VARIABLE-LENGTH TEMPLATE PROTECTION BASED ON HOMOMORPHIC ENCRYPTION WITH APPLICATION TO SIGNATURE BIOMETRICS

Marta Gomez-Barrero, Julian Fierrez

Universidad Autonoma de Madrid
Biometric Recognition Group - ATVS

Javier Galbally

European Commission - Joint Research Centre
Inst. for the Protection and Security of the Citizen

ABSTRACT

Any privacy leakage of biometric data poses severe security risks given their sensitive nature. Biometric templates should thus be protected, storing irreversibly transformed or encrypted biometric signals, while preserving the unprotected system's performance. Based on the recent developments by Zhu *et al.* on privacy preserving similarity evaluation of time series data, we present a new biometric template protection scheme based on homomorphic probabilistic encryption, where only encrypted data is stored or exchanged. We then apply the proposed scheme to signature verification and show that all requirements described in the ISO/IEC 24745 standard are met with no performance degradation, using a publicly available database and a free implementation of the Paillier cryptosystem. Moreover, the proposed approach is robust to hill-climbing attacks.

Index Terms— Homomorphic Encryption, Biometric template protection, On-line signature, DTW, Privacy.

1. INTRODUCTION

Among the different biometric traits used in automatic verification systems, one of the most widely spread is the handwritten signature, due to its traditional legal and social acceptance. Like any other biometric data, any information leakage resulting from an inappropriate storage of the derived templates can lead to severe privacy issues. Templates must be therefore protected, so that such a leakage is prevented.

In accordance with the ISO/IEC 24745 standard on biometric information protection [1], biometric template protection (BTP) systems must fulfil three main requirements: i) irreversibility (i.e., no biometric information should be leaked by the template), ii) unlinkability (i.e., given two templates protected with different keys, it should not be feasible to decide whether they belong to the same subject) and iii) renewability (i.e., if one template is lost or stolen, a new one, not matching the old template, should be issued). At the same time, verification performance of encrypted templates should be maintained compared to the unprotected data.

Most previous approaches to BTPs for on-line signature rely on irreversible transformations that obscure the extracted features, thus resulting in some performance degradation. In the BioConvolving scheme [2], the original Hidden Markov Model (HMM) is trained with irreversibly transformed time sequences, thus degrading verification performance. In [3, 4], fuzzy commitment is applied to on-line signature verification. However, several works have proven that fuzzy commitment schemes are vulnerable to several attacks that compromise the privacy of the subject [5].

Regarding fixed-length templates, [6] proposes a biometric cryptosystem based on hashes generated from the templates, and a BCH error correcting code and helper data are used in the BTP scheme presented in [7]. Even though not a big loss on performance is observed, function based approaches usually yield higher verification accuracy for on-line signatures.

As an alternative to those approaches, Homomorphic Encryption (HE) schemes allow for computations to be performed on ciphertexts, generating encrypted results which decrypt to plaintexts that match the result of the operations carried out on the plaintext. Therefore, combining such an encryption approach with biometric verification systems would meet the aforementioned requirements while preserving biometric performance. HE schemes, which only allow a limited subset of operations on the encrypted domain, are nowadays being introduced into many applications based on signal processing, and, particularly, biometrics. In [8] an iris identification scheme based on HE and k-Anonymous Quantization is presented. In [9], the authors propose a scheme based on a fixed-length representation of fingerprints and HE.

In the present article we propose a new BTP system which uses encrypted templates of non-fixed length, based on the Paillier's homomorphic probabilistic encryption scheme [10] and a state-of-the-art local function-based on-line signature verification system [11]. In order to compare the varying length signature functions in the encrypted domain, the protocol presented in [12] for the efficient computation of the Dymanic Time Warping (DTW) algorithm within HE schemes is adapted. To the best of our knowledge, this is the first BTP system based on HE that handles encrypted templates of variable length, carrying out all the computations in the encrypted domain, and the first time HE is applied to signature.

