Privacy-preserving biometric authentication: challenges and directions

Elena Pagnin, Aikaterini Mitrokotsa
Chalmers University of Technology
Gothenburg, Sweden
{elenap, aikmitr}@chalmers.se

May 23, 2017

Abstract

An emerging direction for authenticating people is the adoption of biometric authentication systems. Biometric credentials are becoming increasingly popular as a mean of authenticating people due to the wide rage of advantages that they provide with respect to classical authentication methods (*e.g.,* password-based authentication). The most characteristic feature of this authentication method is the naturally strong bond between a user and her biometric credentials. This very same advantageous property, however, raises serious security and privacy concerns in case the biometric trait gets compromised. In this article, we present the most challenging issues that need to be taken into consideration when designing secure and privacy-preserving biometric authentication protocols. More precisely, we describe the main threats against privacy-preserving biometric authentication systems and give directions on possible countermeasures in order to design secure and privacy-preserving biometric authentication protocols.

1 Introduction

Biometric authentication is a quick, accurate and user-friendly tool that offers an efficient and reliable solution in multiple access control systems. A typical example of biometric authentication systems (BAS) are access control systems equipped with sensors (*e.g.,* for iris or fingerprint scans). In this case, the sensor captures the biometric trait of the person who requests access, while access is granted only after the person has been recongnised as an authorised user of the system. One of the main advantages of biometrics is that they do not require to memorise complicated passwords or carry tokens along since they cannot be forgotten or lost.

While BAS provide important usability advantages, they are susceptible to threats, like any other security system. For biometric authentication, however, a successful attack can have severe implications in the users' lives and privacy. Unlike passwords or tokens, biometric credentials cannot be kept secret or hidden, and *stolen* biometrics cannot be revoked as easily [3]. Thus, the risk of them being compromised (*i.e.,* captured, cloned or forged) is high and may lead to identity theft or individual profiling and tracking in case the templates are used and cross-matched in different biometric databases. In addition, stolen biometrics can be used to learn sensitive information about their owners, such as ethnic group, genetic information [30], medical diseases [7] or even to perform illegal activities by compromising health records [23]. It is therefore of fundamental importance to develop *privacy-preserving* BAS, *i.e.,* biometric authentication systems that can mitigate the aforementioned privacy and security risks listed.

In this article, we present the main challenges in achieving privacy-preserving biometric authentication and we highlight the main threats associated to privacy issues. Furthermore, we describe the main countermeasures to prevent the information leakage in biometric authentication as well as novel possible directions for the design of efficient privacy-preserving biometric authentication protocols.

Paper Organisation: Section 2 describes how biometric authentication works and the challenges encountered to achieve accurate biometric authentication. It also explains the main differences between privacy-preserving and non privacy-preserving systems. The main threats against privacy-preserving BAS are described in Section 3. A particular emphasis is given to biometric reference recovery attacks as well as biometric sample recovery attacks. Section 4 collects suggestions for possible mitigations and countermeasures against the attacks described in Section 3. Eventually, Section 5 concludes the paper.

2 Preliminaries on Biometric Authentication Systems

Generally speaking, a biometric authentication system works in the following way. First, a user (*e.g.,* an employee) registers to the system by providing her identity together with her biometric template, that becomes her *reference* template (registration phase). Subsequently, the user can get authenticated into the system (authentication phase) by submitting an identity and a biometric template, called *fresh* template. The system performs a *matching process*, which aims to check if the provided fresh template is *close enough* to the one stored for the given user [24] (in which case the user is *authenticated / accepted*) or not (in which case the user is *rejected*). Standard BAS aim at authenticating users regardless of what the system may leak about the user's biometric credentials to third parties. In contrast, *privacy-preserving* BAS provide user authentication without revealing any information about the client's biometric data, not only to third parties but also to the system itself.

The base for biometric authentication is the extraction of a biometric trait from the human body or behaviour. Common biometric traits used nowadays for authentication are: voice, signature, DNA, fingerprint [37], iris [13], and ear shape [21]. In all cases, the biometric trait is a distinctive characteristic that is measurable and identifies (almost) uniquely each individual. In practice, the data collecting process of biometric templates is by itself a challenging task due to the inherent *noise* and the natural variability of biometric credentials [22]. For example, two scans of the same fingerprint can differ because of the variance in finger pressure, orientation, dirt or sweat [14]. To overcome the presence of noise, which is inherited in biometric credentials and in the collection process, the comparison between a fresh biometric template and a stored one always takes into account *approximation*.

