A Marriage between Cryptography and Signal Processing for Privacy Protection in Biometric Based Recognition Systems

Mauro Barni, Giulia Droandi and Riccardo Lazzeretti

Systems employing biometric traits for people authentication and identification are witnessing growing popularity due to the unique and indissoluble link between any individual and his/her biometric characters. For this reason, biometric templates are increasingly used for border monitoring, access control, membership verification, and so on. When employed to replace passwords, biometrics have the further advantage that they do not need to be memorized, and are relatively hard to steal. Nonetheless, unlike conventional security mechanisms such as passwords, biometric data are inherent parts of a person's body, and cannot be replaced if they are compromised. Even worse, compromised biometric data can be used to have access to sensitive information and to impersonate the victim for malicious purposes. For the same reason, biometric leakage in a given system can seriously jeopardize the security of other systems based on the same biometrics. A further problem associated to the use of biometric traits is that, due to their uniqueness, the privacy of their owner is put at risk. In fact geographical position, movements, habits and even personal beliefs can be tracked by observing when and where the biometric traits of an individual are used to identify him/her.

Processing biometric signals while they are encrypted provides a secure and elegant way to overcome the above problems [1], especially those related to privacy protection. Thanks to the opportunities offered by Secure Multi Party Computation (SMPC) techniques [2], it is in fact possible to carry out the match between any two biometric templates by working only on encrypted data. Furthermore, it is also possible to design the underlying matching protocol in such a way that the final result of the match is known only to the intended party without leaking any information about the biometric templates or the identity of the biometric owner. The wide range of techniques allowing to process encrypted signals are usually known as Signal Processing in the Encrypted Domain (SPED).

As an example, let us consider a scenario in which a server is interested to know whether

the owner of a biometric template is part of a list of enrolled individuals, e.g. the users who can access a certain service, or the criminals contained in a police record. The server has a database of plain biometric templates and the user submitting the query is interested to access the service without revealing his/her identity. Alternatively, the user submitting the query may be interested to know whether a biometric signal matches with one of the templates stored in the server, without that the server accesses the result of the query. According to the SPED paradigm, the above goals are achieved by letting the server comparing the templates in the database with the one provided by the user directly in the encrypted domain. While apparently impossible, a functionality like the above can be implemented by resorting to SMPC. In fact, it is known that virtually any computable function or algorithm can be evaluated by means of a SMPC protocol [3]. In the simplest cases, like those considered in this paper, the protocol involves only two parties. In this case, we talk about Secure Two-Party Computation (STPC). In a general STPC setting, one party, say the client $\mathcal{C}$, owns a signal that must be processed in some way by the other party, hereafter referred to as the server $\mathcal{S}$. $\mathcal{S}$ must process $\mathcal{C}$'s signal without getting any information about it, in some cases not even the result of the computation. At the same time, $\mathcal{S}$ is interested to protect the information used to process the signal.

Two of the main approaches to SPED are, respectively, Homomorphic Encryption (HE) [4] and Garbled Circuits (GC) [5]. HE provides a way to evaluate linear operations on encrypted data, however when non-linear operations are involved, it is necessary to resort to ad-hoc, interactive and usually complex protocols. On the other hand, GC allows to evaluate any function that can be represented with an acyclic boolean circuit. In some cases, however, the boolean circuit required to describe the functionality is so complex that it makes the use of GC problematic. Given the complementary pros and cons of HE, OT and GC, the use of hybrid protocols has also been proposed to take advantage of the benefits offered by the two approaches [6]. Recently, Fully Homomorphic Encryption (FHE) schemes [7] have been devised, allowing the evaluation of any function without any interaction between the involved parties. Unfortunately, FHE is still highly inefficient, principally due to the huge size of the public key.

Despite many recent advances and the introduction of more efficient cryptographic primitives, the complexity of SPED protocols is often so high to prevent their use in practical applications. In order to reduce the complexity down to a manageable level, it is necessary that the underlying biometric processing algorithms and the STPC protocol are designed jointly, by taking into account both the cryptographic and the signal processing facets of the problem. On the contrary,

the most common approach used so far has been that of taking a classical biometric matching algorithm and transforming it into a protocol to be run in the encrypted domain. It is arguable that much better results can be obtained by developing a class of algorithms that are explicitly thought to ease a SPED implementation, e.g., by considering in advance which are the most complex operations to be carried out in a secure way and trying to avoid them.

In general, it is necessary that the biometric templates are represented through a vector of features of constant length and that a simple distance measure (e.g,. the Hamming or Euclidean distance) can be used to measure the degree of similarity between two vectors. If the above conditions are satisfied, a biometric authentication or verification protocol can be developed easily by composing few blocks: distance computation, minimum selection and comparison against a threshold [8], [9]. The search for efficiency is not limited to the choice of a suitable matching algorithm: representation issues must be considered as well. In the end, the complexity of SPED primitives depends on both the number of features the matching algorithm relies on and the number of bits used to represent them. By using less features and/or less bits, the complexity of the protocol decreases at the expense of matching accuracy. It is then necessary to find a proper configuration to couple efficiency and accuracy. Signal processing expertise can be exploited in several other ways: for example, it has been proven in [10] that using a common mask for iris recognition instead of a varying one, dramatically simplifies the implementation of an iris recognition system in the encrypted domain, with a very reduced impact on the performance of the system.

The present article aims at illustrating the basics of STPC, including the way it can be applied to the protection of biometric templates, and at explaining how the signal processing and cryptographic points of view can be considered together in order to obtain efficient, secure and accurate SPED protocols. We also review some works in which such an approach has been used successfully for different biometric modalities, including fingerprint matching, iris recognition and face recognition.

I. Overview of basic SPED tools

In this section, we provide a concise introduction to the basic primitives SPED technology relies on. The tools presented here and the protocols described in the next sections are provably secure in a semi-honest setting [1], i.e., when the involved parties execute the protocol without deviating from it, but at the same time try to obtain as much information as possible about the other party's data. The choice of a semi-honest model is due to the fact that while protocols

providing security against a malicious party would be preferable, their implementation has a very high complexity. Moreover, at least in principle, protocols guaranteeing security in the semi-honest model can always be modified to make them secure under more stringent threat models, even if such an increased security comes at the price of a higher complexity.

Below we provide a qualitative description of various tools, focusing on their strengths and limitations.

A. Homomorphic Encryption (HE)

A cryptographic scheme (cryptosystem) is *homomorphic* [11] if an operation over encrypted data exists which correspond to another operation over the plain message. In other words, by indicating with $[\![x]\!]$ the encryption of a plain value x, we have $[\![x]\!] \boxtimes [\![y]\!] = [\![x \boxplus y]\!]$, for some operations $\boxtimes$ and $\boxplus$. Most homomorphic encryption schemes rely on asymmetric cryptography, and the homomorphic property holds under encryption with the public key of one of the parties involved in the protocol. Unless otherwise stated, in the following we assume that the private key is known only to the client $\mathcal{C}$, while the server $\mathcal{S}$ has access only to the public key.