978-1-4673-9448-2/16/$31.00

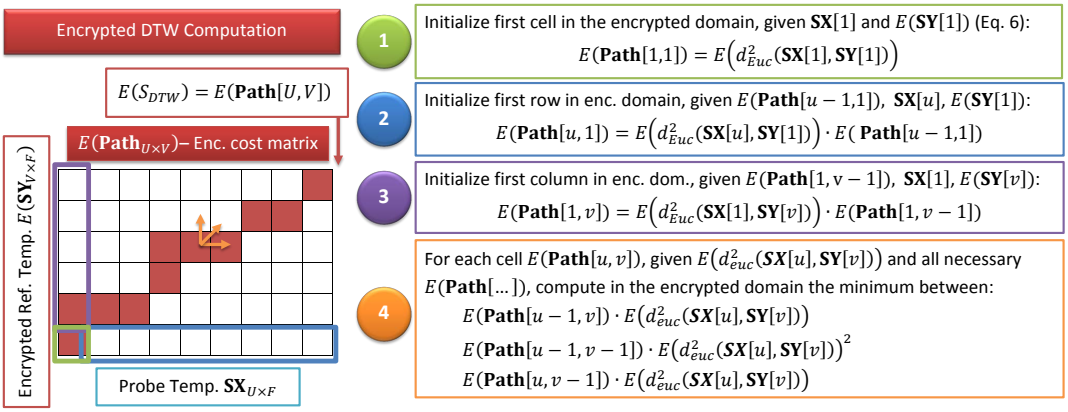

Fig. 1: **Encrypted DTW-based verification**. In order to compare the probe $\mathbf{SX}_{U\times F}$ and the encrypted reference $E\left(\mathbf{SY}_{V\times F}\right)$ signatures, the optimal path, depicted in red, minimizing the Euclidean distance between points, is computed following the DTW algorithm. An encrypted cost matrix, $E\left(\mathbf{Path}\right)$ is built in four steps. The last entry of the matrix contains the final score $E\left(S_{DTW}\right)$. It should be noted that all computations are carried out in the encrypted domain.

The security provided and the computational complexity of the proposed approach, as well as the verification performance, are studied on a reproducible research framework: the publicly available multimodal BiosecurID database [13] and a free implementation of the Paillier cryptosystem are used.

The rest of the article is organized as follows: the proposed system is described in Sect 2. Verification performance is evaluated in Sect. 3, while a security evaluation is carried out in Sect. 4. Computational complexity is analysed in Sect. 5 and final conclusions are drawn in Sect. 6.

2. PROPOSED SYSTEM

In the rest of the paper, we will use the following notation. Signature templates will be denoted as matrices starting with $\mathbf{S}$, such as $\mathbf{SX}_{U\times F}$, where U is the number of points sampled from the biometric sample and F the number of features extracted from each point. The u-th point of the sample is an F-dimensional vector $\mathbf{SX}[u] = \mathbf{x}^u = \{x_1^u, \dots, x_F^u\}$. To simplify notation, to refer to *any* generic point we will use $\mathbf{x} = \{x_1, \dots, x_F\}$. The Euclidean distance between two points $\mathbf{x}$ and $\mathbf{y}$ is denoted as $d_{Euc}\left(\mathbf{x}, \mathbf{y}\right)$. Finally, m denotes a plain message and m^* its corresponding ciphertext: $m^* = E_{pk}\left(m, s\right)$, where E denotes the Encryption function, s a random number and pk the public key. Similarly, $m = D_{sk}\left(m^*\right)$, where sk is the private key and D the Decryption function.

The proposed system is based on the combination of a state-of-the-art on-line signature verification scheme, based on the comparison of local dynamic functions [11] with the Dynamic Time Warping (DTW) algorithm [14] (see Sect. 2.1), and the comparison of signals in the encrypted domain presented in [12] (see Sects. 2.2 and 2.3).

2.1. DTW-Based Verification

In the *unencrypted* signature verification scheme selected in the present work, a subset of $F = 9$ time sequences selected using the Sequential Forward Floating Selection (SFFS) algorithm from the total set of functions defined in [11], is directly compared using DTW [14]. Those time sequences include, for instance, the horizontal and vertical coordinates, the speed or the pressure.

