Multi-factor Authenticated Key Exchange

David Pointcheval and Sébastien Zimmer

ENS – CNRS – INRIA, Paris, France
{pointcheval,zimmer}@di.ens.fr

Abstract. In order to increase the security for authenticated key exchange protocols, various authentication means can be used together. In this paper, we introduce a security model for multi-factor authenticated key exchange, which combines a password, a secure device, and biometric authentications. We thereafter present a scheme, that can be proven secure, in the random-oracle model.

1 Introduction

1.1 Motivation

Authentication is definitely one of the most important goal of modern cryptography. In order to avoid mistakes and impersonations during access control we can use various authentication means, possibly all together, that uniquely identify someone: a secret information, a biometric or user's belongings are the most well-known examples of such authentication factors for human beings. They represent the three classes of human authentication factors generally admitted, namely:

- something you *know* (as a secret password),
- something you *have* (as an unclonable secure device with a secret key),
- something you *are* (as a biometric).

Brainard *et al.* [15] have recently proposed a fourth authentication means: *someone you know*, also called the social networking. However we focus in this paper on the classical "three-factor authentication" technique, involving the three above factors. They are all subject to various types of attacks, notably attacks that cannot be avoided using cryptographic techniques only, but require external security protections:

- the password can be recovered through social engineering (phishing [29] or malwares), and thus users have to be careful when they enter it;
- the device can be stolen, open or cloned, and thus the device must be protected using tamper-resistant techniques;
- the biometric can be copied, and thus the sensor has to be able to correctly detect whether the controlled biometric is a real one, corresponds to the human-being under control, and to certify it.

S.M. Bellovin et al. (Eds.): ACNS 2008, LNCS 5037, pp. 277–295, 2008.

Combining the three factors in the same authentication protocol could increase the security since the adversary would have to break the three protections in order to win. However, involving the three factors does not necessarily requires the adversary to break all the protections in order to break the scheme, if the latter is not well designed: a security model for authentication based on a secret key, on a password and on a biometric, all together, has to be provided, in order to be able to formally prove that the design is correct.

In addition to simple authentication (access control), in case of success, the two parties may be interested in coming up with a common ephemeral secret key to establish a secure channel [8,17,19]. We are thus interested, not only in *authentication*, but in *Authenticated Key Exchange* [4,8]. In the following we focus on such AKE protocols, combining the three above authentication means. This basically means that if the three authentication verifications simultaneously succeed, then the 2 parties should come up with a session key that is semantically secure (indistinguishable from a truly random key to any other party), otherwise nobody learns anything.

Issues raised by PKI-based [8,17,18] and password-based [6,14,16] AKE are now well understood, and several solutions are known. The PKI/public-key setting is definitely the easiest case, since signatures [28] can be used to authenticate the flows, or alternatively the ability to decrypt, using an asymmetric encryption scheme [26,31]. In the password-based setting, one has to take care of the (off-line) dictionary attacks [9]. We indeed cannot avoid the *on-line* dictionary attack, which consists in trying to impersonate one party with a random password, and do it again, until the correct password is used. We thus want to prove that this is the best attack. Note that in many cases, such attacks can be prevented or damages can be reduced with appropriate techniques (delays after a failure, limited number of failures, etc).

However, biometric-based authentication raises quite different issues. First of all, biometric cannot be assumed a secret information. Indeed, recovering a fingerprint from the object someone has just touched is an easy task, or getting an image of the iris simply requires a camera. That is why considering biometric as a truly secret information and treating it the same way as a private key is not reasonable in practice, even if this scenario has often been assumed in the literature [30,24,12,13,23].

On the other hand, if the biometrics are public, how do we prevent an adversary from impersonating an honest user? The only way to use biometrics for authentication is to guarantee that the biometric template comes from a real living human being and not from a fake copy. Several technical solutions have been elaborated to guarantee this (authenticated channels, various biometric features controlled at the same time, sensor under human supervision, ...). The assumption that biometrics really come from the living human being under control is called the *liveness assumption*. It also implies that all computations made from the biometric data are done honestly. This assumption is not only useful for authentication, it is compulsory to ensure authentication security. This is a

strong, but necessary, assumption. Practical solutions exist to achieve, but are out of the scope of the this paper.

Secondly, and more importantly from a technical point of view, two measurements of the same biometric lead to different templates. Since a specific template cannot be reproduced, a matching mechanism has to be used to compare two templates and determine if they come from the same biometric or from two different persons. The matching can for example be based on a simple threshold on the distance between the candidate template and the reference template (as it is the case for iris [20]). However all known matching systems are not foolproof: they introduce two possible errors, "false acceptance" (when the system accepts someone it should not) and "false rejection" (when the system does not recognize someone it should). Therefore, an AKE protocol based on biometrics has to deal with these measurement errors and make this matching possible, but should not increase significantly the "false acceptance" and "false rejection" rates.

Finally, biometrics can be used to unequivocally identify an individual and are often linked with other personal information in the database. Since these databases can be vulnerable to internal or external adversaries, the privacy of the database is a classical requirement. Even if we already noticed that they cannot be considered as private information, biometric templates are critical data, especially when they are gathered in a database. Privacy is thus a major concern here.

1.2 Related Works

As already mentioned, literature about PKI-based and password-based AKE is rich of many results [8,17,18,6,14,16].

Dealing with biometric measurement errors is a much more challenging task and two dedicated tools were formalized by Dodis *et al.* [24]: secure sketches and fuzzy extractors. They allow, from an erroneous biometric measurement and public information, to always generate the same biometric template and random bitstring respectively. These tools were improved and allowed to design biometric-based AKE [24,12,13]. However, these tools rely on the assumption that biometrics are private information, which we do not allow in this paper.

Several efforts were taken to design authentication protocols were the matching is made on the client side [3]. But in client-side protocols the client sensor must record a reference biometric template for the user(s), which can be heavy if numerous people use the same sensor.

Despite all the efforts taken for 1-factor authentication or AKE protocols, literature does not tell much on multi-factor authentication protocols. In [11], an encoding for fingerprints is proposed, which is thereafter included in the design of a two-factor authentication protocol. Their fingerprint encoding has the property that two measurements of the same fingerprint leads to the same encoding, despite the errors. They make good use of it, since no matching is needed anymore and they can use classical cryptographic tools. They propose to use zero-knowledge proofs of knowledge [27], so that the database cannot have any information about the biometric template that it records. They assume that

the biometric is private, however their protocol can be proven secure even if biometric template is public. This protocol has nice features, but it heavily relies on the fact that thanks to their encoding, they get rid of errors. There is not such encoding for all biometrics and this protocol is therefore very restrictive. Furthermore, it achieves authentication only, but does not help to establish a secure channel.

1.3 Our Solution

We propose a Multi-Factor based AKE (MFAKE) which preserves database privacy. From three factors (a password, a high entropy secret key and a biometric template) the protocol generates a common semantically secure secret key, in order to establish a secure channel. The protocol is designed so that the matching is made on the server side and is adapted for a matching based on a simple threshold on the distance between the candidate template, and a reference template. Therefore, it is particularly well-suited to iris which is efficiently encoded on 1024-bit string, but can also apply on some other biometric techniques, with appropriate encoding.

