eSketch: a Privacy-Preserving Fuzzy Commitment Scheme for Authentication using Encrypted Biometrics

Pierluigi Failla
Dept. Information Engineering
University of Siena - Italy
pierluigi.failla@gmail.com

Yagiz Sutcu
Polytechnic Institute of New York University
New York - USA
yagiz@isis.poly.edu

Mauro Barni
Dept. Information Engineering
University of Siena - Italy
barni@dii.unisi.it

ABSTRACT

The fuzzy commitment approach has gained popularity as a way to protect biometric data used for identity verification of authentication. As it has been show recently, though, the use of fuzzy commitment is unavoidably linked to some leakage of information regarding the biometric template. An additional problem typical of authentication systems is that the user may want to protect his privacy, that is it would be desirable that the server only verifies whether the biometric template provided by the user is contained within the list of registered users without that the particular identity of the user accessing the system is revealed. The e-sketch protocol proposed in this paper, solves the above two problems by resorting to tools from Multi Party Computation relying on the additively homomorphic property of the underlying cryptosystem (e.e. the Pailler's cryptosystem). The security and the complexity of the proposed protocol are discussed.

Categories and Subject Descriptors

E.1 [**Data Encryption**]: Public key cryptosystems; H.2.0 [**Database Management**]: General—*Security, integrity, and protection*; K.4.1 [**Computers and Society**]: Public Policy Issues—*Privacy*

General Terms

Security, Algorithms

Keywords

Privacy Preserving protocols, Homomorphic Encryption, Biometric Systems, Multi Party Computation,Cryptography

1. INTRODUCTION

Generally speaking a biometric system may serve one of two basic purposes: *authentication/verification* or *identification*. *Authentication* (or *verification*) is the process of positively verifying the identity of a client, device, or other entity in a computer system. *Identification*, on the other hand, is the process of distinguishing an individual from a larger set of individual records by comparing the presented biometric data with all entries in the database [16].

A typical biometric authentication system consists of two *phases*. During the *enrollment phase*, a client (say, Alice) provides her biometric data, from which features are extracted and a *template* is created and stored, either in a central database, or on a mobile device. During the *authentication phase*, a client who claims to be Alice would give her biometric data again, and the same feature extraction algorithm is applied. The result is then compared with the stored template. If they are sufficiently *similar* according to some similarity measure, the client is authenticated.

Biometric features of individuals are tightly bound with their identities. Moreover, they cannot be easily forgotten or lost. Therefore they provide significant potentials in applications where both security and client convenience are needed. However, achieving the desirable level of security and usability is not trivial. The key challenges, from a security point of view, are the difficulty to protect the biometric templates, ensuring revocability and allowing easy matching.

To be more specific, suppose that an online service (think to a remote medical service) is accessible by using the fingerprint reader of a standard notebook. As a registered client, Alice wants to access the service, but does not want to reveal her identity, because for example she is requesting some particular medical diagnosis and she does not want that anybody knows that she needs a specific diagnosis. The server should be able to verify whether Alice's fingerprint corresponds to a registered client, without knowing which particular client is asking to access the service. This request can be summarized in a *motto* sounding like: *everybody is allowed to know that you are registered to a particular service, but no one is able to know when you use it and for which purpose, moreover none is able to distinguish you among the other clients*. Note that in the above scenario privacy preservation is not needed during the enrollment phase. In fact we may assume that when Alice is enrolled she gives a plain version of her biometric. In this phase we can assume that the server is trusted since, for instance, the client is physically present during the enrollment phase.

There has been intensive study on how to secure the biometric templates in recent years and a comprehensive coverage of many proposed solutions can also be found in [12]. These techniques can be roughly categorized into two types: (1) non – invertible transformation – based approaches where similarity of biometric samples would be preserved through the transformation, yet it is difficult to find the original template from a transformed one (e.g., [1, 19]) and (2) methods based on helper-data, where a recently proposed cryptographic primitive, the *secure sketch*, (or a variant of it) is employed, such that given a noisy biometric sample, the original

MM&Sec'10, September 9–10, 2010, Roma, Italy.

biometric data can be recovered with the help of some additional information (i.e., a sketch), which makes it possible to use biometric data in the same way passwords are used. These techniques include [14, 13, 21].

Secure sketch framework does not only allow more rigorous security analysis (in information theoretic sense) compared to many other approaches, but also helps generalizing much of the prior works based on helper-data. Most importantly, a sketch allows exact recovery of the biometric template. Therefore, a *strong extractor* (such as pair-wise independent hash functions) can be further applied on the template to obtain a key that is robust, in the sense that it can be consistently reproduced given any noisy measurement that is similar to the template. However, although it has been shown that there are a few difficulties in extending these techniques to biometric templates in practice, the most important problem is the fact that, the information leakage on the biometric sample is unavoidable when using these schemes [11].