This way, in order to obtain a dissimilarity score between the probe $\left(\mathbf{SX}_{U\times F}\right)$ and the reference templates $\left(\mathbf{SY}_{V\times F}\right)$, a cost matrix $\left(\mathbf{Path}_{U\times V}\right)$, minimizing the distance between signature points in terms of their Euclidean distance, is computed. To that end, four steps are carried out:

1. Initialize first cell:
$$\mathbf{Path}[1,1] = d^2_{Euc}\left(\mathbf{SX}[1], \mathbf{SY}[1]\right)$$
2. Initialize first row:
$$\mathbf{Path}[u,1] = d^2_{Euc}\left(\mathbf{SX}[u], \mathbf{SY}[1]\right) + \mathbf{Path}[u-1,1]$$
3. Initialize first column:
$$\mathbf{Path}[1,v] = d^2_{Euc}\left(\mathbf{SX}[1], \mathbf{SY}[v]\right) + \mathbf{Path}[1,v-1]$$
4. For each of the remaining cells, $\mathbf{Path}[u,v]$ is defined as the minimum between three options:
$$\begin{gathered}\mathbf{Path}[u-1,v-1] + 2\cdot d^2_{Euc}\left(\mathbf{SX}[u], \mathbf{SY}[v]\right)\\ \mathbf{Path}[u-1,v] + d^2_{Euc}\left(\mathbf{SX}[u], \mathbf{SY}[v]\right)\\ \mathbf{Path}[u,v-1] + d^2_{Euc}\left(\mathbf{SX}[u], \mathbf{SY}[v]\right)\end{gathered}$$

The final dissimilarity score is the last cell of the matrix, namely $S_{DTW} = \mathbf{Path}[U,V]$.

For a more detailed description of the use of DTW for on-line signature verification the interested reader is referred to [11, 14].

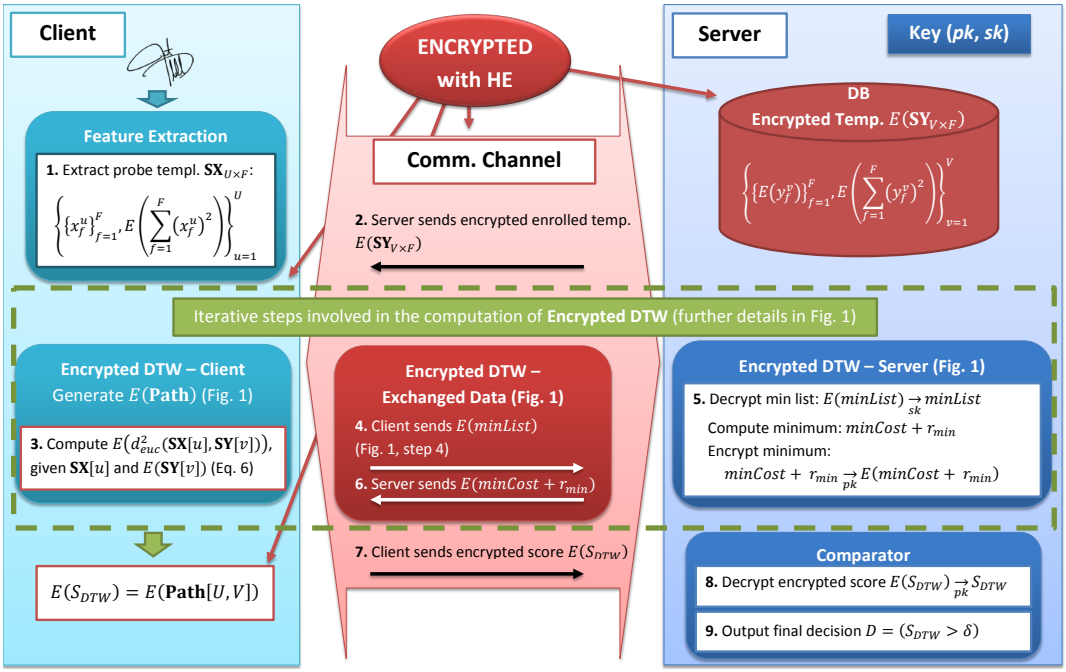

Fig. 2: **General diagram of the proposed scheme**. A local client acquires and extracts the features of the probe signature ($\mathbf{SX}_{U\times F}$) and computes the encrypted dissimilarity score ($E(S_{DTW})$) between the probe and the reference signatures ($E(\mathbf{SY}_{V\times F})$), in collaboration with a centralized server. This server holds the key pair (pk, sk) and the DB comprising encrypted templates, and outputs the final decision. All the encrypted values, either stored or transmitted, are depicted in red.

