) secrecy. On the other hand, if the attacker is the database DB all of the requirements must hold.

The Sensor is trusted, i.e. it does not deviate the protocol, it does not store ID or biometric samples, and the information it handles during the enrollment and verification phases is discarded and only accessible to the legitimate application at run time; and we are not considering the case where it is compromised, e.g. infected by malicious software. We assume that each user has a client device (e.g., a smartphone) with a biometric sensor. It is quite common nowadays that people use their smartphones to do even bank transactions. This does not make the user devices trustworthy, but if the users cannot trust the devices used to log in, a secure access to their remote services cannot be accomplished.

A further assumption we make is that the communication among the protocol entities takes place over a secure channel. This means that an adversary cannot intercept or modify a message in transit. Lastly, we require that before any user authenticates himself to the service provider, the service provider authenticates itself to the client device (i.e., the sensor). This can be achieved by using secure transmission protocols, e.g. TLS or IPsec. Hence, at the conclusion of the protocol there should be a mutual entity authentication, where the service provider is authenticated first and then the user authenticates himself to the service provider. This is to preclude phishing attacks, i.e. to ensure that the user does not blindly give away his identity and authentication data to attackers. The authentication mechanism for the server and the way it is coupled to the user authentication is left outside the scope of this paper.

3 The protocol

The protocol comprises a set S of N sensors, one for each user in a set U of N users, a service provider SP, and a database DB. Each user is assumed to have a client device (e.g., a smartphone), which has a biometric sensor.

Enrolment. The enrolment works as follows. The sensor S_i prompts the user U_i, for $i = 1, \cdots, N$, for his biometrics and password pw_i, and outputs $b_i \oplus r_i$, where b_i is a (binary) reference biometric template of bitlength ℓ extracted by the sensor from the user provided biometrics and $r_i \leftarrow \mathsf{KDF}(\mathsf{pw}_i, \mathsf{ID}_i)$ is also of bitlength ℓ. Then, S_i sends $(\mathsf{ID}_i,\, b_i \oplus r_i)$ to the service provider SP, that first maps the ID_i to a unique index i and locally stores $(i,\, \mathsf{ID}_i)$, and forwards $(i,\, b_i \oplus r_i)$ to DB for storage.

The service provider does not need to store (i, ID_i). Instead, what SP needs is a deterministic one-to-one map to map ID_i to an index i. For the sake of simplicity, however, we assume that SP locally stores the pair (i, ID_i), and that during authentication it just retrieves the index i corresponding to a received ID_i from its local storage.

User authentication. A user U_i authenticates himself to the service provider as follows. After authenticating the service provider, the sensor S_i prompts the user for his data: fresh biometrics, password and identity. Then, the sensor extracts a fresh biometric template b'_i from the user provided biometrics, receives the user password pw_i and identity ID_i from the user U_i. Subsequently, S_i derives r_i using the key derivation function KDF with pw_i and ID_i as input, computes $b'_i \oplus r_i$, and sends $(\mathsf{ID}_i, b'_i \oplus r_i)$ to the service provider SP. SP first obtains i corresponding to ID_i from its own local storage and retrieves $b_i \oplus r_i$ from DB by employing a private information retrieval PIR scheme (see Sect. 3.1 on PIR for details). SP then XORs $b_i \oplus r_i$ and $b'_i \oplus r_i$ to obtain $b_i \oplus b'_i$. Finally, the user is authenticated if the Hamming weight $\mathsf{HW}(b_i \oplus b'_i) \leqslant \tau$; rejected, otherwise.

Note that the sensors S_i, $i = 1, \cdots, N$, do not store any user information (i.e., biometric template data and/or password), thus user's data privacy is still preserved if his terminal is stolen or lost. When a user U_i presents his biometrics to the sensor S_i, it only outputs the XOR of the extracted biometric template with the derived key r_i, i.e., $b'_i \oplus r_i$ or $b_i \oplus r_i$ depending on the protocol phase, and never outputs the biometric template data or the password-derived key r_i in the clear or stores them. Also, the r_i's are generated at run time using the password pw_i and user ID_i as input to a KDF, and r_i and pw_i are erased from memory immediately after use.