The most common homomorphic cryptosystems (see for instance [12], [13]) are additively homomorphic, that is $\boxtimes = \times$ and $\boxplus = +$. An additively homomorphic cryptosystem permits to a party which does not know the decryption key to obtain the encryption of the sum between two values available to him only in encrypted form. In the same way, he can compute the encryption of the product between a known integer value c and a value available to him under encryption as $[\![cx]\!] = [\![x]\!]^c$. More complex operations can be implemented by resorting to an interactive protocol between $\mathcal{S}$ and $\mathcal{C}$.

Despite its elegance, the use of HE to compute with encrypted data comes at quite high computational cost. In Paillier's cryptosystem, for instance, even plain values represented with few bits are encrypted in 2048 bit long ciphertexts (the plaintext after the encryption) so that sums and products between plain values are mapped respectively to products and exponentiations on very long ciphertexts. Non-linear operations, such as products between encrypted values or comparisons, are even more complex and require interaction between the parties. For this reason, the communication complexity of an HE protocol depends on the number of transmitted ciphertexts, as well as on the number of communication rounds, while computation complexity is usually dominated by the number of exponentiations on encrypted values (the most expensive operation) required by the protocol.

Multiplicative homomorphic cryptosystems exist as well [4], [14], allowing the evaluation of products between encrypted values ($\boxtimes = \times, \boxplus = \times$), but they have a lower practical utility with respect to additive HE.

Fully Homomorphic Encryption schemes allow both the evaluation of additions and products in the encrypted domain. C. Gentry [7] developed the first secure *Somewhat Homomorphic Encryption* (SHE) and *Fully Homomorphic Encryption* (FHE) schemes, working on binary data. SHE allows the evaluation of a limited number of additions and multiplications, while FHE extends SHE to bypass such a restriction at the price of a huge increment of memory and computational complexity, thus making all FHE schemes proposed so far highly impractical.

By using the original Gentry's SHE scheme and subsequent improvements, it is possible to evaluate binary circuits composed by up to a maximum number of XOR and AND gates directly on $\mathcal{S}$'s side without any interaction with $\mathcal{C}$, thus making protocols based on SHE very appealing for clients equipped with low power devices. Efficient SHE solutions can be designed to evaluate circuits having a given (small) number of AND gates and then transformed into more expensive FHE solutions, if necessary. Luckily in most biometric recognition algorithms, the number of required operations is known in advance, making the use of protocols based on SHE possible.

A further simplification has been introduced in [15] where a SHE scheme operating on integer values has been proposed, thus allowing to encrypt each input directly, instead of decomposing it into bits and then using bitwise encryption. On the other hand, SHE (or FHE) schemes working on integers permit only the evaluation of polynomial functions (up to a certain degree for SHE).

B. Oblivious Transfer

Oblivious Transfer (OT) [16] is an STPC protocol that enables one party, say the server $\mathcal{S}$, to forward one out of n of messages $(x_1, x_2, \ldots, x_n)$ to the client $\mathcal{C}$. $\mathcal{C}$ chooses the index i of the element that he would like to get. At the end of the protocol, the server gets no information on the index i and the client does not get any information on the other x_j's. The possibility to move great part of the computation to an offline phase, during which several OT's are evaluated on randomly chosen values, permits to greatly simplify the complexity of OT. The random values are replaced by the actual values during a much more efficient online phase [17]. Neglecting the offline complexity and thanks to precomputation, the online communication of multiples 1-out-of-2 OTs is reduced to about 2ℓ bits for each OT, where ℓ is the message bitlength, transmitted in parallel in 2 rounds. With regard to computational complexity, only simple XOR operations are required on both sides.

C. Garbled Circuits

The possibility of securely evaluating any binary circuits was proposed for the first time by Yao in his seminal paper [5]. Yao's protocol, named garbled circuit (*GC*), involves both the parties in the computation and distributes the computation between $\mathcal{S}$ and $\mathcal{C}$. $\mathcal{S}$ *encrypts* (garbles) each gate of the circuit and maps each input bit into a random string. Then $\mathcal{S}$ sends the garbled circuit to $\mathcal{C}$ together with the secrets corresponding to $\mathcal{S}$'s inputs. The secrets associated to $\mathcal{C}$'s inputs are transmitted to $\mathcal{C}$'s by means of OT. In the last phase of the protocol, $\mathcal{C}$ *decrypts* the gates by using the input secrets and obtains the final output of the circuit.

For a long time, GC were thought to be highly impractical. However, they have recently gained renewed popularity, thanks to several efficiency improvements (most of which summarized in [18]). The protocol associates a secret of 80 bits to each bit involved in the computation, making single core operations lighter than in HE (we recall that a Paillier ciphertext is 2048 bit long). Unluckily, even if most of the computation is performed on $\mathcal{S}$'s side, $\mathcal{C}$ must also take an active part in the protocol. The computational complexity depends linearly on the number of non-XOR gates composing the circuit (which in turn depends on the input bitlengths), in fact XOR gates can be evaluated with negligible computational and communicational complexity. It is important to underline that a GC protocol requires only 2 rounds, regardless of the circuit size and the number of input bits (an additional round is necessary if the final result must be sent to $\mathcal{S}$). We also point out that circuit garbling does not depend on the actual inputs and in some particular scenarios, where the functionality to evaluate is known in advance, circuit encryption and transmission can be precomputed.

Given that the complexity depends on the number of gates composing the circuit, GCs are suited for operations such as sums and comparisons, for which the number of gates depends linearly on the input bitlength. On the contrary, GCs are less efficient when the number of gates grows more than linearly with the input bitlength. This is the case, for instance, of products and divisions for which the circuit size depends quadratically on the bitlength of the inputs.