2.2. Paillier Cryptosystem

In the protected system, all the operations involved in the DTW computation are carried out in the encrypted domain. To that end, we have adapted the implementation proposed in [12]. It builds upon the Paillier homomorphic probabilistic encryption scheme [10], based on the decisional composite residuosity assumption: given a composite n and an integer z, it is hard to decide whether z is an n-residue modulo n^2.

As any other public key encryption scheme, the Paillier cryptosystem requires two separate keys: i) a public key $pk = (n, g)$, where $n = pq$ with p and q two large prime numbers such that $\gcd(pq, (p-1)(q-1)) = 1$, and $g \in \mathbb{Z}^*_{n^2}$; and ii) a secret key $sk = (\lambda, \mu)$, where $\lambda = \mathrm{lcm}(p-1, q-1)$ and $\mu = \left(g^\lambda \mod n^2\right)^{-1} \mod n$.

Given a message $m \in \mathbb{Z}_n$, its encryption is denoted as $m^* = E_{pk}(m, s) \in \mathbb{Z}^*_{n^2}$, and computed as follows:

$$m^* = E_{pk}(m, s) = g^m \cdot s^n \mod n^2 \qquad (1)$$

where $s \in \mathbb{Z}^*_n$ is a random number, providing the probabilistic nature of the cryptosystem. In order to decrypt the ciphertext m^*, we have

$$m = D_{sk}(m^*) = L\left(m^{*\lambda} \mod n^2\right) \cdot \mu \mod n \qquad (2)$$

where $L(t) = (t-1)/n$.

Two properties of Paillier cryptosystem will be used in the present scheme. First, the product of two ciphertexts, m^*_1 and m^*_2, will decrypt to the sum of their corresponding plaintexts:

$$D_{sk}\left(m^*_1 \cdot m^*_2 \mod n^2\right) = m_1 + m_2 \mod n \qquad (3)$$

Second, an encrypted plaintext, m^*_1, raised to a constant l, will decrypt to the product of the plaintext and the constant:

$$D_{sk}\left((m^*_1)^l \mod n^2\right) = m_1 \cdot l \mod n \qquad (4)$$

In order to avoid overcomplicated notation, in the description of the algorithm the keys pk and sk, as well as the random number s, will be omitted. Therefore, a generic cyphertext m^* will be simply denoted as $E(m)$.

2.3. Encrypted DTW-Based Verification

The particular implementation of the encrypted DTW proposed in the present article is shown in Fig. 1. In contrast to the unencrypted algorithm described in Sect. 2.1, all computations are now carried out directly in the encrypted domain, yielding an encrypted cost matrix $E(\mathbf{Path}_{U\times V})$, obtained from a plain probe template $\mathbf{SX}_{U\times F}$ and an encrypted reference template $E(\mathbf{SY}_{V\times F})$. To this end, Eqs. 3 and 4 are applied to convert steps 1 to 4 described in Sect. 2.1 to the encrypted domain. This way, summations of plaintexts are substituted by products in the encrypted domain, and products of plaintexts by exponentiations.

The encrypted DTW shown in Fig. 1 is used as the matching function of the full verification system depicted in

Fig. 2. In the complete system, two entities are involved: i) a client, which captures the probe signature sample, extracts the template $\mathbf{SX}_{U\times F}$, and computes the encrypted cost matrix $E\left(\mathbf{Path}_{U\times V}\right)$ and the encrypted score $E\left(S_{DTW}\right)$ between the probe template and the encrypted reference $E\left(\mathbf{SY}_{V\times F}\right)$, and ii) a server, which holds the database comprising encrypted templates, collaborates with the client on computing $E\left(S_{DTW}\right)$ and outputs the final binary verification decision $D = \left(S_{DTW} > \delta\right)$, where δ is the pre-defined verification threshold. Such a client-server model might be found, for example, in banking environments, where a central server holds the clients' information, which can be accessed from any local branch.

Therefore, two different issues need to be solved in the encrypted domain: i) compute the encrypted Euclidean distance between two points $\mathbf{x}$ and $\mathbf{y}$, $E\left(d^2_{Euc}\left(\mathbf{x},\mathbf{y}\right)\right)$, having as input $\mathbf{x}$ and $E\left(\mathbf{y}\right)$ (see steps 1 to 4 in Fig. 1); and ii) compute the minimum between three encrypted values in the $E\left(\mathbf{Path}\right)$ matrix (see step 4 in Fig. 1 and steps 4 to 6 in Fig. 2).