In the last few years, new techniques related to homomorphic encryption showed that it is possible to perform some computations in the encrypted domain in an efficient way and without revealing the information hidden inside the cryptogram (see for example [18] and [3].) Following that direction, researchers developed many protocols to be applied in applications where the privacy and the security of the inputs are crucial. Some applications includes, face recognition [9], ElectroCardioGram (ECG) classification [4], data mining [15] and watermarking [20].

An application of the above techniques to the biometric verification problem has been proposed in [6] where the biometric data stay encrypted during all the computations thanks to the integration of secure sketches into homomorphic cryptosystems. Moreover, confidentiality of requests made to the database is also obtained thanks to a Private Information Retrieval (PIR) protocol. In particular [6] uses the fuzzy commitment scheme described in [14], and solves the correcting code problem by using a linear correcting code implementable using Goldwasser-Micali cryptosystem [10].

As another proposal, Upmanyu et al. in [23] has developed an efficient protocol for biometric verification based on asymmetric cryptosystem (RSA). More specifically, in order to achieve a secure and efficient verification, a linear classifier is used. However, it is highly probable that the same solution using the Paillier cryptosystem would be much more efficient. Moreover, RSA is not semantically secure and due to the structure of the scheme, the client identity is disclosed.

In [7] and [2], searchable encryption techniques [5] are used for building a secure protocol that is able to identify subjects using encrypted biometric data. Specifically in [7], the main objective is to solve a problem of *identification* using an encrypted database with Private Information Storage (PIS) and Private Information Retrieval (PIR). Similarly, in [22], a secure two-party computation technique is proposed to find the legitimate owner of a query biometric data without revealing any sensitive information to any party.

In this paper, we propose a simple authentication scheme based on fuzzy commitment scheme [14] which makes possible to perform all operations in encrypted domain. In addition to ensuring the security of the biometric data that is always managed in encrypted format, and the revocability of the biometric template, the proposed scheme is also capable of protecting the privacy of the client that is going to be authenticated. The proposed scheme addresses the above scenario wherein a client entitled to access a given service is asked to provide her biometric data for accessing the service. Our protocol permits to verify whether a client is included in a list of registered clients without that the server is able to track which client accessed the system and when.

The rest of this paper is organized as follows. In Section 2 the fuzzy commitment scheme is summarized. Section 3 is devoted to introduce the notation we will use while Section 4 introduces the building blocks we need in our construction. Finally in Section 5 our protocol is presented, paying attention also to the question related with security and computational complexities. In Section 6 conclusion and future work are discussed.

2. FUZZY SKETCH IN A NUTSHELL

The Fuzzy Commitment Scheme as proposed in [14] is a technique that combines well-known approaches in the areas of Error Correcting Codes (ECC) and cryptography to reach the goal of an efficient commitment scheme. Formally speaking, an ECC is a set of codewords $\mathcal{C} \subseteq \{0,1\}^n$ selected for mapping the information. Therefore, for a message space of size 2^k we need at least $n = k$, but to achieve redundancy, in general, we require that $n > k$. Given the message space $\mathcal{M} = \{0,1\}^k$, we define $g : \mathcal{M} \rightarrow \mathcal{C}$ as the *translation function* (sometimes called coding function), thus g is a map from $\mathcal{M}$ to $\mathcal{C}$. Conversely g^{-1} is the inverse map from $\mathcal{C}$ to $\mathcal{M}$. The function f is the *decoding function* $f : \{0,1\}^n \rightarrow \mathcal{C}$ that maps arbitrary n-bit strings to the nearest codeword in $\mathcal{C}$. We say that f has a correction capability of t if it can correct up to t bit errors.

In the fuzzy commitment scheme, biometric data are treated as a corrupted codeword. Therefore, we use only the decoding function to reconstruct the right associated codeword and we do not care about g and g^{-1} functions. A fuzzy commitment scheme F works on codewords c and binary vectors x where both are strings of length n-bit. In particular for any given x and codeword c, we can express x uniquely by means of the codeword c and an offset δ ($x = c \oplus \delta$). It is simple to show that the information of x contained in δ depends on the cardinality of $\mathcal{C}$ [1].