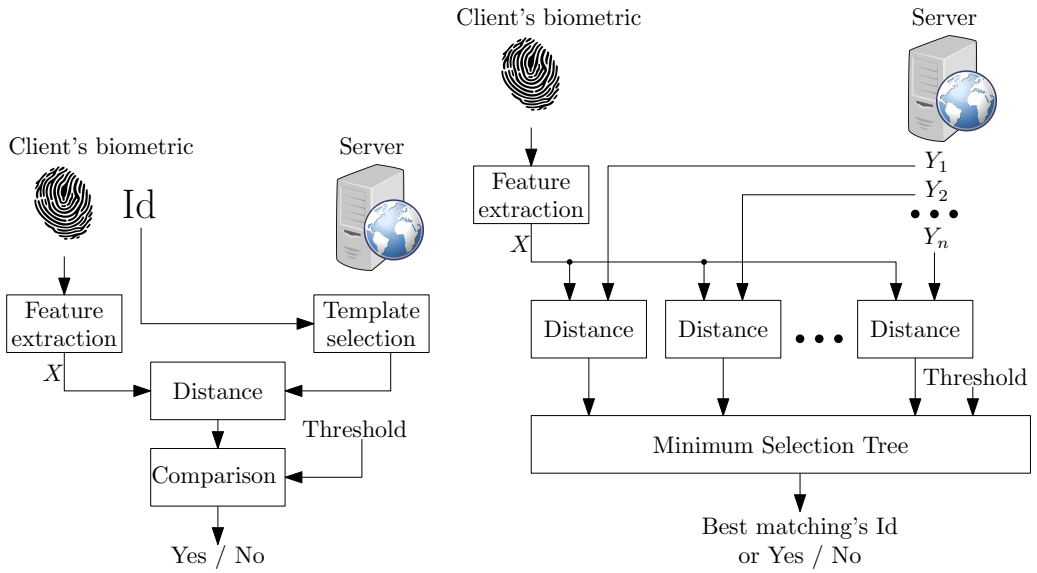
D. Hybrid protocols

Sometimes, complex protocols can be divided into subprotocols and different tools can be used for their implementation, in order to take the best from each approach. Such an idea has been applied to develop hybrid protocols working with HE and GC in [6], but can be extended also to different tools. Hybrid protocols require the adoption of proper interfacing protocols to link subparts implemented by relying on different technologies. For instance, it may happen that an

intermediate value x output by a HE protocol must be used as input in a GC subroutine, or vice versa. In this case, the different parts of the protocol must be connected in such a way that the security of the whole system is guaranteed. At the same time the representation of the variable x must be adapted to the subprotocol requirements.

II. Biometric recognition protocols

Biometric recognition protocols can be divided in two main categories: in the first scenario, usually referred to as *authentication*, the user is interested to demonstrate that he is who he claims to be, while in the second one, called *identification*, the goal of the protocol is to determine the identity of the user submitting the biometric template. To bette protect the users' privacy, in some cases, SPED-based identification protocols simply verify whether the user is enrolled in the database or not. The server $\mathcal{S}$ owns a database of enrolled biometric feature vectors ($\{Y_i\}$, $i = 1, \ldots, n$) and the client $\mathcal{C}$ owns a biometric vector X. In all cases, $\mathcal{S}$ and $\mathcal{C}$ are interested to protect the privacy of their data.



(a) Authentication (b) Identification

Fig. 1: Biometric recognition protocols.

In the authentication problem (Figure 1(a)), $\mathcal{C}$ submits a new instance of his biometrics. The fresh biometric template is processed to extract a feature vector X that is sent to $\mathcal{S}$ together with an identifier, used by $\mathcal{S}$ to select the corresponding enrolled template Y_{id} in the database. The distance $d(X, Y_{id})$ between the query X and the template Y_{id} is evaluated and the result is compared against an acceptance threshold.

In the identification scenario (Figure 1(b)), the client extracts the feature vector X from the fresh biometric template and submits it to the server without revealing his identity. The server

must verify whether an index i exists such that $d(X, Y_i) < \varepsilon$. To do so, $\mathcal{C}$ and $\mathcal{S}$ first evaluate $d_i = d(X, Y_i)$ for all $i = 1 \dots n$, then they find the minimum among all d_i and the threshold through a minimum selection tree returning *yes* if the minimum distance is below the threshold, and *no* otherwise. It is also possible to modify the minimum tree so that the output is a user's identification index instead of a yes/no answer.

As it can be seen, a general recognition protocol is composed by a few number of basic blocks: feature extraction, distance computation, comparison and minimum selection. Feature extraction involves only data provided by one party, hence it is usually implemented in the plain domain. On the other hand, distance computation, comparison and minimum selection involve private data owned by $\mathcal{C}$ and $\mathcal{S}$ and for this reason must be implemented by resorting to SPED. There are many possibilities to implement these blocks in a privacy preserving way. The choice depends on many factors, such as device configuration, network bandwidth and latency, computational capabilities of $\mathcal{S}$ and $\mathcal{C}$. In this section, we provide a brief description of how the various blocks can be implemented, leaving a more detailed description to the next sections.

The Hamming and the squared Euclidean distances are the most commonly used distances because they can be easily implemented in a SPED setting. The Hamming distance is used whenever the biometric template corresponds to a binary vector, while the squared Euclidean distance is used on integer biometric vectors (the squared version is used to avoid the expensive computation of the square root). Both distances can been implemented by using GC, HE or OT. In [8, Chapter 7] the authors show that, due to its binary nature, the Hamming distance can be efficiently implemented by using GC, while an HE implementation is preferable for the squared Euclidean distance [19], since HE allows an efficient computation of products. An efficient OT implementation of both Hamming and squared Euclidean Distance has been proposed in [20]. It is also possible to implement such distances through SHE [21], while, given the limited number of operations required in both cases, resorting to FHE is not necessary.

Comparison is needed to verify whether a certain distance is lower than the acceptance threshold (squared threshold if the squared Euclidean distance is used). Its implementation [8, Chapter 7] requires that the involved quantities are represented in binary form, thus making GC-based implementations more attractive. Implementations based on HE [19] have also been proposed, but they require several interactions between the parties.

Starting from a comparison protocol, it is possible to evaluate the minimum among two encrypted values by using the output of the comparison to select between two numbers x and y in a multiplexer. Given the necessity of a comparison operator, a GC implementation is usually

preferable. The protocol for the selection of the minimum between two numbers can be easily extended to the computation of the minimum among n values using a reverse tree implementation [8, Chapter 7] where each node computes the minimum between the results of the previous left and right subtrees. The minimum selection tree can be modified to output the minimum value or the corresponding identifier.

III. Optimization of SPED protocols through cryptographic primitive selection

In this section, we provide an overview of how the use of different cryptographic primitives can be exploited to improve the performance of biometric recognition protocols. For the sake of simplicity we do not discuss the improvements in the implementation of the basic cryptographic primitives and we leave the description of signal processing optimizations to the next section.

One of the first papers addressing privacy preserving biometric authentication is [22]. The protocol does not focus on a specific biometric modality, but rather on a general biometric representation consisting of a binary string. It then presents a secure implementation of the Hamming distance computation based on Private Information Retrieval.