Derived from PKI-AKE, PAKE and biometric-based AKE security models, we first define a new and clear security model for MFAKE protocols, which combines all the corresponding security properties. We chose to extend the Real-or-Random model, since the latter is strictly stronger than the Find-then-Guess model in the password-based setting [1]. The model allows the adversary to make several corruptions, on the secret key, the password, or the sensor. And despite two corrupt queries, the new keys should still remain semantically secure: in this model a protocol is provably as secure as the strongest remaining factor. Furthermore, our model also deals with the forward-secrecy, which means that, even when all the authentication means are corrupted, a session key established before the last corruption remains semantically secure. However, note that we only consider client-authentication (Test-queries will be allowed to the server only, in the formal security model below). This can be seen as a strong limitation, but it is not in practice: if the password and the secret keys are compromised, an adversary can easily play the role of the server, since there is no more secret (the biometric is public and the liveness assumption is valid on the client side only), whatever the protocol is. Authentication of the server to the client could be satisfied, until the two secret information related to the client (secret key and password) are compromised, but we do not address it in this model.

Then we also provide a protocol that is secure, according to this model, in the random-oracle [7]. This protocol records an encrypted version of the biometric template on the server side. Therefore privacy of the database (and thus of all the biometric templates) is preserved, even to the server, and thus even if the server is compromised. The protocol is proven to have a tight security proof: when the password is the last factor not to be corrupted, on-line dictionary attacks are the most efficient attacks that can be mounted; when the biometric is the last one, the adversary probability to be accepted is nearly equal to the false-acceptance

probability; when the secret key is still private, the security level is quite strong since it requires the adversary to break the Diffie-Hellman problem [22].

2 Security Model

In this section, we describe the security model for multi-factor authenticated key exchange (later denoted MFAKE). This model is built upon the usual password-authenticated key exchange security model [8,6], in the Real-or-Random indistinguishability framework [5,1].

2.1 Notation

We first explain the notation and the assumptions about the authentication means.

Participants, Sessions and Partnering. In a MFAKE, participants are either clients $\mathcal{C}$ or a unique, trusted server S. The server and every client can activate several instances at a time, in order to run several sessions concurrently. The instance i of the entity U, where U is a client or the server, is denoted as Π_U^i. This instance includes three variables, initialized as *null*:

- pid_U^i: the *partner identifier* which is the instance with whom Π_U^i believes it is interacting,
- sid_U^i: the *session identifier*, in practice it can be the transcript seen by Π_U^i (concatenation of the received/sent flows, excepted the last one).
- acc_U^i: a boolean variable which is fixed at the end of the session and denotes whether the instance Π_U^i goes in an *accepted state* or not.

The two instances Π_U^i and $\Pi_{U'}^j$ are said to be *partners* if the following conditions are fulfilled:

1. $\mathsf{pid}_U^i = \Pi_{U'}^j$ and $\mathsf{pid}_{U'}^j = \Pi_U^i$;
2. $\mathsf{sid}_U^i = \mathsf{sid}_{U'}^j \neq null$;
3. $\mathsf{acc}_U^i = \mathsf{acc}_{U'}^j = 1$.

Long-Lived Keys. Each client $\mathcal{C}$ owns a tuple $t_{\mathcal{C}} = (\mathcal{D}_{\mathcal{C}}, \mathsf{sk}_{\mathcal{C}}, \mathsf{pwd}_{\mathcal{C}})$, where $\mathcal{D}_{\mathcal{C}}$ is a probability distribution for his biometric, while $\mathsf{sk}_{\mathcal{C}}$ and $\mathsf{pwd}_{\mathcal{C}}$ are a high-entropy private key and a low-entropy password respectively. The server holds a list of tuples $t_S = \langle t_S[\mathcal{C}] \rangle$, where $t_S[\mathcal{C}]$ is a transformed-tuple of $t_{\mathcal{C}}$. More precisely, when the client $\mathcal{C}$ enrolls in the system, he generates a biometric template $W_{\mathcal{C}}$, according to the distribution $\mathcal{D}_{\mathcal{C}}$, as well as two private data $\mathsf{sk}_{\mathcal{C}}$ and $\mathsf{pwd}_{\mathcal{C}}$. The tuple $t_S[\mathcal{C}]$ is then an (injective) transformation of $(W_{\mathcal{C}}, \mathsf{sk}_{\mathcal{C}}, \mathsf{pwd}_{\mathcal{C}})$.

Biometric Templates. As explained above, for each client $\mathcal{C}$, $\mathcal{D}_{\mathcal{C}}$ defines the probability distribution of his biometric (fingerprint, face, iris, etc). In order to be relevant for authentication, we have to make some assumptions about the matching process, and more precisely about the encoding and the Hamming distance, since we will use this distance in the matching decision:

– on the one hand, the distance between two templates $W_{\mathcal{C}}$ and $W'_{\mathcal{C}}$ of the same biometric is low with great probability. More concretely, there is a threshold t, such that for any $\mathcal{C}$,

$$\Pr[W_{\mathcal{C}} \leftarrow \mathcal{D}_{\mathcal{C}}, W'_{\mathcal{C}} \leftarrow \mathcal{D}_{\mathcal{C}} : d_H(W_{\mathcal{C}}, W'_{\mathcal{C}}) \leq t] \geq 1 - \varepsilon_{\mathsf{fr}}.$$

The subscript fr stands for "false rejection".

– on the other hand, for any pair of distinct clients $\mathcal{C} \neq \mathcal{C}'$, the distance between $W_{\mathcal{C}}$ and $W_{\mathcal{C}'}$ is high with great probability. More precisely, there exist a threshold $\tau \geq t$, such that for any $\mathcal{C} \neq \mathcal{C}'$,

$$\Pr[W_{\mathcal{C}} \leftarrow \mathcal{D}_{\mathcal{C}}, W_{\mathcal{C}'} \leftarrow \mathcal{D}_{\mathcal{C}'} : d_H(W_{\mathcal{C}}, W_{\mathcal{C}'}) > \tau] \geq 1 - \varepsilon_{\mathsf{fa}}.$$

The subscript fa stands for "false acceptance".

We assume that for all the clients $\mathcal{C}$, the biometric distribution $\mathcal{D}_{\mathcal{C}}$ is public. Under the liveness assumption explained below the biometric acceptance will guarantee that the client *is* like the intended client.

Private Data. The private key component $\mathsf{sk}_{\mathcal{C}}$ is chosen uniformly in a set of private keys $Keys$, where $Keys$ is assumed to be very large (with high entropy), such that $1/\#Keys$ is negligible. It will be stored in a secure device. The acceptance of this private key, with respect to a public key, will guarantee that the client *has* the device.

On the opposite, the password component $\mathsf{pwd}_{\mathcal{C}}$ is chosen in a fixed low-entropy dictionary $Dict \subset \mathbb{Z}_p^\star$, according to the probability distribution $\mathcal{D}_{\mathsf{pwd}}$. We denote by $\mathcal{D}_{\mathsf{pwd}}(q)$ the sum of the probability of the q most probable passwords according to $\mathcal{D}_{\mathsf{pwd}}$. The knowledge of the password will guarantee that the client *knows* it.

Liveness Assumption. Since we assume the biometric to possibly be public (the opposite assumption is not reasonable in practice), then the *liveness assumption* [32,21], though quite strong, is necessary. It prevents the attacker from making replay attacks and from altering the computations made by the sensor. The liveness assumption implies that the biometric is fresh, comes from a real living person (and not using a fake biometric feature), and that the computations are made from this biometric honestly.

To model this assumption, a computation oracle $\mathsf{Compute}(\Pi^i_{\mathcal{C}}, W', \mathsf{sk}, \mathsf{pwd})$ is used: according to the state of the client instance $\Pi^i_{\mathcal{C}}$, from the secrets $\mathsf{sk}, \mathsf{pwd}$ and a random value of W', it computes honestly the message which would have been generated by $\mathcal{C}$ with these inputs, following the protocol.