To highlight the feasibility of our protocol, below we elaborate further on the KDF and the PIR scheme that can be employed in our protocol. However, since we would like to keep it as generic as possible, we leave the choice for specific KDF and PIR schemes for the users of our protocol.

3.1 KDF

In our protocol, both the reference b_i and fresh b'_i biometric templates are bound (i.e., XORed) with keys r_i generated from the second authentication factor (e.g., password) using a KDF. KDF is a useful tool in cryptography and often used in diverse applications to derive cryptographic keys from a secret input. According to PKCS # 5 [25], for a password-based KDF, it is recommended to salt the password in order to prevent dictionary attacks and to compute the hash many times to slow down the KDF process, which is also known as *key-stretching* [26]. If password is used as the second factor in our protocol, then the salt needs to be stored in the user device. We refer the interested reader to Yao *et al.* [27] for a formal treatment of password-based KDF, and to Krawczyk [1] for a more general treatment and rigorous security definitions of KDF. What is important to note when choosing a specific KDF for our protocol is that the chosen KDF must be secure according to our Def. 6.

Regarding the security requirements on the inputs to the KDF, we note that the password should have at least the same min-entropy as the one required for the output. Since only the XOR of the KDF output with the biometric template is stored, the security requirement should be referred to this combination (i.e., $r_i \oplus b_i$), whose min-entropy is greater than or equal to $\max\{H_\infty(r_i), H_\infty(b_i)\}$, where H_∞ stands for min-entropy. Therefore, as long as one of the factors provides sufficient min-entropy, the entropy requirement on the other can be relaxed if the security is our only concern. However, the min-entropy of the second authentication factor impacts privacy. The min-entropy of the second factor should be greater than or equal to the entropy of the biometric template for avoiding biometric information leakage to internal adversaries.

3.2 PIR

As mentioned briefly in Sect. 2, PIR schemes allow for the retrieval of the content of a database entry, say the i-th bit of an N-bit database, without revealing to the database which content or entry is retrieved (i.e., the value of i in the example). Chor *et al.* [8] were the first to introduce the notion of PIR, and they studied information-theoretically secure PIRs in the case of single database or multiple non-communicating databases. Since then, there has been a substantial amount of work on PIR; we present here a quick review of the work relevant to our protocol.

Recall that we assume that there is a single database. In practice, however, one can use multiple databases (e.g., multiple cloud storage providers) storing the same information. This is more robust, because even if some databases are down, e.g., due to power outage, users can still authenticate themselves to the service provider. So, we divide our discussions on PIRinto single database PIR and multiple database PIR.

Single DB PIR. Since Kushilevitz and Ostrovsky [28] proposed the first single DB PIR scheme, the field has evolved and important connections between single DB PIR and other cryptographic primitives, such as oblivious transfer and collision resistant hashing, have been identified [29]. Ostrovsky and Skeith give a nice survey on single DB PIR schemes in [29]. Here we describe a simple scheme that appeared in [10],which utilises the Goldwasser-Micali cryptosystem [7], a bit-wise encryption scheme with an homomorphic property: $\mathsf{Enc}(m)\mathsf{Enc}(m') = \mathsf{Enc}(m \oplus m')$, where m and m' are two message bits. Suppose that the SP wants to retrieve the i-th user's data item $b_i \oplus r_i$ from the DB. Also suppose that SP generates a private and public key pair (sk, pk) for the Goldwasser-Micali encryption schemes, and gives the public key pk to the database and keeps the secret key sk to itself. Assume from now on that the content of the DB is an $N \times \ell$ binary matrix A. Then, SP forms $\mathsf{PIR}(i)$ as follows: for $j = 1, \cdots, N$, $s_j = 1$, if $j = i$, 0 otherwise. It sends $\mathsf{Enc}(s) = \big(\mathsf{Enc}(s_1), \cdots, \mathsf{Enc}(s_N)\big)$ to the DB, which computes, for $n = 1, \cdots, \ell$, $C_{i,n} := \big(\prod_{j=1}^{N} \mathsf{Enc}(s_j)^{A_{j,n}} \mathsf{Enc}(0)\big) = \mathsf{Enc}\big(A_{i,n}\big) = \mathsf{Enc}\big(b_{i,n} \oplus r_{i,n}\big)$, and returns $C_i = (C_{i,1}, \cdots, C_{i,\ell})$ to SP. Note that $\mathsf{Enc}(0)$ is used to randomise the response in order to resist an attack similar to the one described by Barbosa *et al.* [11]. Finally, SP decrypts C_i to obtain $b_i \oplus r_i$. This scheme has a communication complexity of $\mathcal{O}(Nc + \ell c)$, where c is the ciphertext length, which needs to be at least 2048 bits for 112-bit security. Furthermore, this PIR scheme is computationally secure according to our Def. 7, if the Goldwasser-Micali encryption is IND-CPA secure [10].