In order to understand how biometric authentication is performed, and subsequently discuss what attacks and mitigations are possible, we need to formally present the two main phases that compose a privacy-preserving BAS. Figure 1 depicts the authentication phase for a distributed architecture [4, 20], *i.e.,* where every entity involved in the authentication process performs only a single task. More precisely, by adopting a distributed architecture in the biometric authentication process (*e.g.,* computational server ($\mathcal{CS}$), authentication server $\mathcal{AS}$, database $\mathcal{DB}$), it is possible to limit the amount of information each entity has at its disposal and thus avoid single point of failures. Furthermore, a distributed architecture provides higher privacy guarantees since no single entity has access to all sensitive data (*i.e.,* fresh biometric template, stored biometric template, user's identity).

This architecture is adopted as a security countermeasure against internal honest-but-curious adversaries. In most systems, even if one entity among $\mathcal{CS}$, $\mathcal{DB}$ and $\mathcal{AS}$ is corrupted, an adversary (malicious third party) cannot learn anything about the biometric templates unless it behaves maliciously. In non-distributed architectures the computational server and the authentication server merge into a single entity, leading to a single point of failure.

The enrolment phase: This phase takes place only once and is performed before the authentication. A user $\mathcal{C}$ (client) registers to a trusted party her biometric template (usually encrypted in a digital string $\widetilde{b}$) along with her identity (possibly a pseudonym $\widehat{\mathsf{ID}}$). These two data are then stored in the database $\mathcal{DB}$ of the authentication system. Once enrolled in the system, the client can authenticate herself an unlimited number of times.

The authentication phase: This phase is depicted in Figure 1. The client provides her fresh biometric trait (through the sensor $\mathcal{S}$) together with her identity. These two pieces of information are then elaborated by the sensor, and transmitted to the computational server $\mathcal{CS}$, as $\widetilde{b'}$ (*e.g.,* the encryption of the fresh template) and $\widehat{\mathsf{ID}}$ (*e.g.,* a pseudonym). The computational server $\mathcal{CS}$ queries the database $\mathcal{DB}$ for the stored template $\widetilde{b}$ linked to $\widehat{\mathsf{ID}}$. After receiving $\widetilde{b}$, $\mathcal{CS}$ computes the (possibly encrypted) distance d between b' and b (*e.g., d* could be the Euclidean or the Hamming Distance). Let $\Delta = \widetilde{d(b, b')}$ be the output that $\mathcal{CS}$ sends to $\mathcal{AS}$. The authentication server uses Δ to derive the actual distance between b' and b, and compares it with τ, the threshold of the system. The threshold τ can be thought as the accuracy level of the system, indeed, if the templates are close enough (*i.e.,* $d(b, b') < \tau$), the user is authenticated, otherwise the user is rejected.

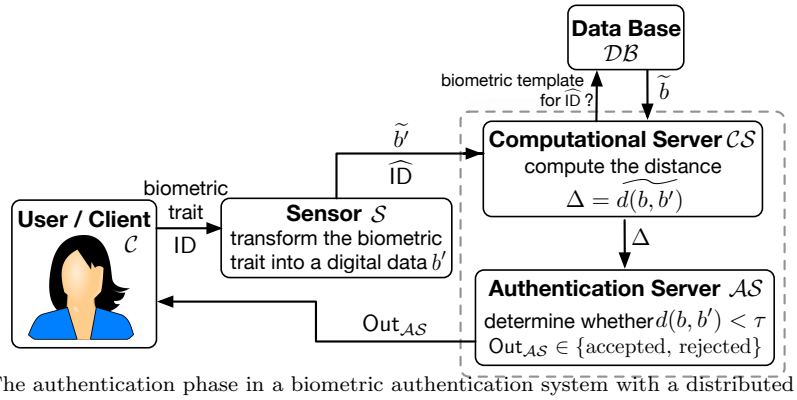

Figure 1: The authentication phase in a biometric authentication system with a distributed architecture.