As it models an attempt of the attacker to authenticate using its own biometric, W' has to be chosen according to a (wrong) distribution $\mathcal{D}$, such that $\Pr[W' \leftarrow \mathcal{D}, W_{\mathcal{C}} \leftarrow \mathcal{D}_{\mathcal{C}} : d_H(W', W_{\mathcal{C}}) > \tau] \geq 1 - \varepsilon_{\mathsf{fa}}$.

With the *liveness assumption* for the client $\mathcal{C}$, we consider that all the messages involving the biometric, claimed to be sent by $\mathcal{C}$, have been previously generated by the computation oracle.

Corruption. As explained below, the adversary will be allowed to corrupt a client $\mathcal{C}$, by learning the password $\mathsf{pwd}_{\mathcal{C}}$ (phishing), by getting the private key $\mathsf{sk}_{\mathcal{C}}$ (side-channel attack on the device), or by breaking the above liveness assumption (attack on the sensor).

2.2 Semantic Security

Adversarial Capabilities and Goals. The semantic security of the key is modeled using the Real-or-Random paradigm [5,1]. At the beginning of the game, the challenger chooses a random bit b which determines its behavior when answering Test-queries during the game (it provides either real keys or random keys to the adversary). The adversary may interact with protocol instances through several oracles, and at the end of the game, she sends a bit b'. If $b = b'$, she wins, otherwise, she looses. The available queries are as follows:

- $\mathsf{Send}(m, \Pi_U^i)$: this query allows the adversary to play with the instances, by intercepting, forwarding, modifying or creating messages. The output of this query is the answer generated by instance Π_U^i to the message m. As stated above, if $\mathsf{pid}_S^i = \Pi_{\mathcal{C}}^j$ is the client instance with whom the server believes to talk, if the liveness assumption still holds for the client $\mathcal{C}$ (no corruption) and if the computation of m involves the biometric, then m has to have been previously generated through a $\mathsf{Compute}(\Pi_{\mathcal{C}}^j, W', \mathsf{sk}, \mathsf{pwd})$ query.
- $\mathsf{Reveal}(\Pi_U^i)$: this query models the leakage of information about the session key agreed on by the parties. For example, if it is misused afterward. Therefore, if no session key is defined for this instance, or if the instance (or its partner) has been tested (see below), then the output is $\perp$. Otherwise, the oracle outputs the session key computed by the instance Π_U^i.
- $\mathsf{CorruptKey}(\mathcal{C}, a)$: this query models corruption capabilities of the adversary. She can indeed steal/break one or several authentication factors of clients.
  - If $a = 1$, the oracle outputs the password $\mathsf{pwd}_{\mathcal{C}}$ of $\mathcal{C}$;
  - if $a = 2$, the oracle outputs the secret key $\mathsf{sk}_{\mathcal{C}}$ of $\mathcal{C}$;
  - if $a = 3$, the attacker is now allowed to submit *any* message involving the biometry, without asking the computation oracle $\mathsf{Compute}$ before. It models the attack against the liveness assumption.

  Note that in the following, we will restrict to non-adaptive corruptions: no corruption can be performed during a session, but before a new session starts.

To formally model the semantic security with respect to client authentication, the adversary can ask Test-queries, but to the server only: we are interested in the privacy of the key established with the real server only. We only consider adversaries whose goal is to impersonate a client to the server. Of course, to achieve this goal, the adversary may try to impersonate the server to the client in order to learn some information about the long-lived keys of the client. But only a client impersonation will be considered as a successful attack:

- $\mathsf{Test}(\Pi_S^i)$: if Π_S^i is not *fresh* (see below), then output $\perp$, otherwise the oracle sends
  - the session key of instance Π_S^i (that is $\mathsf{Reveal}(\Pi_S^i)$), if $b = 1$ – the real case;
  - a random key from the same domain, if $b = 0$ – the random case.

Freshness. The freshness notion basically defines session keys that are not trivially known to the adversary. Since we will focus on the freshness of the server only, we say that the session key of instance Π_S^i is fresh if:

- upon acceptance, $\mathcal{C}$ (corresponding to the partner of Π_S^i) was not *fully corrupted*. This means that strictly less then 3 $\mathsf{CorruptKey}$-queries had been asked to the client $\mathcal{C}$;
- no Reveal-query has been sent to either Π_S^i or its partner.

Semantic Security. Let denote by Succ the event that the adversary $\mathcal{A}$ correctly guesses the bit b used by the challenger during the above attack game. The **mfake-advantage** $\mathsf{adv}_P^{\mathtt{mfake}}(\mathcal{A})$ and the advantage function of the protocol P are respectively:

$$\mathsf{adv}_P^{\mathtt{mfake}}(\mathcal{A}) = 2 \cdot \Pr[\mathsf{Succ}] - 1, \quad \mathsf{adv}_P^{\mathtt{mfake}}(t, Q) = \max_{\mathcal{A}} \left\{ \mathsf{adv}_P^{\mathtt{mfake}}(\mathcal{A}) \right\},$$

where the maximum is over all the attackers with time-complexity at most t and number of queries at most Q.

Forward-Secrecy. Forward-secrecy means that as soon as a session key is securely generated (semantically secure), it will remain secure even after corruption. In order to capture this security level, the model must allow the adversary to perform Test-queries, even when the 3 $\mathsf{CorruptKey}$-queries have been asked, but on sessions completed before the full corruption of the client. One can also consider that upon acceptance, a session is fresh if less than 3 $\mathsf{CorruptKey}$-queries have been asked.

Client Authentication. We also usually model an attack against the unilateral authentication of the client to the server by considering sessions where the server accepts, but without any client-partner. Let denote by Succ the event that a server instance accepts with no partner instance of the client (with the same partial transcript).

The **auth-success** $\mathsf{Succ}_P^{\mathtt{auth}}(\mathcal{A})$ and the success function of the protocol P are respectively:

$$\mathsf{Succ}_P^{\mathtt{auth}}(\mathcal{A}) = \Pr[\mathsf{Succ}], \quad \mathsf{Succ}_P^{\mathtt{auth}}(t, Q) = \max_{\mathcal{A}} \left\{ \mathsf{Succ}_P^{\mathtt{auth}}(\mathcal{A}) \right\},$$

where the maximum is over all the attackers with time-complexity at most t and number of queries at most Q.

3 Description of the Protocol

The complete description of our protocol is provided Figure 1. It assumes a common setup, with parameters (u, v, p, g, q), where g is an element of order q in $\mathbb{Z}_p^\star$, and generates the subgroup $\mathbb{G}$. Then, u and v are random elements in $\mathbb{G}$. We also model H as a random oracle [7].

The server stores all the data corresponding to user $\mathcal{C}$, provided during the enrollment phase:

- the public key $h = g^{x_{\mathcal{C}}}$, related to the high-entropy secret $x_{\mathcal{C}}$;
- an El Gamal encryption [25] of the reference biometric template $W_{\mathcal{C}} = (W_i)_{i \le N}$ —where W_i is the i-th bit of $W_{\mathcal{C}}$ and N the number of bits— under the public key $h = g^{x_{\mathcal{C}}}$, that is tuple of pairs $(g^{r_i}, h^{r_i} g^{W_i})_i$;
- the password $\mathsf{pwd}_{\mathcal{C}} \in \mathcal{D} \subset \mathbb{Z}_q^\star$.

One can note that the server actually does not know the biometrics of the clients since they are encrypted under keys chosen by the clients.