The original fuzzy commitment scheme in [14] works as follows. During the enrollment phase, the client presents a biometric data x and the server chooses a codeword c. At this point the server stores, for that client, the pair $(\delta, Hash(c))$ where: $\delta = x \oplus c$ and $Hash(c)$ is the hash of the codeword c. During the matching phase a new noisy biometric data $\widehat{x}$ is presented by a client who claimed his identity, the server computes $\widehat{c} = \widehat{x} \oplus \delta$ and also $Hash(f(\widehat{c}))$. If $Hash(c) = Hash(f(\widehat{c}))$ than the client is authenticated. In case of identification, the basic scheme outlined above is repeated for all registered clients, resulting in a 1 to M matching request (M is the total number of enrolled clients).

As shown in [11] the simple sketch approach described above suffers from a leakage of information that cannot be avoided with standard algorithms. Therefore, in case of the non-trusted parties, the protocol should be secure, in the sense that, after running the protocol, neither the server nor the client obtain any information beside the output of the protocol. The encrypted-sketch (e-sketch) scheme described in the following sections prevents the information leakage and provides an efficient solution to the user privacy along with template security.

3. NOTATION AND PRELIMINARIES

In the rest of the paper we will use the following notation:

- $x \in \{0,1\}^n$ is the biometric data consisting of a binary string of length n. We indicate with x_i the i-th bit of the string;

[1]Please note that the binarization of raw biometric data is out of the scope of this paper.

- with $\overline{a}$ we refer to the bit-wise representation of a;
- c is a codeword in the set $\mathcal{C}$;
- with $[\![a]\!]$ we indicate the Paillier [17] encryption of a;with $[\![\overline{a}]\!]$ the bitwise encryption of a. Sometimes we indicate with $[\![a]\!]_i$ the encryption of a with the key of the client i;
- PuK and PrK are respectively the public key and the private key of the cryptosystem adopted in the protocol;
- s is the cryptosystem security parameter (i.e. short term security 1024 bit) and ℓ is the bit size of a cryptogram[2].

We recall that the following basic mapping holds for Paillier's cryptosystem: $[\![x]\!][\![y]\!] = [\![x+y]\!]$ and $[\![x]\!]^y = [\![xy]\!]$.

Moreover we recall the Big-$\mathcal{O}$ notation that measures the computational complexity in bit operations, for instance considering numbers of at most ℓ bits we have $add = \mathcal{O}(\ell)$ or $mult = \mathcal{O}(\ell^2)$ and $exp = \mathcal{O}(\ell^3)$. In particular we consider also that for Paillier cryptosystem $enc = dec = \mathcal{O}(\ell^3)$.

4. BASIC BUILDING BLOCKS

In this section we introduce the basic building blocks that we will use to construct the e-sketch protocol.

4.1 Computing XOR in the Encrypted Domain

By assuming that x and y are binary values, computing the XOR function is equivalent to the following:

$$x \oplus y = x + y - 2xy \tag{1}$$

which can be used to compute the XOR function in the encrypted domain. Therefore we consider two main cases: **i)** x is encrypted and y is not and **ii)** both x and y are encrypted.

Computing $[\![x \oplus y]\!]$ from $[\![x]\!]$ and y.

Due to the additive homomorphic properties of Pailler's cryptosystem, it is possible to rewrite equation (1) as follows:

$$[\![x \oplus y]\!] = [\![x]\!][\![y]\!][\![x]\!]^{-2y}, \tag{2}$$

The computational complexity is mainly the cost to compute the modular inversion (i.e. $x^{-1} \bmod n$) that requires the same complexity of an exponentiation, so it is $2\ mult + 1\ exp \simeq 1\ exp$.

In this case the bandwidth requirement is 0 since there is no interaction between the parties. Thus, the round complexity is also 0.

Computing $[\![x \oplus y]\!]$ from $[\![x]\!]$ and $[\![y]\!]$.

We now suppose that both bit values are available in encrypted format, i.e. the server knows $[\![x]\!]$ and $[\![y]\!]$, where encryption is carried out by using the client's PuK. The server does not want to reveal neither x nor y to the client, so it chooses two additional random bits r_x and r_y and uses equation (2) to compute $[\![x \oplus r_x]\!]$ and $[\![y \oplus r_y]\!]$ then sends them to the client. Note that x and y are perfectly obfuscated by the xor-ing with r_x and r_y, so the client can decrypt them, compute the encryption of $[\![(x \oplus r_x) \oplus (y \oplus r_y)]\!]$ and send the result back to the server. At this point the server using again equation (2) can remove r_x and r_y from the result and obtain $[\![x \oplus y]\!]$.