An implementation of privacy preserving biometric identification protocols operating in the semi-honest setting, implemented according to the overall scheme presented in the previous section, has been proposed by Erkin et al. in [19]. The recognition protocol is based on eigenfaces [23], it achieves 96% correct classification averaged over different lightning conditions, 85% when different face orientations are considered and 64% when face size varies as well. In contrast to most SPED biometric recognition protocols, the feature extraction step is carried out in the encrypted domain by relying on the homomorphic properties of Paillier cryptosystem [12]. Squared Euclidean distance computation is also implemented by relying on Paillier system, while the comparison protocol is implemented according to the scheme proposed by Damgard et al. in [13]. The protocol complexity was evaluated by running it on a computer with a 2.4 GHz dual-core processor, and using the *ORL Database of Faces* [24] obtaining a runtime of about 40 seconds for a single match. The runtime could be reduced to 18 seconds by resorting to precomputation. As shown in Table I, the authors have demonstrated that it is possible to further reduce the computational and communication complexity by assuming that the parameters of the eigenface extraction protocol are public (such an assumption has been adopt by virtually all subsequent works on the same topic).

DB size	Computational complexity (sec)			Communication complexity (KB)	
n	Full Query	With precomputation	Public Eigenfaces	Full Query	Public Eigenfaces
10	24	8.5	1.6	2725	149
200	34.2	14.5	11.4	5497	2921
320	40	18	18.2	7249	4674

TABLE I: Computational and communication complexity of privacy-preserving face recognition [19].

Erkin et al. protocol has been improved by Sadeghi et al. [25], who proposed a full-GC and a hybrid protocol for eigenface biometric recognition, where HE is used to compute the distance and GC for the comparison. As shown in Figure 2, the resulting protocol is 30% faster than [19], when implemented on a PC having a 2.6 GHz processor.

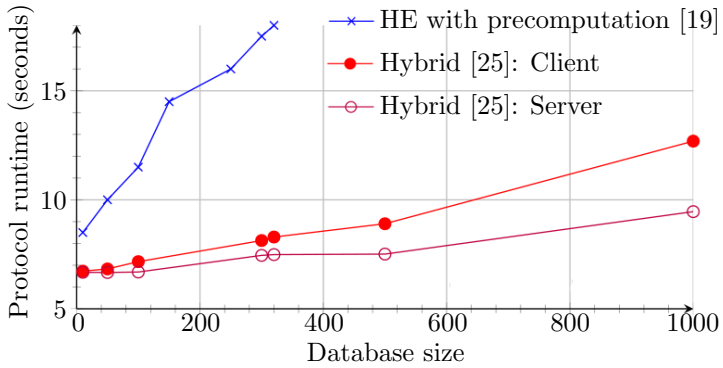


Fig. 2: Runtime comparison of HE [19] and hybrid [25] implementations of the Eigenface protocol.

In [26] the authors propose a new technique (described in the section) for template extraction, called *SCiFI*. The protocol evaluates distances between faces by using Paillier HE and then implements the comparison by using an 1-out-of-d OT, where d is the maximum value that the distance can assume. The experiments were performed on two computers with a 2.6 GHz processor and a 2.8 GHz dual-core processor respectively. The online time complexity is about 0.30s for a single match.

Moving from face recognition to iris-based systems, Luo et al. [27] implemented a HE-based privacy preserving iris identification protocol based on iriscode [28] and tested it on the CASIA Iris database [29], containing 100 iriscodes of 9600 bits each. The resulting protocol needs 27.1 minutes on average for a single query on a computer equipped with a 2.4 GHz processor. Such a large complexity is justified by the very large bit length of iriscodes (9600-bit) which are bitwise encrypted by means of Paillier cryptosystem. A different approach is presented in [30], where the authors use a hybrid (HE and GC) protocol for biometric identification and optimize it by precomputing most of the operations. Further improvements are obtained by optimizing the multiplication protocols and by using the DGK scheme [13] for comparison

	Server Runtime	Client Runtime	Bandwidth
Iriscode	89ms+149.25ms/rec	0ms+22.61ms/rec	0.5KB+19.9KB/rec
Fingercode	0.22ms+1.42ms/rec	4.7ms+1.08ms/rec	2.12KB+0.86KB/rec
Minutiae	6ms+339ms/rec	25ms+1876ms/rec	16KB+294KB/rec

TABLE II: Online performances of iriscode, fingercode and minutiae based fingerprint identification [30]. Some of the overheads depend on the server's database size, in which case the computation are indicated per record ("/rec").

computation. A C implementation of the protocol has been tested on a 2.13 GHz dual-core processor obtaining results about 25% faster with respect to the same protocol implemented by using HE. Online computation times are summarized in Table II. In particular the comparison between two encrypted 2048-bit iriscodes requires only 0.15 sec.

In [31] and [10], the authors present an iris identification protocol based on two different full-GC implementations (more details in the next section). In [31] the authors run a Java implementation of the protocol on a client with a 2.66 GHz quad-core processor connected through a local-area network with a server equipped with a 2 GHz processor. They tested the protocol on databases of different sizes n obtaining a total bandwidth of $475n + 0.08n^2$ KB and a runtime of about 2.4 sec for each match.

The protocol described in [10] has been implemented in Java and run on a machine mounting a 3.00 GHz processor over iriscodes of the CASIA Iris database [29] represented with 9600 and 2048 bits. Thanks to offline computation of the circuit garbling phase and circuit transmission, the matching between two iriscodes represented with 2048 bits needs 0.56 sec and the transmission of 571KB, while the matching between two iriscodes represented with 9600 bits needs 2.5 sec and the transmission of 2655KB.

We conclude this section by considering fingerprint matching. Given the necessity of working with finite length feature vectors, most schemes proposed so far rely on the fingercode representation of fingerprints [32]. This is the case of the system proposed by Barni et al. [33], [34] implementing a Paillier based identification protocol. The execution of the protocol on a database with 64 identities takes about 16 sec on a PC equipped with a 2.4 GHz dual-core processor. Fingerprint identification is also addressed in [30], where protocols similar to those used for iris recognition are used. With respect to [34], the implementation based on fingercode is 35 times faster (client online runtime is 0.35 sec while server's one is 0.45 sec). The protocol has been also adapted to operate on minutiae [35] (results in [32] reports a false acceptance rate lower than 1%), but runtimes increase significantly. Table II shows the performance of the protocol when 32 minutiae are used to represent the fingerprint. Yet another hybrid implementation is described

Database size	Running time(sec)	Bandwidth (KB)
128	2.22	966.84
256	4.33	1927.71
512	9.12	3849.48
1024	18.11	7692.98

TABLE III: Online performances of the fingercode identification presented in [36].

in [36] for fingercode-based identification. Table III shows the online computation time obtained with a Java implementation running on two machines equipped with a 2.0 GHz processor.

A somewhat different approach, relying on a different use of the available cryptographic primitives, has been proposed by Bringer et al. [20]. The new approach, called GSHADE, is based on a hybrid use of OT and GMW [37]. GMW is a SMPC primitive similar to Yao's garble circuits. It implements the to-be-computed functionality as a binary circuit, however, it performs the secure evaluation by relying on shares rather than encrypted gates. GSHADE has been tested by running a C++ implementation on two computers with 3.2 GHz precessor. By considering a database of 320 iriscodes of 2048 bit each, the communication complexity of GSHADE is around 3 times larger than that of the hybrid protocol described in [30]. However, the GSHADE protocol is 35 times faster than the system presented in [30]. Similar results have been obtained with fingercodes (runtime improves by a factor 500 with respect to [36]) and eigenfaces (with a runtime runtime improvement of a factor ranging from 66 to 100 with respect to [19]).