$\mathcal{C} : (W'_{\mathcal{C}} = (W'_i)_i, \mathsf{sk}_{\mathcal{C}} = x_{\mathcal{C}}, \mathsf{pwd}_{\mathcal{C}})$		$\mathcal{S} : \left((g^{r_i}, h^{r_i} g^{W_i})_i, h = g^{x_{\mathcal{C}}}, \mathsf{pwd}_{\mathcal{C}}, \mathcal{C}\right)_{\mathcal{C}}$
$b \xleftarrow{\$} \mathbb{Z}_q$ and $B = g^b$, $B^* = B \cdot v^{\mathsf{pwd}_{\mathcal{C}}}$	$\xrightarrow{\mathcal{C},\, B^*}$	
		For $1 \le i \le N$,
		$r'_i \xleftarrow{\$} \mathbb{Z}_q$ and compute
		$g^{s_i} = g^{r'_i} \cdot g^{r_i}, h^{s_i} g^{W_i} = h^{r'_i} \cdot h^{r_i} g^{W_i}$
	$\xleftarrow{\mathcal{S}, (g^{s_i})_i, A^*}$	$a \xleftarrow{\$} \mathbb{Z}_q$, $A = g^a$, $A^* = A \cdot u^{\mathsf{pwd}_{\mathcal{C}}}$
For $1 \le i \le N$,		
compute $H(K'_i) = \alpha'_i \Vert \beta'_i \Vert k'_i$ with:		
$K_{\mathcal{C}} = \left(\frac{A^*}{u^{\mathsf{pwd}_{\mathcal{C}}}}\right)^b, K^i_{\mathcal{C}} = (g^{s_i})^{x_{\mathcal{C}}} \cdot g^{W'_i}$		
$K'_i = \mathcal{S} \Vert \mathcal{C} \Vert (g^{s_i})_i \Vert A^* \Vert B^* \Vert K^i_{\mathcal{C}} \Vert K_{\mathcal{C}} \Vert \mathsf{pwd}_{\mathcal{C}} \Vert i$	$\xrightarrow{(\alpha'_i)_i}$	
		For $1 \le i \le N, H(K_i) = \alpha_i \Vert \beta_i \Vert k_i$ with:
		$K_{\mathcal{S}} = (\frac{B^*}{v^{\mathsf{pwd}_{\mathcal{C}}}})^a, K^i_{\mathcal{S}} = h^{s_i} \cdot g^{W_i}$
		$K_i = \mathcal{S} \Vert \mathcal{C} \Vert (g^{s_i})_i \Vert A^* \Vert B^* \Vert K^i_{\mathcal{S}} \Vert K_{\mathcal{S}} \Vert \mathsf{pwd}_{\mathcal{C}} \Vert i$
		If $\#\{i \colon \alpha_i \ne \alpha'_i\} \le t$
		Then $\mathsf{acc} = 1$, $K = \mathsf{lsb}_{\mathsf{k}} \left(\Vert_{i \colon \alpha_i = \alpha'_i} k_i \right)$
	$\xleftarrow{(\beta_i)_i}$	Else $\mathsf{acc} = 0$, $K \xleftarrow{\$} \{0,1\}^{\mathsf{k}}, \beta_i \xleftarrow{\$} \{0,1\}^{\ell}$
If $\#\{i \colon \beta_i \ne \beta'_i\} \le t$		
Then $\mathsf{acc} = 1$, $K' = \mathsf{lsb}_{\mathsf{k}} \left(\Vert_{i \colon \alpha_i = \alpha'_i} k'_i \right)$		
Else $\mathsf{acc} = 0$, $K' \xleftarrow{\$} \{0,1\}^{\mathsf{k}}$		

Fig. 1. Our MFAKE Protocol

For authenticating himself, the client $\mathcal{C}$ owns an ephemeral biometric template $W'_{\mathcal{C}} = (W'_i)_i$; the long-term private key $x_{\mathcal{C}}$; and the password $\mathsf{pwd}_{\mathcal{C}} \in \mathcal{D} \subset \mathbb{Z}_q^\star$.

The protocol guarantees that, if the ephemeral template $W'_{\mathcal{C}}$ is close enough to the reference template $W_{\mathcal{C}}$ (who you are), if the private key $x_{\mathcal{C}}$ corresponds to the public key h (what you have), and if the passwords are the same (what

you know), then the server accepts the client, and they agree on an ephemeral common secret $K' = K$.

We namely want to prove that unless the three authentication factors have been corrupted, no adversary can impersonate a client to the server. And all the keys actually agreed on between a client and the server are semantically secure, even after corruptions (forward-secrecy).

Basically, (B^*, A^*) corresponds to the Password-Authenticated part (similar to EKE [9,2]); for each i, (h, g^{s_i}) is a Diffie-Hellman key agreement which leads to the key $g^{x_{\mathcal{C}} \cdot s_i}$, with $x_{\mathcal{C}}$ used for authentication. Note that the s_i are rerandomized every time, so that the Diffie-Hellman key exchange is not static. The bit W_i (or W'_i for the client) of the biometric template is used as a mask and we obtain $g^{x_{\mathcal{C}} \cdot s_i} \cdot g^{W_i}$ ($g^{x_{\mathcal{C}} \cdot s_i} \cdot g^{W'_i}$ for the client). If for most of i the masks W_i and W'_i are equal, then for most of i the authenticators α'_i and verifiers α_i will be equal also, as well as β_i and β'_i, and k_i and k'_i.

To define the partnership in our protocol, we have to precise that the sid is equal to the first two flows $((\mathcal{C}, B^*), (\mathcal{S}, (g^{s_i})_i, A^*))$.

4 Security of the Protocol

Before stating the security result, let us remind the computational assumption on which the security will rely.

Computational Diffie-Hellman Problem. Let $\mathbb{G}$ be a cyclic group of order q. Let g be a generator of $\mathbb{G}$, let (x, y) be two integers uniformly chosen in $\mathbb{Z}_q$. The computational Diffie-Hellman problem states that, given (g^x, g^y), it is difficult to compute $g^{xy} = \mathsf{CDH}_g(g^x, g^y)$.

Let A be a CDH-adversary with running time at most T. We denote by $\mathsf{Succ}^{\mathsf{cdh}}_g(A)$ the probability that A succeeds in computing g^{xy} from (g^x, g^y) and by $\mathsf{Succ}^{\mathsf{cdh}}_g(T) = \max_A\{\mathsf{Succ}^{\mathsf{cdh}}_g(A)\}$ where the maximum is taken over all the adversaries with running-time at most T.

Biometric. We remind that for any client $\mathcal{C}$ and any adversary which uses a true biometric W', we have $\Pr[d_H(W', W_{\mathcal{C}}) > \tau] \leq 1 - \varepsilon_{\mathsf{fa}}$ where τ is an integer greater than t. The protocol does not increase the false-rejection probability but it does increase the false-acceptance probability, due to the additional check on the $\alpha_i = \alpha'_i$ equalities. The increasing is upper-bounded by

$$\Pr\left[\#\{i : \alpha'_i \neq \alpha_i\} \leq t \,|\, d_H(W', W_{\mathcal{C}}) > \tau\right] \leq \frac{\binom{\tau}{\tau - t}}{2^{\ell(\tau - t)}}.$$

A protocol session between two honest entities is correct if for all i, $K_i = K'_i$ is equivalent to $\alpha_i = \alpha'_i$ or $\beta_i = \beta'_i$. It fails if there is an index i such that $K_i \neq K'_i$ and $\alpha_i = \alpha'_i$ or $\beta_i = \beta'_i$. As there are at most t indexes i such that $K_i \neq K'_i$ the probability that an honest protocol session is not correct is upper bounded by $2t \Pr[\alpha_i = \alpha'_i \colon K_i \neq K'_i]$ which is equal to $2t/2^{\ell}$.