Encrypted Euclidean distance. Given two F-dimensional signals $\mathbf{x}$ and $\mathbf{y}$, their square Euclidean distance is

$$d^2_{Euc}\left(\mathbf{x},\mathbf{y}\right) = \sum_{f=1}^{F} x_f^2 + \sum_{f=1}^{F} y_f^2 - 2\sum_{f=1}^{F} x_f y_f \qquad (5)$$

Based on the Paillier cryptosystem properties described in Eqs. 3 and 4, given $\mathbf{x}$ and $E\left(\mathbf{y}\right)$, the computation in the encrypted domain can be performed as follows:

$$E\left(d^2_{Euc}\left(\mathbf{x},\mathbf{y}\right)\right) = E\left(\sum_{f=1}^{F} x_f^2\right)\cdot E\left(\sum_{f=1}^{F} y_f^2\right)\cdot\prod_{f=1}^{F} E\left(y_f\right)^{-2x_f} \qquad (6)$$

The first factor of the product can be locally computed on the client side when the template is extracted. The second factor is part of the encrypted template stored in the server DB, which is sent encrypted to the client (Fig. 2, step 2). In order to compute the third factor, the server sends $E\left(y_f\right)$, for $f = 1,\ldots,F$ to the client as part of the encrypted reference template (Fig. 2, step 2), and the client computes $E\left(y_f\right)^{-2x_f}$.

Encrypted minimum computation. In step 4 in Fig. 1 we need to compute the minimum between three values, without revealing to the server any information about the plain values involved to the server. To that end, a two-phase protocol is established (Fig. 2, steps 4 to 6). First (step 4), the client generates a set of K random values $R = \{r_{min}, r_2, \ldots, r_K\}$, where $r_k > r_{min}$ for $k = 2,\ldots,K$. Then, the values to be minimized are obscured with r_{min}: $E\left(m\right) \rightarrow E\left(m + r_{min}\right) = E\left(m\right)\cdot E\left(r_{min}\right)$. To further hide those values, $K-1$ additional numbers are generated randomly choosing one of those original three values $\left(E\left(m_k\right)\right)$ and obscuring it with the remaining values in R: $\left(E\left(m_k\right) \rightarrow E\left(m_k\right)\cdot E\left(r_k\right)\right)$, for $k = 2,\ldots,K$. The client then sends the complete list $E\left(minList\right)$ to the server, who decrypts all the values using its secret key sk, and computes the obscured minimum (step 5), sending it back to the client (step 6): $E\left(minCost + r_{min}\right)$. Finally, the client can compute $E\left(minCost\right) = E\left(minCost + r_{min}\right)\cdot E\left(r_{min}\right)^{-1}$.

Encrypted verification. Taking into account the description of the computation of the "encrypted Euclidean distance" and the "encrypted minimum" given above, the complete verification process is composed of nine steps (see Fig. 2):

0. During enrolment, the reference templates $\mathbf{SY}$ are acquired, encrypted using the server public key pk to generate $E\left(\mathbf{SY}\right)$ (Eq. 1) and finally stored in the database:

$$E\left(\mathbf{SY}_{V\times F}\right) = \left\{\left\{E\left(y_f^v\right)\right\}_{f=1}^{F}, E\left(\sum_{f=1}^{F}\left(y_f^v\right)^2\right)\right\}_{v=1}^{V} \qquad (7)$$

1. The client captures the probe sample and extracts the template, computing the encrypted summations for the first factor in Eq. 6:

$$\mathbf{SX}_{U\times F} = \left\{\left\{x_f^u\right\}_{f=1}^{F}, E\left(\sum_{f=1}^{F}\left(x_f^u\right)^2\right)\right\}_{u=1}^{U} \qquad (8)$$

2. The server sends the encrypted reference template $E\left(\mathbf{SY}\right)$ to the client.

Steps 3 to 6 are related to the iterative encrypted DTW verification algorithm, depicted inside a green box in Fig. 2. In order to obtain the encrypted score, $E\left(S_{DTW}\right)$, between de probe template, $\mathbf{SX}_{U\times F}$, and the encrypted reference, $E\left(\mathbf{SY}_{V\times F}\right)$, each value of the encrypted cost matrix $E\left(\mathbf{Path}[u,v]\right)$ is computed as follows:

3. The client calculates the encrypted Euclidean distance $E\left(d^2_{euc}\left(\mathbf{SX}[u],\mathbf{SY}[v]\right)\right)$ according to Eq. 6.