Since the server needs to compute the XOR function by using equation (2) four times, the client computes two decryptions and one encryption, the complexity is $4\ exp + 2\ dec + 1\ enc \simeq 7\ exp$.

This sub-protocol requires a bandwidth of 3ℓ because the server sends two cryptograms to the client that responds with one cryptogram. The round complexity is 2.

[2]Using the Paillier cryptosystem we have the following equality: $\ell = 2s$.

4.2 Decoding in the encrypted domain

A key step in the fuzzy commitment scheme is the search for the codeword in $\mathcal{C}$ that is closest to $\widehat{c}$, i.e. the computation of $f(\widehat{c})$. In this subsection we present a protocol to compute such a function when $\widehat{c}$ is available in encrypted form to the client. We will refer to such a protocol as eSearch functionality. The approach that we will follow is to delegate the computation of f to the client in a such way that the client is not able to understand which are the input and the output of the computation. The details of the ECC code are supposed to be public.

To describe the eSearch protocol we start by assuming that the space $\mathcal{C}$ of all the codewords is a linear subspace that is closed under bitwise XOR operation[3]. The following property holds.

PROPERTY 1. *We have $f(\widehat{c} \oplus d) \oplus d = f(\widehat{c})$, $\forall d = c_j \in \mathcal{C}$.*

PROOF. Let $c_i = f(\widehat{c})$. We surely have $\widehat{c} = c_i \oplus \varepsilon$ for some ε. We have:

$$f(c_i \oplus \varepsilon \oplus d) \oplus d = f(c_i \oplus \varepsilon \oplus c_j) \oplus c_j = c_i \tag{3}$$

where we have assumed that the decoding function is able to correct the error ε whatever codeword ε is added to, and where due to the linearity assumption the addition of two codewords always results in a valid codeword. □

Thank to the above result, a very simple eSearch protocol can be obtained: the server blinds $\widehat{c}$ by adding to it a random codeword d, then it asks the client to decode the blinded message. The client evaluates f in the plain domain, re-encrypts the result and sends it back to the server, that can obtain the encrypted version of the decoded codeword by XOR-ing back the result with d. A more detailed description of the eSearch protocol outlined so far is given below and depicted in Fig. 1. Note that all codewords are encrypted sample wise so to allow the application of the first of the two secure XOR protocols described in the previous section.

Protocol 1 eSearch

1: The Server knows a noisy encrypted codeword $[\![\overline{\widehat{c}}]\!]$
2: The Server chooses a random codeword c_j and by homomorphic properties computes $[\![\overline{\widehat{c}} \oplus \overline{c_j}]\!]$
3: $\leftarrow$ The Server sends to the Client $[\![\overline{\widehat{c}} \oplus \overline{c_j}]\!]$
4: The Client decrypts and finds: $\overline{\widehat{c}} \oplus \overline{c_j}$
5: The Client applies the decoding function: $f(\overline{\widehat{c}} \oplus \overline{c_j}) = \overline{c_i} \oplus \overline{c_j}$
6: $\rightarrow$ The Client computes $[\![\overline{c_i} \oplus \overline{c_j}]\!]$ and sends it back to the Server
7: The Server computes $[\![\overline{c_i}]\!] = [\![\overline{c_i} \oplus \overline{c_j} \oplus \overline{c_i}]\!]$

Security discussion. In the following we argue that, under the assumption that the client is allowed to know ε, eSearch is secure in the *honest but curious* model [8]. This is a reasonable assumption since ε reveals only the error between the enrolled biometric data and the new one, and the error between two biometric measurements can be assumed to be uncorrelated to the biometric value itself. Furthermore this information is revealed only to the client that owns the biometric data. So, while ε can be seen as a leakage of information, this leakage is seen by the client only. This can not be considered to be a problem since the sensible information that needs to be protect is the codeword that the server does not want to reveal. More specifically, the eSearch protocol achieves both client and server privacy, in fact during the whole protocol the

[3]This is always the case with the most common ECC.