Theorem 1. *Let us consider the above protocol P over a group of prime order q, where the dictionary of passwords is equipped with the distribution $\mathcal{D} \subset \mathbb{Z}_q^\star$. Let $\mathcal{A}$ be an adversary against the semantic security within a time bound T, with less than q_{session} Send-queries and asking less than q_h queries to the random oracle. Then we have*

$$\mathsf{adv}_P^{\mathtt{mfake}}(\mathcal{A}) \le 2\sum_{\mathcal{C}} \mathcal{D}(q_{\mathcal{C}}) + 4q_h^2 \cdot \mathsf{Succ}_g^{\mathsf{cdh}}(T + 4\tau_e) + \frac{q_{\mathsf{session}}^2}{q} + \frac{2q_h}{q}$$

$$\mathsf{Succ}_P^{\mathtt{auth}}(\mathcal{A}) \le \sum_{\mathcal{C}} \mathcal{D}(q_{\mathcal{C}}) + 2q_h^2 \cdot \mathsf{Succ}_g^{\mathsf{cdh}}(T + 4\tau_e) + \frac{q_{\mathsf{session}}^2}{2q} + \frac{q_h}{q}$$
$$+ q_{\mathsf{session}}\left(\varepsilon_{\mathsf{fa}} + \frac{\binom{\tau}{\tau-t}}{2^{\ell(\tau-t)}} + \frac{N^t \cdot (2^\ell - 1)^t}{2^{\ell N}(t-1)!}\right)$$

where τ_e denotes the computational time for one exponentiation and $q_{\mathcal{C}}$ the number of active sessions the adversary ran against client $\mathcal{C}$.

Proof. The proof consists of a sequence of games:

Game 0. This is the real attack game, against the protocol. We are interested in the two following events:

- $\mathbf{S}_0$ (for semantic security) which occurs if the adversary correctly guesses the bit b chosen at the beginning of the game.
- $\mathbf{A}_0$ (for client authentication), which occurs if a server instance accepts with no partner instance of the client (with the same transcript).

Actually in any game $\mathbf{G}_n$, we study the event $\mathbf{A}_n$, and the event $\mathbf{S}_n$. Note that

$$\mathsf{adv}^{\mathtt{mfake}}(\mathcal{A}) = 2\Pr[\mathbf{S}_0] - 1, \quad \mathsf{Succ}^{\mathtt{auth}}(\mathcal{A}) = \Pr[\mathbf{A}_0].$$

Therefore

$$\mathsf{adv}_P^{\mathtt{mfake}}(\mathcal{A}) = 2\Pr[\mathbf{S}_n] - 1 + 2(\Pr[\mathbf{S}_0] - \Pr[\mathbf{S}_n]) \le 2\Pr[\mathbf{S}_n] - 1 + 2\sum_{i=0}^{n-1} \Delta_i$$
$$\mathsf{Succ}_P^{\mathtt{auth}}(\mathcal{A}) = \Pr[\mathbf{A}_n] + (\Pr[\mathbf{A}_0] - \Pr[\mathbf{A}_n]) \le \Pr[\mathbf{A}_n] + \sum_{i=0}^{n-1} \Delta_i,$$

if we denote by Δ_i the distance between games $\mathbf{G}_i$ and $\mathbf{G}_{i+1}$.

Game 1. In this game, we simulate the random oracles (H, but also an additional function H' that will appear in the game $\mathbf{G}_3$) as usual by maintaining lists Λ_H and $\Lambda_{H'}$. We also simulate all the instances, as the real players would do, for the Send-queries and the Reveal and Test-queries. From this simulation, we see that the game is indistinguishable from the real attack: $\Delta_0 = 0$.

Note that since the probability distributions of the biometrics are public, we draw a random reference template for each client, to be used/known by the server. And when needed (simulation of a client), we can draw a random biometric according to the (public) probability distribution of the client's biometric.

Game 2. In order to guarantee independence of the sessions, we cancel games in which some collision on the session transcripts $((\mathcal{C}, B), (\mathcal{S}, A^*, (g^{s_i})_i))$ appear. Since transcripts involve at least one honest party, $(A^*, (g^{s_i})_i)$ or B^* is truly uniformly distributed. Therefore the collision probability is upper bounded by $q_{\mathsf{session}}^2/2q$, where q_{session} is the number of sessions: $\Delta_1 \leq q_{\mathsf{session}}^2/2q$.

Game 3. We now replace the generation of the authenticators and session keys with a private oracle H' instead of H, for all the sessions that are fresh (which can be tested: involving a server so that the intended client is not fully corrupted), and also for all the sessions involving a client (but no server) for which the password and the secret key are unknown (none of the 1-CorruptKey and 2-CorruptKey-queries has been asked): instead of using the public oracle H, we use the private oracles H', on K_i and K'_i computed as

$$K_i = \mathcal{S}\Big\|\mathcal{C}\Big\|(g^{s_i})_i\Big\|A^*\Big\|B^*\Big\|g^{W_i}\Big\|i \quad K'_i = \mathcal{S}\Big\|\mathcal{C}\Big\|(g^{s_i})_i\Big\|A^*\Big\|B^*\Big\|g^{W'_i}\Big\|i.$$

As already explained, we have chosen a random reference biometric template for each user. And when needed, we can draw a random biometric according to the (public) probability distribution of the client's biometric. Thus we can include g^{W_i} or $g^{W'_i}$ in the above public computations, in order to make K_i and K'_i possibly different, even for compatible biometric templates.

We do not use none of $K_{\mathcal{C}}$, $K^i_{\mathcal{C}}$, $K_{\mathcal{S}}$ and $K^i_{\mathcal{S}}$ anymore, therefore we can omit their computations. Besides, we do not use A and B anymore, therefore we can change the computations of A^* and B^* by $a^* \xleftarrow{\$} \mathbb{Z}_q$, $A^* = g^{a^*}$ and $b^* \xleftarrow{\$} \mathbb{Z}_q$, $B^* = g^{b^*}$. Lastly, since we do not use neither the password nor the secret key, we can choose them at the last moment: for the password-corrupt query (1-CorruptKey) or for the secret key-corrupt query (2-CorruptKey), or at the very end of the game only (when the adversary gives her answer).

However, when a client is fully corrupted (adversary against the server) or the adversary plays against a client from which she knows the password and the secret key, the keys K_i and K'_i are computed normally and we use the public oracle H again.

Note that we restrict to non-adaptive corruptions, and thus, when a session starts, we know the corruption status of a client. Then, requests to the Compute-oracle will also focus on such a session for which we know the corruption status, since the biometric is only involved in the third round. The latter oracle indeed has to know how to perform the simulation of K'_i, using either H or H'.

The games $\mathbf{G}_2$ and $\mathbf{G}_3$ are indistinguishable unless some specific hash query is asked (for a session made before the last CorruptKey-query): if the adversary asks either

$$\mathcal{S}\Big\|\mathcal{C}\Big\|(g^{s_i})_i\Big\|A^*\Big\|B^*\Big\|K^i_{\mathcal{C}}\Big\|K_{\mathcal{C}}\Big\|\mathsf{pwd}_{\mathcal{C}}\Big\|i \text{ or } \mathcal{S}\Big\|\mathcal{C}\Big\|(g^{s_i})_i\Big\|A^*\Big\|B^*\Big\|K^i_{\mathcal{S}}\Big\|K_{\mathcal{S}}\Big\|\mathsf{pwd}_{\mathcal{C}}\Big\|i,$$

for some transcript $((\mathcal{C}, B^*), (\mathcal{S}, A^*, (g^{s_i})_i))$, and some index i, to the H function, whereas the H' function has been used by the simulator. We denote by AskH_3 such an event. Note that, it can be decided whether this event happened only when the password and the secret key have both been chosen.