4. If $u, v \neq 1$ (Fig. 1 step 4), the minimum between three values is computed following the two step protocol established above. In this first step, the client generates an encrypted list of values $E\left(minList\right)$ and sends it to the server.

5. The server decrypts the list using sk, finds the obscured minimum $minCost + r_{min}$ and re-encrypts it with pk.

6. The server sends the re-encrypted minimum value to the client, setting $E\left(\mathbf{Path}[u,v]\right) = E\left(minCost\right)$.

7. When the iterative process is finished, the client sends $E\left(S_{DTW}\right) = \mathbf{Path}[U,V]$ to the server.

8. The server decrypts the score with sk, obtaining S_{DTW}.

9. In the last step, the server generates and outputs the final binary verification decision: $D = \left(S_{DTW} > \delta\right)$.

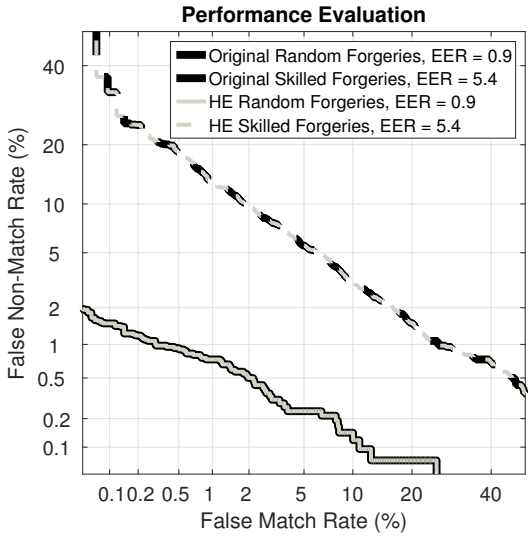

Fig. 3: Performance evaluation: DET curves.

3. PERFORMANCE EVALUATION

According to the ISO/IEC 24745 standard on biometric technologies, the first requirement for biometric template protection schemes is that verification performance is preserved with respect to the equivalent unprotected system. In this section we address this feature of the proposed method, evaluating the performance of both the original unprotected method and the proposed encrypted scheme over the BioSecurID multimodal database [13]. The signature subset used in the present work was captured in four sessions over a four month period. Four genuine signatures and three skilled forgeries were acquired in each session with a Wacom Intuos3 A4/Inking pen tablet. In order to evaluate the performance, the first 350 subjects are enrolled and modelled with the four signatures captured in the first session, being the samples of the remaining 50 subjects used for the random forgeries comparisons.

The Detection Error Trade-off (DET) curves are depicted in Fig. 3, both for the random forgeries (solid lines) and the skilled forgeries (dashed lines) scenarios. As it may be observed, the grey curves (HE protected scheme) and the black curves (original unprotected system) completely overlap. This way, the soundness of the proposed approach is confirmed: performance is not degraded at any operating point.

4. IRREVERSIBILITY AND UNLINKABILITY ANALYSIS

Let us assume an honest-but-curious threat model: both parties, client and server, follow the established protocols but may try to learn additional information about the template on the other side. Therefore, three different pieces of information should be hidden: i) only the client can have access to the plain probe template $\mathbf{SX}_{U\times F}$; ii) the plain reference template $\mathbf{SY}_{V\times F}$ should not be seen by the client, and only its encryption should be stored; and iii) the optimal path should be hidden both from the server and from the client: if the server could access this path, with the knowledge of the reference template $\mathbf{SY}_{V\times F}$, it could reconstruct the probe sample being verified, $\mathbf{SX}_{U\times F}$, and similarly the client could use $\mathbf{SX}_{U\times F}$ to guess the reference template $\mathbf{SY}_{V\times F}$.

Regarding the reference template, only encrypted values are stored and shared with the client. Similarly, the client only shares with the server encrypted distances obscured with random values ($E\left(\mathbf{Path}[u-1,v-1]+r_{min}\right)$, $E\left(\mathbf{Path}[u-1,v]+r_{min}\right)$, $E\left(\mathbf{Path}[u,v-1]+r_{min}\right)$), and padded with additional encrypted values, so that not even the minimum distance $minCost$ is known to the server.