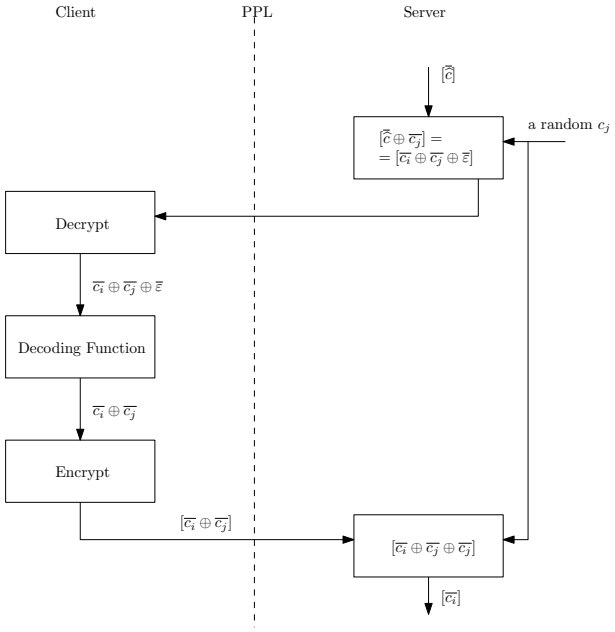

Figure 1: eSearch.

server sees only encrypted data, from which it can not get any information due to IND-CPA security of the underlying cryptosystem. Considering the server privacy note that an eavesdropper can not get any information from the encrypted values due to the IND-CPA security of Pailler's cryptosystem. As to the client, he is only able to know ε and the blinded codeword message $\widehat{c} + c_j$. We already discussed why disclosing ε is not a problem. As to the blinded message it corresponds to $c_i + \varepsilon + c_j$. Since the client knows ε this is equivalent to knowing $c_i + c_j$. If the server chooses c_j randomly and uniformly over all possible codewords in $\mathcal{C}$, then it is easy to show that the mutual information $I(c_i; c_i \oplus c_j)$ is equal to zero, hence proving the server privacy of the protocol.

Complexity. The most expensive operation in Protocol 1 is computing XOR in Step 2 and 7, by this, the computational complexity is dominated by: $2\,exp$ and n encryptions (we recall that the codewords are bitwise encrypted), so: $2\,exp + n\,enc \simeq (n+2)\,exp$. The bandwidth is exactly $2n\ell$ because just 2 blocks of n cryptograms are transmitted. Finally, 2 rounds are needed to run eSearch.

5. THE ESKETCH PROTOCOL

We are now ready to describe the overall eSketch protocol for privacy preserving authentication. In the rest of this work we suppose that there are M registered clients, moreover we consider that all the values involved in the protocol are bitwise encrypted so for the sake of simplicity we omit the notation $[\![\overline{x}]\!]$ and we will use just $[\![x]\!]$.

Enrollment. Let us start by considering the enrollment phase for a generic client j. The j-th Client sends the plain version of his biometric data x_j to be enrolled in the system, moreover he sends also an encrypted obfuscated version $[\![x_j \oplus R_j]\!]_j$, where R_j is a random blinding factor chosen by the Client. The Server chooses a codeword c, computes $\delta_j = x_j \oplus c$ and stores the pair δ_j and $[\![x_j \oplus R_j]\!]_j$. Protocol 2 and Fig. 2 show those steps. As we already said, in this phase we assume that the client trusts the server. This is possible, for instance, because the client goes physically to the server to perform the enrollment, and control himself that the Server destroys any sensible information.

Protocol 2 E-sketch protocol: enrollment

1: $\rightarrow$ The Client sends x_j and $[\![x_j \oplus R_j]\!]_j$
2: The Server chooses a codeword c and computes $\delta_j = x_j \oplus c$
3: The Server stores δ_j and $[\![x_j \oplus R_j]\!]_j$

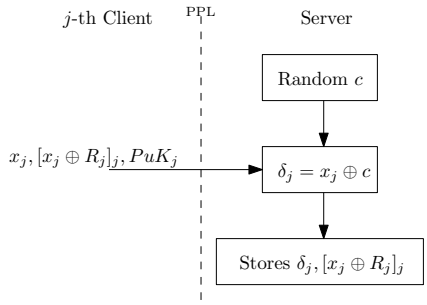

Figure 2: eSketch – Enrollment.

Upon presentation of a new noisy biometric data $\widehat{x_j}$ from the client, the server must check whether the this biometric data corresponds to one of the M enrolled users. In this phase, the client wants that nothing is disclosed about the noisy biometry $\widehat{x_j}$. At the same time the server wants that nothing is revealed about the biometric data of the other users and must avoid that a non-registered user results in a positive match. At the end of the protocol, the server will know only if the user trying to access the system is a registered user, but will not know which user is accessing the system. The above goals are obtained by means of the following protocol.