Multiple DB PIR. When there are k $(\geqslant 1)$ copies of $(i, b_i \oplus r_i)$, for $i = 1, \cdots, N$, stored in k DBs, we can use the following information-theoretic PIR scheme in our protocol. Suppose that there are 2 DBs and that the SP wants to retrieve i entry, which is $b_i \oplus r_i$, from the DBs. Then, the PIR scheme works as follows:

- The SP prepares the queries as follows:
  - generate at random a bitstring s of length N.
  - flip the i-th bit of s; let us denote the resulting bitstring by s'.
  - send s to DB1, and s' to DB2.
- DB1 returns $t = sA \mod 2$, where s is used as a row vector, to SP.
- DB2 returns $t' = s'A \mod 2$, where s' is used as a row vector, to SP.
- Finally, SP computes $t \oplus t' = sA \oplus s'A = (s \oplus s')A = b_i \oplus r_i$. Note that $s \oplus s'$ is all 0s except at the i-th position, where it has a 1.

Obviously, this 2-DB PIR scheme has a communication complexity of $\mathcal{O}(N + \ell)$. And the computation performed by the DBs is just the XOR of the rows (of A) corresponding to the components of s (or s') that are 1. We note that this scheme, or for that matter most k-DB PIR schemes, assumes that the DBs are trusted not to collude with each other; otherwise, the DBs can learn the value of i. There are, however, also k-DB PIR schemes that remain secure even if all databases collude with each other [30]. We refer to the excellent survey by Gasarch [31] for more on multiple DB PIR.

4 Security and privacy analysis

We assume that the protocol setup and enrolment phases are done securely and all involved entities behaved honestly in these phases. Therefore, we focus on the authentication phase in our analysis. We distinguish the following attack scenarios from each other.

1. *Attacker = The service provider SP*: Its objective is to learn the user biometric template or the user password. It has access to $b'_i \oplus r_i$ and $b_i \oplus r_i$, and $b'_i \oplus b_i$. The identity privacy and unlinkability, however, are not relevant if SP is compromised, as it knows the user IDs.
2. *Attacker = The database DB*: Its objectives are to learn (a) user identity, (b) biometric templates, (c) passwords or password-derived keys, and (d) link different authentication attempts. It knows only $b_i \oplus r_i$, but it does not know to which user it belongs, since the database is anonymised. Also, since a secure PIR is employed during authentication, the identity privacy and unlinkability requirements are also satisfied. So all of the privacy requirements are satisfied in this case.
3. *Attacker = SP+DB*: Their objective is to learn the user biometric template or the user password. In this case, they know $b_i \oplus r_i$, $b'_i \oplus r_i$, and $b_i \oplus b'_i$, as in the case of the attacker being SP. Therefore, against the collusion between SP and DB, the biometric template (both reference and sample) privacy and the password (or the password-derived key r_i) privacy are preserved.
4. *Attacker = The sensor S_i*: we only consider the case where S_i is used by a malicious user to impersonate its legitimate owner. Since the sensor does not store any information about its legitimate user's biometrics and password, the attacker cannot learn anything or impersonate the user.

We now state the security and privacy properties of our protocol. The proofs are presented in Appendix A.