In this game, for all clients, the $(\alpha_i)_i$, the $(\beta_i)_i$ and the key are computed from a private random oracle. Therefore, whatever the bit b involved in the Test-query, the answer is random, and independent for all the sessions, unless some transcript $((\mathcal{C}, B^*), (\mathcal{S}, A^*, (g^{s_i})_i))$ appeared twice, but this has already been excluded in game $\mathbf{G}_2$. Therefore we have:

$$\Delta_2 \leq \Pr[\mathsf{AskH}_3] \qquad \text{and} \qquad \Pr[\mathbf{S}_3] = \frac{1}{2}.$$

Similarly, the only possibility for the adversary to authenticate against a true server instance is to guess the α_i at random or to use the Compute-oracle, unless some transcript $((\mathcal{C}, B^*), (\mathcal{S}, A^*, (g^{s_i})_i))$ appeared twice.

If she tries to guess the α_i at random, since $|\alpha_i| = \ell$, then her probability to succeed is upper-bounded by:

$$\frac{1}{2^{N\ell}} \cdot \sum_{k=0}^{t} \binom{N}{k} (2^\ell - 1)^k \leq \frac{N^t \cdot (2^\ell - 1)^t}{2^{\ell N}(t-1)!},$$

If she uses the Compute-oracle, all the α'_i and β_i are generated through a trusted computation oracle and since the adversary uses her own biometric W', which is, with high probability, quite different from the client biometric $W_{\mathcal{C}}$, her probability to succeed is exactly the false-acceptance probability computed earlier.

As a consequence,

$$\Pr[\mathbf{A}_3] \leq q_{\mathsf{session}} \left(\varepsilon_{\mathsf{fa}} + \frac{\binom{\tau}{\tau - t}}{2^{\ell(\tau - t)}} + \frac{N^t \cdot (2^\ell - 1)^t}{2^{\ell N}(t-1)!} \right).$$

Game 4. Our goal is now to upper-bound the probability of event AskH_3. We denote by AskH_4 the same event in this game and have $\mathsf{AskH}_3 \leq \mathsf{AskH}_4 + \Delta_3$. In this game, we receive a Diffie-Hellman pair $(X = g^x, Y = g^y)$, and we will try to show that the probability of event AskH is related to the probability of computing the Diffie-Hellman value of (X, Y). We set $u = X$ and $v = Y$. We furthermore cancel games in which, for a transcript $((\mathcal{C}, B^*), (\mathcal{S}, A^*, (g^{s_i})_i))$, which both

- was generated before a password-corrupt query to the client $\mathcal{C}$ was made
- comes from an execution involving the adversary, against either an instance of the client $\mathcal{C}$ or the server $\mathcal{S}$

there are two tuples $(A^*, B^*, \mathsf{CDH}_g(A^*/u^{\mathsf{pwd}_k}, B^*/v^{\mathsf{pwd}_k}), i_k)$, with two different passwords pwd_0 and pwd_1 and two, possibly different, indexes i_0 and i_1, such that

$$\mathcal{S} \| \mathcal{C} \| (g^{s_i})_i \| A^* \| B^* \| K_{\mathcal{S}}^{i_k} \| \mathsf{CDH}_g(A^*/u^{\mathsf{pwd}_k}, B/v^{\mathsf{pwd}_k}) \| \mathsf{pwd}_k \| i_k$$

is in Λ_H.

Distance. We first easily show that $\Delta_3 \leq q_h^2 \cdot \mathsf{Succ}_g^{\mathsf{cdh}}(T + 3\tau_e)$. To this aim, we remind that the distance we study comes from the fact we have canceled games

in which, for some specific transcript $((\mathcal{C}, B^*), (\mathcal{S}, A^*, (g^{s_i})_i))$ —which was generated before a password-corrupt query to the client $\mathcal{C}$ was made and which comes from an execution involving the adversary, against either an instance of the client $\mathcal{C}$ or the server $\mathcal{S}$—, there are two tuples $(A^*, B^*, \mathsf{CDH}_g(A^*/u^{\mathsf{pwd}_k}, B^*/v^{\mathsf{pwd}_k}), i_k)$, with two different passwords pwd_0 and pwd_1 and two, possibly different, indexes i_0 and i_1, such that

$$\mathcal{S}\Big\|\mathcal{C}\Big\|(g^{s_i})_i\Big\|A^*\Big\|B^*\Big\|K_{\mathcal{S}}^{i_k}\Big\|\mathsf{CDH}_g(A^*/u^{\mathsf{pwd}_k}, B/v^{\mathsf{pwd}_k})\Big\|\mathsf{pwd}_k\Big\|i_k$$

is in Λ_H.

If such a pair exists, then for $k = 0, 1$: $\mathsf{CDH}_g(A^*/u^{\mathsf{pwd}_k}, B^*/v^{\mathsf{pwd}_k})$ is equal to

$$\frac{\mathsf{CDH}_g(A^*, B^*) \cdot \mathsf{CDH}_g(u^{-1}, B^*)^{\mathsf{pwd}_k} \cdot \mathsf{CDH}_g(A^*, v^{-1})^{\mathsf{pwd}_k}}{\mathsf{CDH}_g(u, v)^{\mathsf{pwd}_k^2}}.$$

Since we simulated either A^* or B^*, knowing the discrete logarithms, we can extract $\mathsf{CDH}_g(X, Y)$. Let us show it when $B^* = g^{b^*}$, it works similarly when we know $A^* = g^{a^*}$: since the two passwords are different and non-zero,

$$\mathsf{CDH}_g(X, Y) = \frac{\mathsf{CDH}_g(A^*/u^{\mathsf{pwd}_0}, B^*/v^{\mathsf{pwd}_0})^{1/\mathsf{pwd}_0(\mathsf{pwd}_1 - \mathsf{pwd}_0)}}{\mathsf{CDH}_g(A^*/u^{\mathsf{pwd}_1}, B^*/v^{\mathsf{pwd}_1})^{1/\mathsf{pwd}_1(\mathsf{pwd}_1 - \mathsf{pwd}_0)}} \cdot (A^*)^{-b^*/\mathsf{pwd}_0\mathsf{pwd}_1}.$$

Conclusion. In order to conclude with the computation of $\Pr[\mathsf{AskH}_4]$, we distinguish the events when the transcript $((\mathcal{C}, B^*), (\mathcal{S}, A^*, (g^{s_i})_i))$ comes from an execution between:

- two instances of $\mathcal{C}$ and $\mathcal{S}$, or an instance of $\mathcal{C}$ or $\mathcal{S}$ and the adversary but the flows are all oracle-generated, this event is denoted by $\mathsf{AskH\text{-}Passive}_4$;
- an instance of $\mathcal{C}$ and the adversary, where at least one flow is not oracle-generated, this event is denoted by $\mathsf{AskH\text{-}with}\mathsf{C}_4$;
- an instance of $\mathcal{S}$ and the adversary, where at least one flow is not Compute-oracle-generated, this event is denoted by $\mathsf{AskH\text{-}with}\mathsf{S}_4$;

Assume that there is a tuple $(A^*, B^*, D = \mathsf{CDH}_g(A^*/u^{\mathsf{pwd}}, B^*/v^{\mathsf{pwd}}))$ such that

$$\mathcal{S}\Big\|\mathcal{C}\Big\|(g^{s_i})_i\Big\|A^*\Big\|B^*\Big\|K'_{\mathcal{C}}\Big\|D\Big\|\mathsf{pwd}\Big\|i \text{ or } \mathcal{S}\Big\|\mathcal{C}\Big\|(g^{s_i})_i\Big\|A^*\Big\|B^*\Big\|K'_{\mathcal{S}}\Big\|D\Big\|\mathsf{pwd}\Big\|i$$

is in Λ_H, for any password pwd of the adversary's choice.