Matching. In our description we refer to Fig. 3 and Protocol 3. Let us assume that the j-th client wants to access the system. He sends $[\![\widehat{x_j}]\!]_j$ and his PuK_j to the server. Note that in our framework revealing PuK_j does not reveal the identity of the client. The reason for this is the in our set up PuK_j and PrK_j are generated directly by the client during the enrollment with no intervention of a certification authority, so there is nobody that would be able to associate a given PuK_j to the particular j-user. Actually the server could be able to trace the behavior of the clients by keeping trace of the usage of the M PuK's of the clients. This could be a problem for small values of M since it could be possible to trace back to the identity of the client from his behavior. However, for large values of M as those typically encountered in on-line services, this is unlikely to be a problem. On the contrary, the possibility of tracking users's behavior collectively without that a particular behavior is associated to a given user could be seen as an advantage of the e-sketch protocol. In any case, to prevent this kind of attack, all the client may be asked to re-enroll with a new PuK regularly, depending on the application.

Since the user did not claim his identity the server cannot index the database for a given client. For this reason, for each entry in the database, the server computes $[\![\widehat{c_i}]\!]_j = [\![\widehat{x_j} \oplus \delta_i]\!]_j$ (he can do that by exploiting the homomorphic property of the cryptosystem as in equation (2) obtaining M noisy codewords each one encrypted with the j-th client's PuK). At this point the server and the client run the eSearch protocol M times to obtain M denoised codewords ($[\![c'_i]\!]_j$), then the server XOR's each of them with δ_i. In this way he obtains a set of M enrolled encrypted *probable*-biometrics: $[\![x'_i]\!]_j$. For each entry in the database, the server has also stored $[\![x_i \oplus R_i]\!]_i$ so he can compute $[\![W_i]\!]_j = [\![x_i \oplus R_i \oplus x'_i \oplus R]\!]_j$, where R is an additional random number chosen by the server. Note that only if

$i = j$ the homomorphic property make sense (this is due to the standard properties of IND-CPA cryptosystems), in all the other cases the result of this operation is simply a random string of numbers. In addition only if there is one $x_i = x'_i$ the j-th Client can be authenticated. To do so the server sends all the $[\![W_i]\!]_j$ values $(i = 1, M)$ to the client. The client decrypts them and subtracts to each the value R_j he used in the enrollment phase. Then he scrambles over i (to obfuscate the matching position to the Server) and sends the results back to the Server. The Server removes the blinding factor R homomorphically and checks if in the list he obtained there is a 0 vector. If this is the case, access is granted.

Protocol 3 E-sketch protocol: Matching

1: $\rightarrow$ The Client sends $[\![\widehat{x_j}]\!]_j$ and PuK
2: the Server computes the noisy codewords $[\![\widehat{c_i}]\!]_j = [\![x_j \oplus \delta_i]\!]_j \ \forall i \in [1, M]$
3: The server and the client run the eSearch protocol for each entry $i \in [1, M]$ at the end the server obtains a set of M codewords: $[\![c'_i]\!]_j$
4: The Server computes $[\![x'_i]\!]_j = [\![c'_i \oplus \delta_i]\!]_j \ \forall i \in [1, M]$
5: For each i the Server adds $[\![x'_i]\!]_j$ to $[\![x_i \oplus R_i]\!]_i$ and adds a random R: $[\![W_i]\!]_j = [\![x_i \oplus R_i \oplus x'_i \oplus R]\!]_i \ \forall i \in [1, M]$
6: $\leftarrow$ The Server sends the values $[\![W_i]\!]_j$ to the client
7: The Client decrypts all $[\![W_i]\!]_j$ and for each removes its own R_j
8: $\rightarrow$ The Client sends to the server the scrambled decryptions
9: The Server removes his R for each i
10: The Serve checks if in the list there is a 0 if it is so the client can access the service

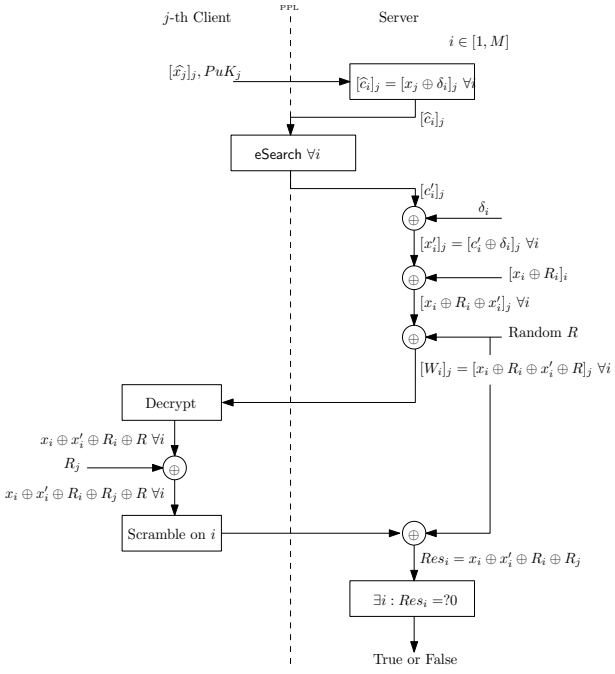

Figure 3: eSketch – Matching.