Theorem 1. *Our proposed protocol is secure according to our Def. 2, if the employed KDF and PIR are secure according to Def. 6 and 7, respectively, and $i \leftarrow \mathsf{ID}_i$ procedure is known only to the SP.*

Unlinkability against malicious DB. Recall that if the adversary cannot distinguish a user who is authenticating himself from a user who is not, then unlinkability holds. Therefore, we state the unlinkability result against malicious DB as follows.

Theorem 2. *Our proposed protocol has unlinkability against* malicious *DB according to Def. 3, if the employed PIR is secure according to Def. 7.*

Identity privacy against malicious DB. If the DB cannot tell to which ID a database entry belongs, then we say that the identity privacy is preserved. This is summarised in the next theorem.

Theorem 3. *Our proposed protocol has identity privacy against* malicious *DB according to Def. 4, if the employed PIR is secure according to Def. 7 and $i \leftarrow \mathsf{ID}_i$ procedure is known only to the SP.*

User data privacy against malicious SP+DB. Our last result relates to the privacy of user data, i.e., the fresh and reference biometric templates, password and password-derived key. Note that when we say that the password has sufficient min-entropy, the word "password" is used just as a reference to the second authentication factor which is given as an input to the KDF. The following theorem states that as long as the KDF is secure, the privacy of the user data is preserved against malicious and colluding SP and DB.

Theorem 4. *Our protocol preserves the privacy of the user data (i.e., the fresh and reference biometric templates, password and password-derived key) against* malicious *SP+DB according to Def. 5, if the employed KDF is secure according to Def. 6 and the password has sufficient min-entropy.*

5 Conclusions

In this paper, we proposed a two-factor privacy-preserving authentication protocol that is secure against malicious and possibly colluding adversaries. The second factor (e.g., password) adds another layer of security in that even if an attacker successfully forges a user biometrics (e.g., a fingerprint), he/she cannot impersonate the user without knowing the password. Furthermore, as our analysis shows, the privacy of the users' identities, their passwords and biometric template data is preserved even if the protocol actors are compromised. The protocol is efficient and employs a distributed model for the protocol actors and thus suitable for applications where users authenticate themselves to a service provider using their smart devices that have embedded biometric sensors and where part of the user data (i.e., the encrypted biometric reference template) are outsourced to cloud storage providers. Hence, our protocol paves the way towards a secure and privacy-preserving authentication cloud service business model. In this model, services provided by the Database in our protocol can be securely transferred to one or several cloud-based services. Such cloud storage services can provide a secure and private storage and retrieval of user's authentication data to different Service Providers.

Acknowledgements. This work was funded by the European Commission through the FP7 project "EKSISTENZ," with grant number: 607049.