With the increased popularity of fully and somewhat homomorphic encryption schemes, a few completely non-interactive solutions for privacy preserving biometric recognition have been proposed. In [21], the first non-interactive biometric authentication protocol, based on an integer extension of the SHE scheme described in [38], is presented. All the computation is moved on the server's side, leaving only the encryption of the inputs and the decryption of the result to the client. With regard to complexity, a C++ implementation of the protocol has been run on a machine mounting a 3.30 GHz processor. With respect to an equivalent implementation based on Pailler cryptosystem, the computational complexity is considerably reduced (59 sec for Troncoso et al. implementation versus the 420 sec of an equivalent Paillier implementation), with the additional advantage of avoiding the interaction between the parties. On the other hand, due the larger expansion factor of lattice based cryptosystem like [38], the communication complexity is larger than the Paillier-based version: 393MB in [21] and 16.4MB for the Paillier-based version. Another authentication protocol based on SHE has been proposed in [39]. Thanks to a packed representation of the biometric templates, the protocol is able to compute the Hamming distance

with only three products. Tests performed on a 3.07 GHz processor show that only 18.10ms are necessary for distance computation, which is not only faster than the SHE based implementation of [21], but also faster than the Hamming distance computational time of SCiFI (310ms) [26] and [30] (150ms). In both [21] and [39], only the distance is computed by means of SHE operating on integers. Such schemes, in fact, permit only the computation of polynomial functions of the inputs and hence they cannot be used for comparisons. For this reason, in [21] and [39] the final comparison is carried out in plain by the client.

For completeness, we highlight that beyond papers strictly focusing on biometric recognition, other interesting privacy preserving applications that can be also applied to biometric protocols have been daveloped. For example, [40] presents a new scheme for privacy preserving evaluation of sample set similarity (*EsPRESSo*) that can be used for iris matching, while in [41] the authors address privacy-aware media classification, and hence also face recognition, on public databases.

IV. Signal processing optimization

Even if the development of more and more efficient cryptographic primitives and their adaptation to the specific needs of biometric recognition protocols, has led to considerable complexity reduction, further ways to reduce the complexity of SPED protocols are needed to match the requirements set by practical applications. A less explored, but promising, strategy is the optimization of the signal processing aspects of the algorithms to be implemented in a SPED fashion. Generally speaking, signal processing optimization can be carried out at three different levels[1]: i) algorithmic level, ii) feature choice and distance selection, iii) feature representation level. In the first case, the matching algorithm is designed in such a way to avoid the operations that most complicate a SPED implementation. As an example, when considering an HE-based implementation, algorithm designers should try to minimize the use of non-linear operations. With regard to feature and distance selection, it is desirable that the computation of distances between feature vectors can be easily implemented by means of the available STPC primitives. As a matter of fact, in identification scenarios the number of distances to be computed grows linearly with the size of the database [31], hence calling for a careful design of this part of the protocol. The last optimization level concerns the size of the feature vector and the number of bits used to represent the feature values. In fact, both aspects have a great impact on protocol efficiency. Investigating the relationship between the size of the feature vector and the number

[1]While this classification is quite general, in some cases the various levels can not be clearly identified and optimizations operating at different levels may depend on each other in a complex way.

of bits used to represent it on one side and the accuracy of the matching process on the other side, may lead to a significant simplification of the resulting protocol. Of course, all the above considerations are not independent from the STPC primitives the protocol relies on. Hence the preferable tool for each algorithm configuration must be be selected among all the available SPED tools. As shown in the previous section, this is often a hard choice depending on many factors such as the bandwidth and the latency of the network, the characteristics of the devices available at the client and server side, etc.

In the following, the various optimization levels are described in more details. For each level, we provide one or more practical examples of its use in a biometric matching protocol.

A. Algorithm level optimization

Given a matching algorithm, some optimizations can be applied to improve its performance, trying to avoid the operations that are most expensive when implemented in a SPED setting.

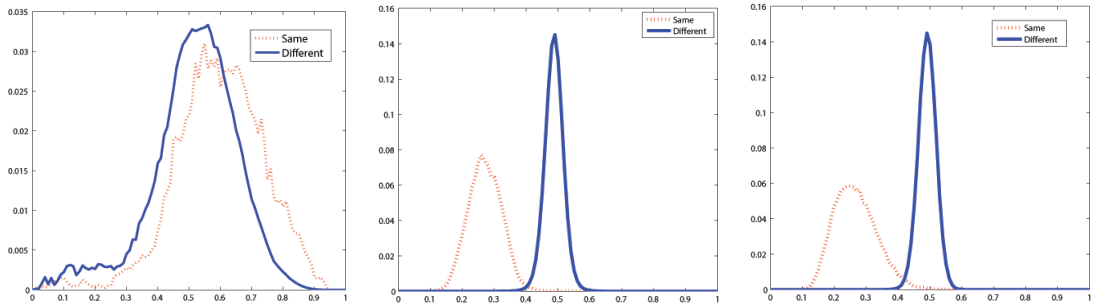
In identification protocols, the complexity mainly depends on the number of biometric templates contained in the database, since this directly affects the number of matches that must be computed. In the iris recognition protocol presented in [31], the matching between two iriscodes is based on a normalized Hamming distance involving two iris masks (one for each iris template) which are used to remove the non-informative parts of the iris code, usually those impaired by reflexes, eyelashes and shades. Given the binary nature of the iriscode, a GC solution is very efficient with regard to Hamming distance computation, but the use of the two masks involves 2 non-free AND gates for each bit, approximately tripling the complexity of the modified Hamming distance circuit. The idea put forward in [31] is to reduce the DB size through a filtering phase during which only the most promising templates are selected. The non-masked Hamming distance is evaluated on a subset of 128 bits, whose position is chosen between the usually unmasked bits, selected in the query and all the n templates in the database. Then the randomized indexes of the k templates with the smallest distances are passed to the client. After the filtering phase, $\mathcal{C}$ and $\mathcal{S}$ run an identification protocol where the masks are used to refine the distance computation and the input secrets of the k templates and masks are retrieved by $\mathcal{C}$ through an OT protocol. Thanks to the above solution, the complexity of the protocol is significantly reduced: with $k \approx n/10$ a total bandwidth of $475n + 0.08n^2$ KB is reported, which is considerably lower than the $3.6n$ MB needed for an exhaustive comparison. On the negative side, the protocol does not guarantee that the correct biometrics are selected for the second phase hence decreasing the accuracy of the

$k = 1$	$k = 10$	$k = 20$	No filter
19.5%	8.2%	6.1%	3.1%

TABLE IV: False Rejection Rates of [31] according to the number of biometric templates selected in the filtering phase among the 2710 elements in the database.

identification. Table IV shows the False Rejection Rates with different values of k and without filtering.