In classical authentication systems (*i.e.,* non privacy-preserving), the biometric data is sent and stored in the clear. In this case, $\widetilde{b'} = b'$, $\widetilde{b} = b$ and $\widetilde{\mathsf{ID}} = \mathsf{ID}$. In these systems an eavesdropper adversary can easily retrieve the biometric templates of any user.

In contrast, *privacy-preserving* biometric authentication systems aim at protecting the users' biometric templates against both passive and active adversaries. A common practice is to preserve the user's privacy by *encrypting* the sensitive data. For example, Yasuda *et al.*'s privacy-preserving biometric authentication scheme works as follows [36]. The sensor $\mathcal{S}$ encrypts the provided fresh biometric template b' obtaining $\widetilde{b'} = \mathsf{Enc}(b')$ (here the encryption scheme is based on a packing method for polynomials). For privacy reasons also the reference template b is stored encrypted as $\widetilde{b} = \mathsf{Enc}(b)$. The computational server computes Δ, which is the encrypted Hamming Distance of the two templates, and forwards it to $\mathcal{AS}$. The authentication server decrypts Δ and checks whether the distance is less than the predefined threshold τ. In the protocol outlined above, the biometric templates are always handled in an encrypted way. The only entity in possess of the decryption key is $\mathcal{AS}$, which never receives an encrypted template, but only encrypted distances.

3 Main Threats against Privacy - Preserving Biometric Authentication Systems

Attacks against privacy-preserving biometric authentication systems aim at learning information about the user's biometric trait or identity. What we describe in this section are attack strategies and goals connected to security and privacy issues that have severe impact in users' lives, especially considering the irrevocability of biometrics templates [3]. Below, we list the four main threats that afflict privacy-preserving biometric authentication systems [34].

1. **Biometric sample recovery.** In this case, the goal of the adversary is to determine a fresh biometric template b' which is accepted by the authentication server. The consequences of a successful attack are similar to the reference recovery attack, apart from the fact that the produced *matching* template may differ from the user's real one, and so the adversary can recover less information regarding the user's private information (*e.g.,* physical characteristics, DNA etc.).

2. **Biometric reference recovery.** A non-authorised party (usually called the *adversary*) succeeds in recovering the (plain-text) reference biometric template b. This is the most harmful threat since by recovering the reference template the adversary may gain unauthorised access to any system that uses b as a reference template, and also collect sensitive information about the user's physical characteristics and health.

3. **User's traceability.** An unauthorised party (*e.g.,* the adversary) is able to trace a user's authentication attempts over different applications. Consequences of a successful traceability attack are cross-matching, profiling and tracking of individuals.

4. **User's distinguishability.** The adversary recovers the link between a biometric template b, or b', and a user identity ID. Compromising this relation may lead to the disclosure of more sensitive information and often breaks the anonymity of the system.