5.1 Security

To discuss the security of the e-sketch protocol we observe that with respect to the XOR and e-search protocols (that we already proved to be secure) the only additional steps in which we may have some leakage of information are steps 6 through 9. To see that no leakage of information occurs during these steps let us consider the two case: $i = j$ and $i \neq j$. In the first case the Client *sees*: $x_j \oplus R_j \oplus x'_j \oplus R \oplus R_j$ if the biometric is not too noisy[4] we have that $x_j = x'_j$ and so the above computation returns only R that is a random value chosen by the Server, and so no leakage of information occurs here. When $i \neq j$ the Client receives: $[\![W_i]\!]_j = [\![x_i \oplus R_i \oplus x'_i \oplus R]\!]_j$, when he applies the decryption function $\mathcal{D}_j$ using his PrK, he obtains something that is completely random, since part of the cryptograms is encrypted with a different PuK and so the decryption is completely meaningless.

After that the Client subtracts R_j and sends back to server $x_i \oplus R_i \oplus x'_i \oplus R \oplus R_j$. The server removes R and obtains $x_i \oplus R_i \oplus x'_i \oplus R_j$ that is a completely random number. The server, then, *sees* a string composed by random numbers and, possibly, a zero in a random position, hence non leakage of information occurs on his side well. Finally we observe that if someone tries to access the system without knowing the correct keys, he only sees random string values due to the security of the underlying cryptosystem.

Table 1: Protocol eSearch Complexities

	# Exp	Bandwidth	Round
Enrollment	n	$(n + 2.5)\ell$	1
Matching	$(n + 4)M + n$	$(3.5Mn + n + 2)\ell$	$3 + 2M$

5.2 Complexity

We now briefly discuss the complexity of the e-sketch protocol. In doing so we focus on the most expensive operations. During the *enrollment* phase the computational complexity is:

$$n\ enc + 1\ add \simeq n\ exp$$

with just 1 round.

The *matching* phase is much more complex and requires:

$$n\ enc + 4M\ mult + 2M\ add + M\ exp + \underbrace{M(n+2)\ exp}_{M\ eSearch} + M\ dec \quad (4)$$

that is dominated by $(Mn + 4M + n)\ exp$. Moreover in the matching phase 3 rounds are needed plus those needed to compute M eSearch, for a total of $3 + 2M$ rounds.

Bandwidth. The enrollment phase requires a transfer of 1 plain (we recall that the plaintext size is $\frac{\ell}{2}$ bits), n encrypted values and the PuK so: $\frac{\ell}{2} + n\ell + 2\ell = (n + 2.5)\ell$ bits while the matching phase:

$$2\ell + n\ell + 2nM\ell + nM\ell + nM0.5\ell = (3.5M + 1)n\ell + 2\ell \quad (5)$$

bits.

Table 1 shows a summary of the complexities involved in the protocol.

By considering asymptotic complexities we have $\mathcal{O}(n\ell^3)$ for the enrollment and $\mathcal{O}(Mn\ell^3)$ for the matching; similarly the asymptotic bandwidth required is $\mathcal{O}(n\ell)$ in the enrollment phase and $\mathcal{O}(Mn\ell)$ in the matching phase. In the end, the protocol requires $\mathcal{O}(1)$ rounds in enrollment and $\mathcal{O}(M)$ for the matching.

6. CONCLUSION

We have presented a protocol that relies on the additive homomorphic properties of the underlying cryptosytem (e.g. Pailler's cryptosystem) which prevents the leakage of information unavoidable in the the fuzzy commitment scheme. The proposed protocol

[4]This is an assumption that must hold if we want that the whole fuzzy sketch approach works.

successfully addresses also the problem of keeping the identity of the owner of the biometric data secret and the biometric data itself. This is a powerful property that makes the proposed scheme suitable for protecting the user privacy by allowing them to be authenticated anonymously. We have also provided an outline of the security proof for the proposed protocol under the semi-honest model, and by assuming that the error correction code employed in the fuzzy commitment scheme satisfies certain, rather common, properties.