References

1. Krawczyk, H.: Cryptographic extraction and key derivation: The HKDF scheme. In: Advances in Cryptology–CRYPTO 2010. Volume 6223 of LNCS. Springer (2010) 631–648
2. Daugman, J.: The importance of being random: statistical principles of iris recognition. Pattern recognition **36**(2) (2003) 279–291
3. Rua, E.A., Maiorana, E., Castro, J.L.A., Campisi, P.: Biometric template protection using universal background models: An application to online signature. IEEE Transactions on Information Forensics and Security **7**(1) (2012) 269–282
4. Rabin, M.O.: How to exchange secrets with oblivious transfer. IACR Cryptology ePrint Archive **2005** (2005) 187
5. Yao, A.C.C.: How to generate and exchange secrets. In: Foundations of Computer Science, 1986., 27th Annual Symposium on, IEEE (1986) 162–167
6. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: EUROCRYPT '99. Volume 1592 of LNCS. Springer (1999) 223–238
7. Goldwasser, S., Micali, S.: Probabilistic encryption & how to play mental poker keeping secret all partial information. In: STOC, ACM (1982) 365–377
8. Chor, B., Kushilevitz, E., Goldreich, O., Sudan, M.: Private information retrieval. Journal of the ACM **45**(6) (1998) 965–981
9. Ostrovsky, R., Willian E. Skeith, I.: A survey of single-database private information retrieval: techniques and applications. In: PKC. Volume 4450 of LNCS., Springer Verlag (2007) 393–411
10. Bringer, J., Chabanne, H., Izabachène, M., Pointcheval, D., Tang, Q., Zimmer, S.: An application of the Goldwasser-Micali cryptosystem to biometric authentication. In: ACISP 2007. Volume 4586 of LNCS., Springer (2007) 96–106
11. Barbosa, M., Brouard, T., Cauchie, S., de Sousa, S.M.: Secure biometric authentication with improved accuracy. In: ACISP. Volume 5107 of LNCS., Springer (2008) 21–36
12. Stoianov, A.: Cryptographically secure biometrics. SPIE 7667, Biometric Technology for Human Identification VII (2010) 76670C–12
13. Simoens *et al.*, K.: A framework for analyzing template security and privacy in biometric authentication systems. IEEE Transactions on Information Forensics and Security **7**(2) (2012) 833–841
14. Abidin, A., Mitrokotsa, A.: Security aspects of privacy-preserving biometric authentication based on ideal lattices and ring-lwe. In: Proceedings of the IEEE Workshop on Information Forensics and Security. (2014) 1653–1658
15. Abidin, A., Pagnin, E., Mitrokotsa, A.: Attacks on privacy-preserving biometric authentication. In: NordSec 2014. Volume 8788 of LNCS., Springer (2014) 293–294
16. Pagnin, E., Dimitrakakis, C., Abidin, A., Mitrokotsa, A.: On the leakage of information in biometric authentication. In: INDOCRYPT 2014. Volume 8885 of LNCS., Springer (2014) 265–280
17. Abidin, A., Matsuura, K., Mitrokotsa, A.: Security of a privacy-preserving biometric authentication protocol revisited. In: CANS 2014. Volume 8813 of LNCS., Springer (2014) 290–304
18. Syta, E., Wolinsky, D., Fischer, M., Silberschatz, A., Ford, B., Gallegos-Garcıa, G.: Efficient and privacy-preserving biometric authentication. Yale University Technical Report TR1469 (2012)
19. Lee, J., Ryu, S., Yoo, K.: Fingerprint-based remote user authentication scheme using smart cards. Electronics Letters **38**(12) (2002) 554–555

20. Lin, C.H., Lai, Y.Y.: A flexible biometrics remote user authentication scheme. Computer Standards & Interfaces **27**(1) (2004) 19–23
21. Khan, M.K., Zhang, J.: Improving the security of a flexible biometrics remote user authentication scheme. Computer Standards & Interfaces **29**(1) (2007) 82–85
22. Li, C.T., Hwang, M.S.: An efficient biometrics-based remote user authentication scheme using smart cards. Journal of Network and computer applications **33**(1) (2010) 1–5
23. Li, X., Niu, J.W., Ma, J., Wang, W.D., Liu, C.L.: Cryptanalysis and improvement of a biometrics-based remote user authentication scheme using smart cards. Journal of Network and Computer Applications **34**(1) (2011) 73–79
24. Li, X., Niu, J., Khan, M.K., Liao, J.: An enhanced smart card based remote user password authentication scheme. Journal of Network and Computer Applications **36**(5) (2013) 1365–1371
25. Kaliski, B.: PKCS #5: Password-based cryptography specification version 2.0. RFC 2898 (2000)
26. Kelsey, J., Schneier, B., Hall, C., Wagner, D.: Secure applications of low-entropy keys. In: ISW 1997. Volume 1396 of LNCS., Springer-Verlag (1998) 121–134
27. Yao, F.F., Yin, Y.L.: Design and analysis of password-based key derivation functions. In: Topics in Cryptology–CT-RSA 2005. Volume 3376 of LNCS., Springer (2005) 245–261
28. Kushilevitz, E., Ostrovsky, R.: Replication is not needed: single database, computationally-private information retrieval. In: FOCS, IEEE Computer Society (1997) 364–373
29. Ostrovsky, R., Skeith III, W.E.: A survey of single-database private information retrieval: Techniques and applications. In: PKC 2007. Volume 4450 of LNCS., Springer (2007) 393–411
30. Goldberg, I.: Improving the robustness of private information retrieval. In: IEEE SP'07, IEEE (2007) 131–148
31. Gasarch, W.: A survey on private information retrieval. Bulletin of the EATCS **82** (2004) 72–107