Furthermore, in order to avoid information leakage about the optimal path, one requirement should be met: given two matrix entries with the same values, their encryption should be different. Otherwise, a malicious attacker could find out identical segments within sequences. Similarly, in the computation of $E\left(minCost\right)$, the server re-encrypts the value of $E\left(minCost+r_{min}\right)$, thus yielding a different ciphertext from the one the client sent. This is ensured by the probabilistic nature of the Paillier cryptosystem.

We may thus conclude that the first requirement established by the ISO/IEC 24745 standard, irreversibility, is met. Similarly, should an encrypted template be stolen, a new key pair (sk, pk) could be generated and shared with the client, while the entire database could be re-encrypted: renewability is also achieved.

Finally, unlinkability is also granted: since the Paillier cryptosystem is semantically secure against chosen-plaintext attacks, no information about the plain templates can be feasibly derived from encrypted templates. Therefore, no relationship can be established between the underlying biometric data. Similarly, since unencrypted distances are not preserved in the encrypted domain, no comparison can be established between templates based on the encrypted score. Additionally, given the probabilistic nature of the encryption, if $\mathbf{SX}$ is encrypted twice with the same key, the corresponding ciphertexts could not be matched: $E_{pk_1}\left(\mathbf{SX}, s_1\right) \neq E_{pk_1}\left(\mathbf{SX}, s_2\right)$.

It should be also noted that, as stated in Sect. 2, only the server has access to the plain S_{DTW} score, and the only output is a binary verification decision. Therefore, hill-climbing attacks based on the evolution of the score for different probe templates [15, 16] are prevented.

5. COMPUTATIONAL COMPLEXITY ANALYSIS

Regarding the computational complexity of the verification process, the cost is estimated in this section in terms of the encryptions and decryptions carried out, since those are the

Table 1: Computational cost.

	Client	Server
Encryptions	$\mathcal{O}\left(U^2K\right)$	$\mathcal{O}\left(U^2\right)$
Decryptions	0	$\mathcal{O}\left(U^2K\right)$
Comm. channel	$\mathcal{O}\left(U^2K\right)$	

most costly computations.

In Fig. 2 step 1, the client extracts the probe template and computes the encryption of U ciphertexts. Then, the server sends to the client the encrypted reference template $E\left(\mathbf{SY}_{V\times F}\right)$, comprising $V\cdot(F+1)$ ciphertexts (Eq. 7). In order to compute the encrypted cost matrix, it should be noted that no additional encryptions or decryptions are needed for the encrypted distances calculations, since all values had been already encrypted in step 1 or during enrolment. On the other hand, for each of the $(U-1)\cdot(V-1)$ iterations involving a minimum computation (Fig. 1 step 4), the client needs to encrypt each of the K random values r_k and send an encrypted list comprising $K+2$ values to the server. The server, in turn, needs to perform $K+2$ decryptions, one encryption, and send one ciphertext back (Fig. 2 step 4–6).

To sum up, the client needs to encrypt $U+(U-1)\cdot(V-1)\cdot K=\mathcal{O}\left(UVK\right)$ ciphertexts. The server, on the other hand, decrypts $(U-1)\cdot(V-1)\cdot(K+2)=\mathcal{O}\left(UVK\right)$ ciphertexts and encrypts $(U-1)\cdot(V-1)=\mathcal{O}\left(UV\right)$ ciphertexts. Finally, $V\cdot(F+1)+(U-1)\cdot(V-1)\left(K+3\right)=\mathcal{O}\left(VF+UVK\right)=\mathcal{O}\left(UVK\right)$ ciphertexts are exchanged between server and client ($F\ll UK$). These results are summarized in Table 1, where, without loss of generality, UV has been substituted by U^2.

For the particular system here proposed, $F=9$ time sequences are used, $K=10$ random values added in step 4 in Fig. 1 to the three values to minimize, and the average sequence length in BiosecurID is $\bar{U}=370$. Using Kun Liu's implementation of the Paillier cryptosystem in Java[1], and running the experiments in a machine with an Intel Core i7 with four 2.67 GHz cores, one comparison takes approximately one minute and 216 MB of data are exchanged. It should be noted that this is just an illustrative approximation: code should be optimized and a server, instead of a regular desktop computer, would bear the highest computational cost.