Considering just the matching phase, which is the most complex part, the computational complexity of the protocol is linear in both; the number of entries in the database (M) and length of template representation (n). Moreover, also the bandwidth required to compute the protocol depends linearly on the number of entries and the template size. Finally the number of rounds depends just on the number of enrolled clients. Hence, making the proposed protocol efficient and secure.

Investigation about real world implementations and study about fuzzy commitment schemes for verification are topics of future work.

Acknowledgment

This work was partially sponsored by MIUR (Italian Ministry of Education and Research) under project PrivWare (contract n. 2007 JXH7ET).

7. REFERENCES

[1] R. Ang, R. Safavi-Naini, and L. McAven. Cancelable key-based fingerprint templates. In *ACISP*, volume 3574 of *LNCS*, pages 242 – 252, 2005.

[2] M. Barbosa, T. Brouard, S. Cauchie, and S. Sousa. Secure biometric authentication with improved accuracy. *Proceedings of the 13th Australasian conference on Information Security and Privacy*, 5107:21 – 36, 2008.

[3] M. Barni. Processing encrypted signals: a new frontier for multimedia security. In *Proceedings of the 8th workshop on Multimedia and security*, page 1. ACM, 2006.

[4] M. Barni, P. Failla, V. Kolensikov, R. Lazzeretti, A. Paus, A. Sadeghi, and T. Schneider. Efficient Privacy-Preserving Classification of ECG Signals. In *Workshop on Information Forensics and Security 2009*, 2009.

[5] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search. *Lecture notes in computer science*, pages 506 – 522, 2004.

[6] J. Bringer and H. Chabanne. An authentication protocol with encrypted biometric data. *Lecture Notes in Computer Science*, 5023:109 – 124, 2008.

[7] J. Bringer, H. Chabanne, and B. Kindarji. Identification with Encrypted Biometric Data Made Feasible. *Arxiv preprint arXiv:0901.1062*, 2009.

[8] R. Cramer. Introduction to Secure Computation. *Lectures on data security: modern cryptology in theory and practice*, 1561:16 – 62, 1999.

[9] Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Lagendijk, and T. Toft. Privacy-preserving face recognition. In *Privacy Enhancing Technologies*, pages 235 – 253. Springer, 2009.

[10] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270 – 299, 1984.

[11] T. Ignatenko and F. Willems. On Privacy in Secure Biometric Authentication Systems. In *IEEE International Conference on Acoustics, Speech and Signal Processing, 2007. ICASSP 2007*, volume 2, pages 121 – 124, 2007.

[12] A. Jain, K. Nandakumar, and A. Nagar. Biometric template security. *EURASIP Journal on Advances in Signal Processing, Special Issue on Pattern Recognition Methods for Biometrics*, 2008(11):1 – 17, 2008.

[13] A. Juels and M. Sudan. A fuzzy vault scheme. *Designs, Codes and Cryptography*, 38(2):237 – 257, 2006.

[14] A. Juels and M. Wattenberg. A fuzzy commitment scheme. In *Proceedings of the 6th ACM conference on Computer and communications security*, pages 28 – 36. ACM New York, NY, USA, 1999.

[15] Y. Lindell and B. Pinkas. Privacy preserving data mining. *Journal of cryptology*, 15(3):177 – 206, 2008.

[16] L. O'Gorman. Comparing passwords, tokens, and biometrics for user authentication. *Proceedings of the IEEE*, 91(12):2021 – 2040, December 2003.

[17] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. *Advances in Cryptology EUROCRYPT 1999*, pages 223 – 238, 1999.

[18] A. Piva and S. Katzenbeisser. Signal Processing in the Encrypted Domain. *EURASIP Journal on Information Security*, 2007.

[19] N. K. Ratha, S. Chikkerur, J. H. Connell, and R. M. Bolle. Generating cancelable fingerprint templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):561 – 572, 2007.

[20] M. Shao. A privacy-preserving buyer-seller watermarking protocol with semi-trust third party. *Lecture Notes in Computer Science*, 4657:44, 2007.

[21] Y. Sutcu, Q. Li, and N. Memon. Protecting biometric templates with sketch: Theory and practice. *IEEE Transactions on Information Forensics and Security*, 2(3):503 – 512, September 2007.

[22] B. Sy. Secure Computation for Privacy Preserving Biometric Data Retrieval and Authentication. In *Proceedings of the 1st European Conference on Intelligence and Security Informatics*, pages 143 – 154. Springer, 2008.

[23] M. Upmanyu, A. Namboodiri, K. Srinathan, and C. Jawahar. Efficient Biometric Verification in Encrypted Domain. *Advances in Biometrics*, pages 899 – 908, 2009.