If the corresponding transcript $((\mathcal{C}, B^*), (\mathcal{S}, A^*, (g^{s_i})_i))$ comes from an execution between instances of $\mathcal{C}$ and $\mathcal{S}$, it means that both A^* and B^* have been simulated (and the adversary was only passive). In this case, we know the discrete logarithms a^* and b^*, and

$$\mathsf{CDH}_g(A^*/u^{\mathsf{pwd}}, B^*/v^{\mathsf{pwd}}) = \frac{g^{a^*b^*} \cdot (v^{a^*}u^{b^*})^{\mathsf{pwd}}}{\mathsf{CDH}_g(v, u)^{\mathsf{pwd}^2}}.$$

Since pwd is non-zero in $\mathbb{Z}_q$, it can be inverted modulo q and then,

$$\mathsf{CDH}_g(X, Y) = \left(\frac{g^{a^*b^*} \cdot v^{a^* \cdot \mathsf{pwd}} \cdot u^{b^* \cdot \mathsf{pwd}}}{\mathsf{CDH}_g(A^*/u^{\mathsf{pwd}}, B^*/v^{\mathsf{pwd}})}\right)^{1/\mathsf{pwd}^2}.$$

Therefore $\Pr[\mathsf{AskH\text{-}Passive}_4] \leq q_h \times \mathsf{Succ}_g^{\mathsf{cdh}}(T + 4\tau_e)$.

If the corresponding transcript $((\mathcal{C}, B^*), (\mathcal{S}, A^*, (g^{s_i})_i))$ comes from an execution between an instance of $\mathcal{C}$ and the adversary, where at least at one flow is not oracle-generated, it means that B^* has been simulated and the other has been generated by the adversary. We know that either the secret key-corrupt query or the password corrupt query has not been asked (otherwise the simulation was performed using H in game $\mathbf{G}_3$).

- Assume that the secret key-corrupt query has not been made before this session, then x_c and h are unknown to the adversary. Then it is quite hard to compute $h^{s_i} = (g^{s_i})^{x_\mathcal{C}}$ (no information at all): q_h/q.
- If the secret key-corrupt query has been made, it implies that the password-corrupt query has not been made. Due to the games which were canceled in this game, there is at most one password pwd such that there exists an index i, $1 \leq i \leq N$, such that:

$$\mathcal{S} \| \mathcal{C} \| (g^{s_i})_i \| A^* \| B^* \| K_\mathcal{S} \| K_\mathcal{S} \| \mathsf{pwd} \| i$$

  is in Λ_H. In other words, for every transcript, there is only one password which can be tested by the adversary: $\sum_\mathcal{C} \mathcal{D}(q_\mathcal{C})$.

If the corresponding transcript $((\mathcal{C}, B^*), (\mathcal{S}, A^*, (g^{s_i})_i))$ comes from an execution between instances of $\mathcal{S}$ and the adversary, where at least at one flow is not $\mathsf{Compute}$-oracle-generated, it means that $(A^*, (g^{s_i})_i)$ has been simulated and B^* has been generated by the adversary. Since the server accepted a non-$\mathsf{Compute}$-oracle-generated, it means that the biometric corrupt query has been made for the corresponding client $\mathcal{C}$. Thereafter, the same analysis, according to the secret key-corrupt status and the password-corrupt status, as above can be done.

We can thus conclude with

$$\Pr[\mathsf{AskH}_4] \leq \sum_\mathcal{C} \mathcal{D}(q_\mathcal{C}) + \frac{q_h}{q} + q_h \cdot \mathsf{Succ}_g^{\mathsf{cdh}}\ (T + 4\tau_e).$$

□

5 Discussion

5.1 Optimality and Tightness

The authentication probability upper bound presented in Theorem 1 has two leading terms which are

$$q_S \left(\varepsilon_{\mathsf{fa}} + \frac{\binom{\tau}{\tau - t}}{2^{\ell(\tau - t)}} + \frac{N^t \cdot (2^\ell - 1)^t}{2^{\ell N}(t-1)!} \right) \qquad \text{and} \qquad \sum_\mathcal{C} \mathcal{D}(q_\mathcal{C}).$$

If ℓ is large enough, then the last two terms in the parenthesis are negligible, that is why we focus on the two terms which cannot be made negligible even with larger parameters: $q_S \cdot \varepsilon_{\mathsf{fa}}$ and $\sum_\mathcal{C} \mathcal{D}(q_\mathcal{C})$. We claim that our scheme is optimal

and the security result is tight: these two terms could not be avoided, with any better protocol.

Let us consider the following adversary: $\mathcal{A}$ asks for both a password and a secret key corrupt queries and then tries to authenticate using her own biometric. Every time she tries to authenticate, her success probability is equal to the false acceptance probability. Thus, her global success probability is approximately equal to $q_S \cdot \varepsilon_{\mathsf{fa}}$. This attack is generic, independent of any specific protocol, and therefore this shows that the first upper bound cannot be avoided by any cryptographic means.

Secondly, let us consider the adversary which asks for all the secret key-corrupt and (liveness assumption) biometric-corrupt queries, against all the clients: the system is now protected by the passwords only. Thereafter, for each client $\mathcal{C}$, she makes $q_{\mathcal{C}}$ impersonation attempts with the server, using the $q_{\mathcal{C}}$ most probable passwords. For every client $\mathcal{C}$, the success probability is upper-bounded by $\mathcal{D}(q_{\mathcal{C}})$, therefore the global success probability is approximately equal to $\sum_{\mathcal{C}} \mathcal{D}(q_{\mathcal{C}})$ (it shows that the best attack consists in trying the most probable passwords against as many clients as possible). Once again, the adversary is generic and independent of any protocol. Therefore, this bound cannot be avoided either.

The other terms being negligible, our global upper-bound against authentication is tight, and our protocol optimal. The same way, one can show optimality and tightness for the semantic security.

5.2 Practical Parameters

Let us see what it gives with practical values. An iris scan is usually encoded over $N = 1024$ bits and $t = 300$ is considered as a good threshold for the Hamming distance between two measurements of the same biometrics. With such parameters, the false acceptance rate is estimated to 2^{-14}. For a similar false rejection rate, we can assume $\tau = 400$ as a reasonable threshold. In this case, if $\ell \geq 4$ then

$$\frac{\binom{\tau}{\tau-t}}{2^{\ell(\tau-t)}} + \frac{N^t \cdot (2^\ell - 1)^t}{2^{\ell N}(t-1)!} \leq \frac{\binom{400}{100}}{2^{400}} + \frac{2^{3000}}{2^{2896}(299)!} \leq \frac{2^{321}}{2^{400}} + \frac{2^{104}}{2^{2033}} \leq 2^{-78}.$$

Note that ℓ is the length of the authentication tags. The shorter they are, the more efficient the protocol is, from a communication point of view. Can we reduce this value ℓ? Consider an adversary that has corrupted both the password and the secret key. With very high probability (greater than $\varepsilon_{\mathsf{fa}}$) the Hamming distance between a measurement of the adversary biometric and a client reference biometric is approximately 512 and so there are 512 indices i such that $\alpha_i = \alpha'_i$. If $\ell = 1$ there are approximately 512/2 other α_i and α'_i which are equal, that is, there are 768 indices i for which $\alpha_i = \alpha'_i$ and the adversary is able to impersonate the client. Therefore, if $\ell = 1$, with probability greater than the false acceptance probability an adversary can authenticate.