A different algorithmic optimization for iris-based identification has been proposed in [10]. It relies on the use of a common mask, estimated from all the masks associated to the iriscodes in the database. Given a dataset, the distribution of the mask overlap regions is computed. Figure 3(a) shows that masks from the same individuals have larger overlap than those from different individuals, *concluding that among all masks, those of each individual have larger inter-correlation.* On the other hand, as shown in Figure 3(a), also masks belonging to different



(a) Mask overlap sizes. (b) Real masks. (c) Common mask.

Fig. 3: Distributions in Iriscode identification in [10] over iriscodes in the CASIA Iris database [29].

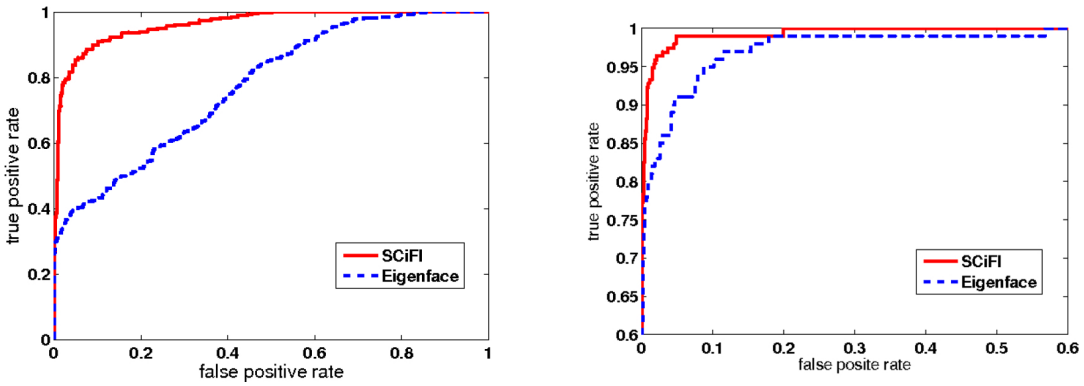
individuals are quite similar. By relying on this observation, the authors proposed to simplify the circuit implementing the masked distance by using a common mask for all the iriscodes. The common mask is set to '1' at all bit positions where the percentage of the pre-aligned masks equal to '1' at those positions exceeds an empirically-determined threshold λ. The common masks do not reveal information about the single templates in the database and can be publicly disclosed. Figures 3(b) and 3(c) show the distribution of the distance when using individual masks and a common mask respectively. By using a common mask, built by setting $\lambda = 0.8$, the overlap between the two distributions increases. Anyway the best result with individual masks are obtained by using a similarity threshold ε between the iris templates equal to 0.41, providing a False Accept Rate (FAR) equal to 0.53% and a False Reject Rate (FRR) equal to 0.54%. By using a common mask, the best FAR and FRR are 1.44% and 1.47% obtained with $\varepsilon = 0.43$, resulting in an accuracy loss lower than 1%. The protocol has been tested on two different datasets, one

containing iriscodes represented with 2048 bits and the other containing iriscodes represented with 9600 bits. By using a common mask, a speedup factor of up to 8.7 can be achieved in the first dataset and of up to 4.7 in the second one. In both cases the bandwidth is reduced by a factor ~ 4.3. As reported in the original paper the online time for an iris match is 65 msec and requires the transmission of 133.7 KBytes.

Another example of algorithmic optimization has been proposed in the SHE-based face recognition protocol described in [21]. The authors use a Gabor filter (a linear filter used for edge detection) to build the feature vector. To minimize the amount of data to be processed, they discard the phase information and use a novel statistical characterization to model the magnitude of Gabor coefficients. Moreover, coefficient representation does not rely on quantization as usual, but is obtained by dividing the probability density function into 2^ℓ numbered sections. A coefficient is represented through the index of the segment it belongs to. The authors compared the performance of such indexing procedure with classical quantization-based schemes while varying the coefficient bitlength. Experiments were run on several databases. Results obtained on the XM2VTS dataset [42] show that 4 bits are sufficient to produce a much better fit, equalling the original performance of [42] ($\sim 96\%$) when using a Support Vector Machine (SVM) implemented as a weighted distance, while the accuracy decreases by $\sim 3\%$ if no SVM is used. On the other hand, the server runtime increases from 59 to 120 sec when an SVM is used.

B. Feature and distance choice

The choice of the features used to represent the biometric templates has a major impact on the complexity of SPED biometric matching protocols, due to the strict correlation between the type of features used to represent the biometric signals and the distance function used to evaluate the match. Let us consider, for example, fingerprint matching. The most popular and efficient matching algorithms are based on minutiae. However, in [34], [33] the authors chose the fingercode representation. In fact, even if the experiments show that filter-based matchers such as the fingercode tend to perform slightly worse than state-of-the-art minutiae-based matchers, the fingercode matching function has a much lower computational complexity and is more suitable for being implemented in a STPC setting. On the contrary, a privacy preserving protocol operating on minutiae would be difficult to implement, mainly due to the variable length of the feature vector and the lack of a simple distance measure between minutiae features. The intuition of [33], [34] has been later validated in [30], wherein a hybrid implementation of both fingercode



(a) Large illumination variation.

(b) Near-frontal changes in pose, mild facial expressions and mild illumination changes

Fig. 4: Robustness of SCiFI protocol [26] compared to Eigenface [19].Tests performed on the ORL *Database of Faces* from AT&T Laboratories Cambridge [24].

and minutia based identification protocols is described. As shown in Table II, the runtime of the protocol based on minutiae is hundred times higher than that of the fingercode protocol.

Another example of protocol simplification through feature selection is the SCiFI protocol for face recognition [26]. The representation used by SCiFI is based on the idea of composing a face as a collection of fragments taken from a dictionary of facial features. The resulting feature vector consists of two parts: the first part with the indexes of the dictionary fragments that better represent the face, the second one with the position of each part with respect to the face center. The feature vector is then represented as a fixed length binary vector and matching is carried out by relying on the Hamming distance. Authors compared SCiFI with eigenface-based recognition [19] by evaluating its robustness to various factors such as large illumination variation and near-frontal changes in pose, mild facial expressions and mild illumination changes. The results shown in Figure 4, where the recognition rate is plotted as a function of the false positive rate, demonstrate that is possible to improve the accuracy of the face recognition protocol, while, thanks to extensive precomputation, the online execution time required for the match of a query and a face in the database is reduced to about 0.31 second.