6. CONCLUSIONS

We have presented a new biometric template protection scheme based on variable length templates and Homomorphic Encryption. The stored templates, the exchanged data and all the computations are carried out on the encrypted domain, so that no biometric information is revealed. It has been shown that verification performance is preserved, while irreversibility, unlinkability and renewability are provided, thus meeting the requirements of the ISO/IEC 24745 standard on biometric technologies. Furthermore, since the plain dissimilarity score is never shared, hill-climbing based attacks are prevented.

As future work, we will study how to reduce the computational cost and amount of exchanged data applying subsampling strategies, comparing it to other BTPs, and analyse the impact of K (see the computation of the encrypted minimum in Sect. 2.3) on the security provided.

[1]Publicly available at http://www.csee.umbc.edu/~kunliu1/research/Paillier.html

Acknowledgements

This work has been partially supported by projects BioShield (TEC2012-34881) from Spanish MINECO, BEAT (FP7-SEC-284989) from EU, and *CECABANK*. M. G.-B. is supported by a FPU Fellowship from Spanish MECD.

7. REFERENCES

[1] ISO/IEC JTC1 SC27 Security Techniques, *ISO/IEC 24745:2011. Information Technology - Security Techniques - Biometric Information Protection*, International Organization for Standardization, 2011.

[2] P. Campisi, E. Maiorana, et al., "Cancelable templates for sequence based biometrics with application to on-line signature recognition," *IEEE Trans. SMC-A*, vol. 40, no. 3, pp. 525–538, 2010.

[3] E. Aragones-Rua, E. Maiorana, et al., "Biometric template protection using universal background models: An application to online signature," *IEEE TIFS*, vol. 7, no. 1, pp. 269–282, 2012.

[4] A. Levi, E. Savas, et al., "Biometric cryptosystem using online signatures," in *Proc. ISCIS*, 2006, vol. 4263 of *LNCS*, pp. 981–990.

[5] C. Rathgeb and A. Uhl, "Statistical attack against fuzzy commitment scheme," *IET Biometrics*, vol. 1, no. 2, pp. 94–104, 2012.

[6] M. R. Freire, J. Fierrez, et al., "Biometric hashing based on genetic selection and its application to on-line signatures," in *Proc. ICB*, 2007, pp. 1134–1143.

[7] M. R. Freire, J. Fierrez, and J. Ortega-Garcia, "Dynamic signature verification with template protection using helper data," in *Proc. ICASSP*, 2008, pp. 1713–1716.

[8] S. Ye, Y. Luo, et al., "Anonymous biometric access control," *EURASIP J. on Inf. Security*, vol. 2009, pp. 1–17, 2009.

[9] M. Barni, T. Bianchi, , et al., "A privacy-compliant fingerprint recognition system based on homomorphic encryption and fingercode templates," in *Proc. BTAS*, 2010, pp. 1–7.

[10] P. Paillier, "Public-key cryptosystems based on composite residuosity classes," in *Proc. EUROCRYPT*, 1999, pp. 223–238.

[11] M. Martinez-Diaz, J. Fierrez, et al., "Mobile signature verification: Feature robustness and performance comparison," *IET Biometrics*, vol. 3, pp. 267–277, 2014.

[12] H. Zhu, X. Meng, and G. Kollios, "Privacy preserving similarity evaluation of time series data," in *Proc. EDBT*, 2014, pp. 499–510.

[13] J. Fierrez, J. Galbally, et al., "BiosecurID: a multimodal biometric database," *Patt. Analysis and App.*, vol. 13, pp. 235–246, 2009.

[14] A. Kholmatov and B. Yanikoglu, "Identity authentication using improved online signature verification method," *Patt. Rec. Letters*, vol. 26, pp. 2400–2408, 2005.

[15] M. Gomez-Barrero, J. Galbally, and J. Fierrez, "Efficient software attack to multimodal biometric systems and its application to face and iris fusion," *Pattern Recognition Letters*, vol. 36, pp. 243–253, 2014.

[16] E. Maiorana, G. E. Hine, and P. Campisi, "Hill-climbing attack: Parametric optimization and possible countermeasures. an application to on-line signature recognition," in *Proc. ICB*, 2013, pp. 1–6.