This means that ℓ must be greater than 2, and the previous bound shows that $\ell = 4$ is a good choice. However if one wants to guarantee the correctness of an

honest execution (for all the indices i, $\alpha_i = \alpha'_i$ and $\beta_i = \beta'_i$ if and only if the biometric bits are the same), then a good solution is to choose a greater ℓ. If $\ell = 24$, an honest execution succeeds with probability $2^{-14} \approx \varepsilon_{\mathsf{fa}}$.

Another solution to guarantee the correctness of an honest session is to add a distillation step [10] after the protocol. Distillation allows two entities, with two secret keys with small Hamming distance, to agree on a common secret key, at the price of revealing some of the bits of the original secret keys. With a distillation step, one can choose $\ell = 4$. Even if the resulting secret is shorter than the original ones, this is not a problem in our case, since the original ones are quite large. The distillation step also allows to prevent some denial-of-service attacks where the adversary flips some of the α'_i (this is possible only if the liveness assumption is broken) or β_i. If for this i, $K_i = K'_i$ then with high probability the two entities will generate two different secret keys, whereas they both accept (a few modifications might not flip the decision), and then think that they share the same secret key. With a classical key confirmation, this attack can be detected, and the affected sessions identified. However the advantage of the distillation is that it allows to correct the errors introduced by the adversary or due to hazard, and then to avoid replaying the protocol once more.

5.3 Conclusion

In this paper, we defined a quite strong security model, since it allows a lot of information leakage for the adversary. It guarantees that the adversary has to break all the protections to impersonate a client. Namely, as long as the secret key is not recovered from the secure device, one can show that the success probability of the adversary against our scheme is negligible. As the unclonable device is probably the strongest and the most realistic protection, we can say that our protocol is quite secure.

Acknowledgment

This work has been partially supported by the French RNRT/ANR BACH Project, and the European Commission through the IST Program under Contract IST-2002-507932 ECRYPT.

References

1. Abdalla, M., Fouque, P.-A., Pointcheval, D.: Password-based authenticated key exchange in the three-party setting. In: Vaudenay, S. (ed.) PKC 2005. LNCS, vol. 3386, pp. 65–84. Springer, Heidelberg (2005)
2. Abdalla, M., Pointcheval, D.: Simple password-based encrypted key exchange protocols. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 191–208. Springer, Heidelberg (2005)
3. Barral, C., Coron, J.-S., Naccache, D.: Externalized fingerprint matching. In: Zhang, D., Jain, A.K. (eds.) ICBA 2004. LNCS, vol. 3072, pp. 309–315. Springer, Heidelberg (2004)

4. Bellare, M., Canetti, R., Krawczyk, H.: Modular approach to the design and analysis of key exchange protocols. In: ACM STOC Annual ACM Symposium on Theory of Computing, pp. 419–428. ACM Press, New York (1998)
5. Bellare, M., Desai, A., Jokipii, E., Rogaway, P.: A concrete security treatment of symmetric encryption. In: FOCS Annual Symposium on Foundations of Computer Science, pp. 394–403. IEEE Computer Society Press, Los Alamitos (1997)
6. Bellare, M., Pointcheval, D., Rogaway, P.: Authenticated key exchange secure against dictionary attacks. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 139–155. Springer, Heidelberg (2000)
7. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Ashby, V. (ed.) ACM CCS 1993, pp. 62–73. ACM Press, New York (1993)
8. Bellare, M., Rogaway, P.: Entity authentication and key distribution. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 232–249. Springer, Heidelberg (1994)
9. Bellovin, S.M., Merritt, M.: Encrypted key exchange: Password-based protocols secure against dictionary attacks. In: 1992 IEEE Symposium on Security and Privacy, pp. 72–84. IEEE Computer Society Press, Los Alamitos (May 1992)
10. Bennett, Brassard, Crepeau, Maurer: Generalized privacy amplification. In: ISIT (1994)
11. Bhargav-Spantzel, A., Squicciarini, A.C., Modi, S., Young, M., Bertino, E., Elliot, S.J.: Privacy preserving multi-factor authentication with biometrics. In: Juels, A., Winslett, M., Goto, A. (eds.) Proceedings of ACM DIM 2006 workshop, pp. 63–72. ACM Press, New York (2006)
12. Boyen, X.: Reusable cryptographic fuzzy extractors. In: Atluri, V., Pfitzmann, B., McDaniel, P. (eds.) ACM CCS 2004, pp. 82–91. ACM Press, New York (2004)
13. Boyen, X., Dodis, Y., Katz, J., Ostrovsky, R., Smith, A.: Secure remote authentication using biometric data. In: Cramer, R.J.F. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 147–163. Springer, Heidelberg (2005)
14. Boyko, V., MacKenzie, P.D., Patel, S.: Provably secure password-authenticated key exchange using Diffie-Hellman. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 156–171. Springer, Heidelberg (2000)
15. Brainard, J.G., Juels, A., Rivest, R.L., Szydlo, M., Yung, M.: Fourth-factor authentication: somebody you know. In: ACM Conference on Computer and Communications Security, pp. 168–178 (2006)
16. Canetti, R., Halevi, S., Katz, J., Lindell, Y., MacKenzie, P.D.: Universally composable password-based key exchange. In: Cramer, R.J.F. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 404–421. Springer, Heidelberg (2005)
17. Canetti, R., Krawczyk, H.: Analysis of key-exchange protocols and their use for building secure channels. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 453–474. Springer, Heidelberg (2001)
18. Canetti, R., Krawczyk, H.: Security analysis of IKE's signature-based key-exchange protocol. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 143–161. Springer, Heidelberg (2002), http://eprint.iacr.org/2002/120/
19. Canetti, R., Krawczyk, H.: Universally composable notions of key exchange and secure channels. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 337–351. Springer, Heidelberg (2002)
20. Daugman, J.: How iris recognition works. In: ICIP (1), pp. 33–36 (2002)
21. Daugman, J.: Iris recognition and anti-spoofing countermeasures. In: 7th International Biometrics Conference (2004)

22. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Transactions on Information Theory 22(6), 644–654 (1976)
23. Dodis, Y., Katz, J., Reyzin, L., Smith, A.: Robust fuzzy extractors and authenticated key agreement from close secrets. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 232–250. Springer, Heidelberg (2006)
24. Dodis, Y., Reyzin, L., Smith, A.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 523–540. Springer, Heidelberg (2004)
25. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 10–18. Springer, Heidelberg (1985)
26. Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sciences 28(2), 270–299 (1984)
27. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof-systems. In: Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing (STOC 1985), pp. 291–304 (1985)
28. Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM Journal on Computing 17(2), 281–308 (1988)
29. Jakobsson, M., Myers, S.: Phishing and Counter-Measures. John Wiley and Sons Inc., Chichester (2006)
30. Juels, A., Wattenberg, M.: A fuzzy commitment scheme. In: ACM CCS 1999, November 1999, pp. 28–36. ACM Press, New York (1999)
31. Rackoff, C., Simon, D.R.: Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 433–444. Springer, Heidelberg (1992)
32. Valencia, V.: Biometric Liveness Testing, ch. 8. Osborne McGraw-Hill, New York (2002)