C. Feature vector size and representation accuracy

A further simplification can be obtained by decreasing the number of features used to represent the biometric template and the number of bits used to represent each feature.

A first example of such an approach is the HE face recognition protocol proposed by Erkin et al [19]. The signal processing analysis is limited to the definition of the scaling factor used to quantize the parameters of the protocol (which in turns determines the number of bits used

to represent the parameters and hence the accuracy of the representation) and the number k of features used to represent a face. The authors aimed at obtaining the same classification accuracy provided by a standard plain implementation, namely a correct recognition rate equal to 96%. As shown in Figure 5, such a goal is reached with a scaling factor ~ 1000. Moreover, experiments proved that no improvement is observed by using $k > 12$. By relying on such an analysis, the authors show that matching a face image against a database of 320 biometrics takes roughly 40 seconds and requires the transmission of 7249 kBytes (see Table I).

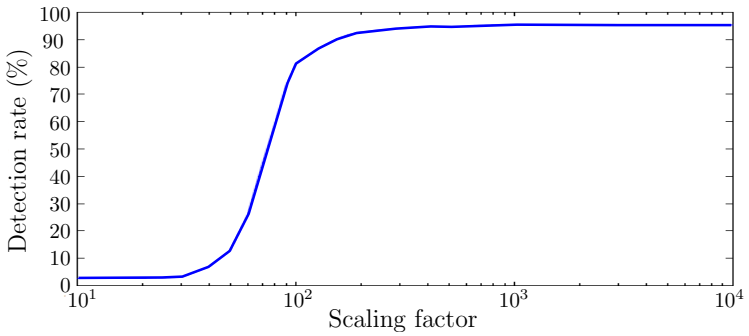


Fig. 5: Correct detection rate vs representation accuracy in the Face recognition system described in [19].

A more accurate signal processing analysis has been performed in the fingerprint recognition protocol described in [33]. Considering that a protocol computing the squared Euclidean distances on 640 features would have a very high complexity, the authors checked if a lower number of features can be used without degrading significantly the matching accuracy and selected the minimum number of bits necessary to represent each feature. To this purpose, the matching algorithm was tested by using 8 different fingercode configurations (Table V) and by varying the feature bitlength between 1 and 8. Figure 6 shows the behavior of the Equal Error Rate (EER) on the test set. As highlighted in the figure, it is evident that the accuracy of the system does not improve significantly when more than 96 features, each represented with 2 bits, are used. At the same time, the EER increases when only 1 bit is used for the representation, thus impeding the use of a more efficient protocol based on the Hamming distance. By the light of the above considerations, the authors chose to focus on configurations C and D, with 2 or 4 bits for feature representation. The results obtained in [33] are reported in Table VI. Moving from 192 features to 96 features and halving the number of bits, we observe a significant simplification of the protocol, with only a minor decrease of matching accuracy.

To improve the efficiency of a protocol, it is also possible to work on the representation of intermediate values. For example in the HE and GC hybrid protocols described in [36], the authors modify the protocol in order to use a more compact representation of intermediate distances. They

Configuration	features
A	640
B	384
C	192
D	96
E	48
F	32
G	16
H	8

TABLE V: Configuration for feature size reduction in fingercode protocol [33].

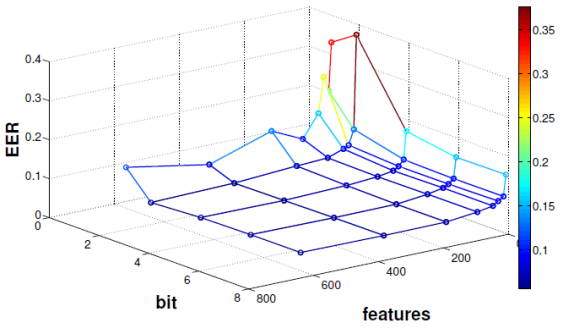


Fig. 6: Equal Error Rate of the different configurations of Fingercode [33] on the fingerprint database [43].

Configuration	Feature bitlength	EER	Bandwidth (bits) 408 entries	Runtime (sec) 100 entries
C	2	0.0715	6902008	44.43
	4	0.0673	8135800	53.66
D	2	0.0758	6568792	37.43
	4	0.0732	7802584	45.58

TABLE VI: Performance of privacy preserving Fingercode protocol [33].

assume that the acceptance threshold and its bitlength κ are publicly known. After computing a distance by means of an HE protocol, they start the GC section by checking if the distance is greater than 2^{κ}. In this case the distance value is replaced with the threshold. In such a way the minimum selection circuit can operate on shorter values hence reducing the total number of gates (results are given in Table III).

V. Conclusion

As shown throughout the present work, processing biometric signals in the encrypted domain provides an elegant and provably secure mechanism to protect both the biometric data and the privacy of the individuals subject to biometric controls. Thanks to the use of STPC cryptographic primitives, in fact, biometric matching algorithms can be implemented in such a way that the parties involved in the matching do not get access to either the data owned by the other party or the result of the match. From a decade of research in the field, it is now well evident that the question is not whether a certain computation can be carried out in the encrypted domain, but whether such a computation can be carried out efficiently.

While the quest for efficiency has driven the agenda of researchers in the last years, research has been mainly focused on the development of more efficient STPC primitives and their use to implement conventional biometric matching algorithms in a SPED framework. We believe, though, that significant advantages can also be obtained by working at the signal processing

level or, even better, by jointly considering the cryptographic and signal processing facets of the problem. It was the goal of this paper to introduce the readers to the main concepts behind SPED biometric matching and to show how a clever design of the underlying matching protocol may help to fill the gap between the complexity of SPED protocols and the efficiency required for the deployment of such protocols in real systems. We hope that the readers appreciated our effort and will contribute to the future advancement of this exciting field.

REFERENCES

[1] R. Lagendijk, Z. Erkin, and M. Barni, "Encrypted signal processing for privacy protection: Conveying the utility of homomorphic encryption and multiparty computation," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 82–105, 2013.

[2] O. Goldreich, "Secure multi-party computation," *Manuscript.*, 1998.

[3] O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game," in *Proceedings of the 19th annual ACM Symposium on Theory of Computing*. ACM, 1987, pp. 218–229.

[4] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[5] A. C. Yao, "How to generate and exchange secrets," in *Proceedings of the 27th Annual IEEE Symposium on Foundations of Computer Science*, 1986, pp. 162–167.

[6] V. Kolesnikov, A.-R. Sadeghi, and T. Schneider, "How to combine homomorphic encryption and garbled circuits," in *Signal Processing in the Encrypted Domain-First SPEED Workshop-Lousanne*, 2009.

[7] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the 41st annual ACM Symposium on Theory of Computing*. ACM, 2009, pp. 169–178.

[8] P. Campisi, *Security and Privacy in Biometrics*. Springer, 2013.