A Proofs

Proof (of Theorem 1). The proof is split into two cases. In the first case, the adversary $\mathcal{A}$ is given a valid password (e.g., $\mathcal{A}$ is given pw_i of user U_i). In the second case, $\mathcal{A}$ is given a valid biometrics, (e.g., $\mathcal{A}$ is given b'_i of user U_i). In both cases, if $\mathcal{A}$ can provide $b'_i \oplus r_i$ such that $\mathsf{HW}(b_i \oplus b'_i) \leqslant \tau$, then $\mathcal{A}$ succeeds in impersonating the user U_i.
Case 1: Assume that the attacker can successfully impersonate a user with a non-negligible probability . This means that $\mathcal{A}$ either (a) can forge the user biometrics and generate b'_i that matches the reference template b_i of the user U_i, or (b) knows $i \leftarrow \mathsf{ID}_i$ so that it can collude with DB to learn b_i. However, the probability of case (a) happening is bounded by the false acceptance rate, which can be bounded to be arbitrarily small, at the price of increased false rejection rate. And case (b) requires that $\mathcal{A}$ can learn i from $\mathsf{PIR}(i)$ or can derive i from ID_i, which contradicts both

the security of the PIR scheme and the fact that $i \leftarrow \mathsf{ID}_i$ is only known to SP. Therefore, $\mathcal{A}$ cannot impersonate a user knowing only the password. *Case 2:* Assume again that the attacker can successfully impersonate a user with a non-negligible probability. As in *Case 1*, this means that $\mathcal{A}$ either can guess the password (or the password-generated key r_i) or knows $i \leftarrow \mathsf{ID}_i$ so that it can collude with DB to learn r_i. However, while the probability of the former is negligible in $H_\infty(\mathsf{pw})$, the latter requires that $\mathcal{A}$ can learn i from $\mathsf{PIR}(i)$ or knows $i \leftarrow \mathsf{ID}_i$.

Therefore, $\mathcal{A}$ cannot successfully impersonate any user without having access to both authentication factors. Note that the use of salt prevents the adversary from practical dictionary attacks. Hence, it is important to salt the KDF, e.g. with the user ID, so that the security of the protocol in *Case 2* can be related to $H_\infty(\mathsf{pw})$.

Proof (of Theorem 2). Suppose that the adversary (i.e., the malicious DB) has a non-negligible advantage, i.e., $\left|\Pr\{\beta = \beta'\} - 1/2\right| \geqslant \mathsf{negl}(\lambda)$, where λ is a chosen security parameter for the protocol. Then, that means DB can guess the value of β (or i_β) from $\mathsf{PIR}(i_\beta)$ with a non-negligible probability. This in turn implies that DB can break the security of the underlying PIR scheme with a non-negligible probability, which contradicts the assumption that PIR is secure according to Def. 7. □

Proof (of Theorem 3). Suppose that the adversary can distinguish $(\mathsf{ID}_{i_0}, c_{i_0})$ from $(\mathsf{ID}_{i_0}, c_{i_1})$. Then the adversary can infer from $\mathsf{PIR}(i_0)$ (and the response to the query) the value of i_0, or infer from ID_{i_0} the value of i_0. This contradicts the security assumptions on the PIR, or the secrecy assumption on the correspondence between ID_{i_0} and i_0, respectively. □

Proof (of Theorem 4). Since the adversary (i.e., malicious SP+DB) has access to $b_i \oplus r_i$, $b'_i \oplus r_i$ and $b_i \oplus b'_i$ only, for all $i \in [1, N]$, it cannot learn more than what can already be learnt from these about b_i, b'_i and r_i (or the password from which the r_i is generated), as long as the KDF is secure and the password has sufficient min-entropy. The adversary can attempt to guess the value of b_i, b'_i or r_i at random using what the information at its disposal, but in order to verify whether the guess is correct, it needs access to an oracle that can answer whether the guessed values are correct. If the KDF is secure and the second factor has sufficient min-entropy, the expected number of queries needed to finally get an affirmative answer from such oracle is exponential in the min-entropy of r_i. □