[9] J. Bringer, H. Chabanne, and A. Patey, "Privacy-preserving biometric identification using secure multiparty computation: An overview and recent trends," *IEEE Signal Processing Magazine*, vol. 30, no. 2, pp. 42–52, 2013.

[10] Y. Luo, S. S. Cheung, T. Pignata, R. Lazzeretti, and M. Barni, "An efficient protocol for private iris-code matching by means of garbled circuits," in *19th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2012, pp. 2653–2656.

[11] C. Fontaine and F. Galand, "A survey of homomorphic encryption for nonspecialists," *EURASIP Journal on Information Security*, vol. 2007, 2007.

[12] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology–EUROCRYPT 99*. Springer, 1999, pp. 223–238.

[13] I. Damgard, M. Geisler, and M. Kroigard, "Homomorphic encryption and secure comparison," *International Journal of Applied Cryptography*, vol. 1, no. 1, pp. 22–31, 2008.

[14] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," in *Advances in Cryptology*. Springer, 1985, pp. 10–18.

[15] P. S. Pisa, M. Abdalla, and O. C. M. B. Duarte, "Somewhat homomorphic encryption scheme for arithmetic operations on large integers," in *Global Information Infrastructure and Networking Symposium (GIIS), 2012*. IEEE, 2012, pp. 1–8.

[16] M. O. Rabin, "How to exchange secrets by oblivious transfer," Technical Report TR-81, Aiken Computation Laboratory, Harvard University, Tech. Rep., 1981.

[17] D. Beaver, "Precomputing oblivious transfer," in *Advances in Cryptology – CRYPT0'95*. Springer, 1995, pp. 97–109.

[18] R. Lazzeretti and M. Barni, "Private computing with garbled circuits [applications corner]," *Signal Processing Magazine (SPM), IEEE*, vol. 30, no. 2, pp. 123–127, 2013.

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

[20] J. Bringer, H. Chabanne, M. Favre, A. Patey, T. Schneider, and M. Zohner, "GSHADE: faster privacy-preserving distance computation and biometric identification," in *Proceedings of the 2nd ACM workshop on Information Hiding and Multimedia Security*. ACM, 2014, pp. 187–198.

[21] J. Troncoso-Pastoriza, D. Gonzalez-Jimenez, and F. Perez-Gonzalez, "Fully private noninteractive face verification," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 7, pp. 1101–1114, July 2013.

[22] J. Bringer, H. Chabanne, D. Pointcheval, and Q. Tang, "Extended private information retrieval and its application in biometrics authentications," in *Cryptology and Network Security*. Springer, 2007, pp. 175–193.

[23] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 1991, pp. 586–591.

[24] "The database of faces, (formerly "the ORL database of faces") AT&T Laboratories Cambridge," http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

[25] A. Sadeghi, T. Schneider, and I. Wehrenberg, "Efficient privacy-preserving face recognition," in *Information, Security and Cryptology–ICISC 2009*. Springer, 2010, pp. 229–244.

[26] M. Osadchy, B. Pinkas, A. Jarrous, and B. Moskovich, "SCiFi - a system for secure face identification," in *IEEE Symposium on Security and Privacy (SP)*. IEEE, 2010, pp. 239–254.

[27] Y. Luo, S.-c. S. Cheung, and S. Ye, "Anonymous biometric access control based on homomorphic encryption," in *IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2009, pp. 1046–1049.

[28] J. Daugman, "How iris recognition works," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 21–30, 2004.

[29] T. Tan and Z. Sun, "Casia-irisv3," *Chinese Academy of Sciences Institute of Automation, http://www. cbsr. ia. ac. cn/IrisDatabase. htm, Tech. Rep*, 2005.

[30] M. Blanton and P. Gasti, "Secure and efficient protocols for iris and fingerprint identification," in *Computer Security–ESORICS 2011*. Springer, 2011, pp. 190–209.

[31] J. Bringer, M. Favre, H. Chabanne, and A. Patey, "Faster secure computation for biometric identification using filtering," in *5th IAPR International Conference on Biometrics (ICB)*. IEEE, 2012, pp. 257–264.

[32] A. K. Jain, L. Hong, S. Pankanti, and R. Bolle, "An identity-authentication system using fingerprints," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1365–1388, 1997.

[33] M. Barni, T. Bianchi, D. Catalano, M. Di Raimondo, R. Donida Labati, P. Failla, D. Fiore, R. Lazzeretti, V. Piuri, A. Piva, and F. Scotti, "A privacy-compliant fingerprint recognition system based on homomorphic encryption and fingercode templates," in *4th IEEE International Conference on Biometrics: Theory Applications and Systems (BTAS)*. IEEE, 2010, pp. 1–7.

[34] M. Barni, T. Bianchi, D. Catalano, M. Di Raimondo, R. Donida Labati, P. Failla, D. Fiore, R. Lazzeretti, V. Piuri, F. Scotti, and A. Piva, "Privacy-preserving fingercode authentication," in *Proceedings of the 12th ACM workshop*

on Multimedia and security. ACM, 2010, pp. 231–240.

[35] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, *Handbook of fingerprint recognition*. Springer Science & Business Media, 2009.

[36] D. Evans, Y. Huang, J. Katz, and L. Malka, "Efficient privacy-preserving biometric identification," in *Proceedings of the 17th Conference on Network and Distributed System Security Symposium, NDSS*, 2011.

[37] S. Goldwasser, S. Micali, and A. Wigderson, "How to play any mental game, or a completeness theorem for protocols with an honest majority," in *Proceedings of the 19th Annual ACM Symposium on Theory of Computing*, vol. 87, 1987, pp. 218–229.

[38] C. Gentry and S. Halevi, "Implementing Gentry's fully-homomorphic encryption scheme," *Advances in Cryptology–EUROCRYPT 2011*, pp. 129–148, 2011.

[39] M. Yasuda, T. Shimoyama, J. Kogure, K. Yokoyama, and T. Koshiba, "Packed homomorphic encryption based on ideal lattices and its application to biometrics," in *Security Engineering and Intelligence Informatics*. Springer, 2013, pp. 55–74.

[40] C. Blundo, E. De Cristofaro, and P. Gasti, "EsPRESSo: efficient privacy-preserving evaluation of sample set similarity," in *Data Privacy Management and Autonomous Spontaneous Security*. Springer, 2013, pp. 89–103.

[41] G. Fanti, M. Finiasz, G. Friedland, and K. Ramchandran, "Toward efficient, privacy-aware media classification on public databases," in *Proceedings of International Conference on Multimedia Retrieval*. ACM, 2014, p. 49.

[42] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre, "XM2VTSDB: The extended M2VTS database," in *2nd International Conference on Audio and Video-based Biometric Person Authentication*, vol. 964. Citeseer, 1999, pp. 965–966.

[43] "Neurotechnology, dataset cross match verifier 300," http://www.neurotechnology.com.