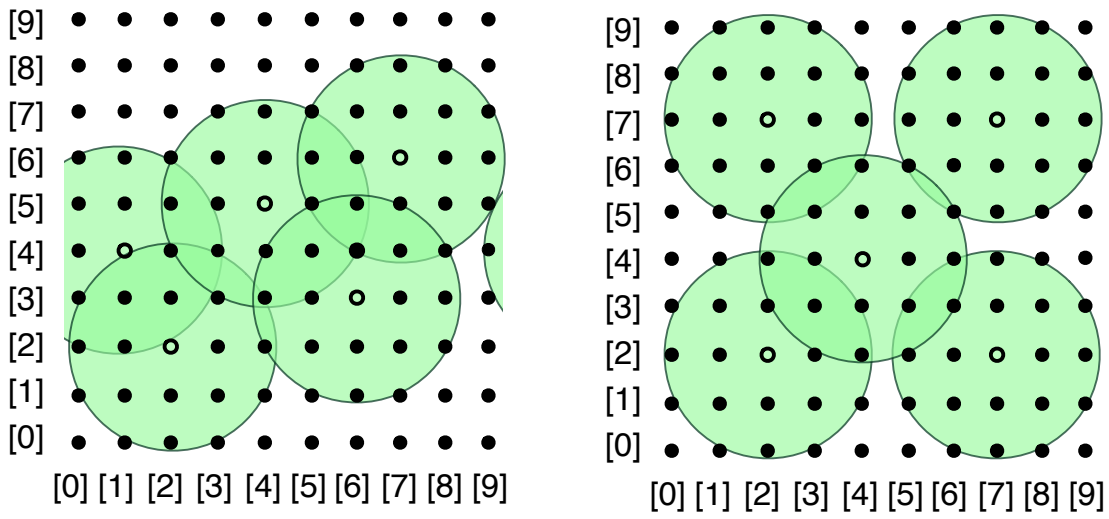

Figure 2: An intuitive example of what the set-covering problem is. The aim is to cover the largest possible area in the space $\mathbb{Z}_{10}^2$ using 5 circles. On the left hand side, the centres of the circles are chosen at random (the covered area is less than 70%), whilst on the right hand side, we provide a better covering of the space (the covered area is about 85%). Finding the *optimal* covering corresponds to solving an NP-had problem for large dimensions of the space.

3.1 Biometric sample recovery attacks

Biometric sample recovery attacks are performed in two main ways: via template *spoofing* (*e.g.,* extracting the fingerprint left on a glass) or via *brute-force* techniques. The most common way to bypass a BAS is by using a *spoof* of a biometric trait. A spoof refers to a fake or an artificial biometric template that does not correspond to a live person. These include for instance, gummy fingers, residual fingerprint impressions of legitimate users, photographs of legitimate users or voice recordings of legitimate users. The only alternative to these *practical* techniques, is to estimate a valid biometric sample using *brute-force* strategies. Below, we list the possible brute-force strategies that could be adopted in recovering a valid biometric template [29]. Luckily, all the approaches run in exponential time and thus most of the current biometric authentication systems are secure.

In the following, we assume that the adversary can see the result of the authentication process $\mathsf{Out}_{\mathcal{AS}}$ at each trial, and that the templates are binary vectors. Binary representation of biometric traits is not far from reality since this is the case for biometric authentication based on iris templates [13].

Blind brute-force. The easiest algorithm to find a matching template from scratch is the blind brute-force. In this case, the attacker picks biometric traits at random. This corresponds to randomly selecting and trying biometric templates from the available space (*i.e.,* $b' \xleftarrow{R} \mathbb{Z}_q^n$) until one template gets accepted by the system.

Set-Covering. This attack strategy represents the *optimal* brute-force solution: pick a random trial template from the set of potential candidates (which at the beginning is the whole space $\mathbb{Z}_q^n$). If the trial template is rejected, remove from C all the points that are within τ distance from it, and pick another point at random from the updated set C. Although this method possibly eliminates from C some of the matching points (if the trial templates are picked with a distance of 2τ one from the other) if such an algorithm exists and was efficient, it would be exponentially fast in finding a matching template. Such an algorithm could also be used to solve the set-covering problem, which is known to be NP-Hard [11]. An intuition of this geometrical challenge is given in Figure 2. The points on the plane are biometric templates, the trial samplings are the centres of the green circles. The green circles delimit the acceptance region around the tried point and have radius equal the threshold τ of the system. Greedy approximations to the optimal solution of the set-covering problem are reachable in an efficient way, in which case the number of trials the adversary needs to perform is only a factor of $O(\tau \ln(n+1))$ more than the optimal cover.

3.2 A biometric reference recovery attack

The most successful strategy to perform a biometric reference recovery attack is to use a *hill-climbing* technique [34] to perform a *centre search attack* [34]. The attack can be launched under three conditions [1, 2, 29]:

1. The adversary is in possess of a matching template (maybe spoofed) for the target biometric reference.
2. The adversary is able to see the output of the authentication process ($\mathsf{Out}_{\mathcal{AS}}$). For instance, this information could be in an access control system a door that is opening.
3. The matching process between a fresh and a stored template relies on specific distances, called *leaking distances*, which include the Euclidean and the Hamming Distance.

Figure 3 provides an intuition of the attack strategy. In the example (Figure 3) the stored reference template is the point $b = (6, 3)$ and the given matching b' is in the point $(6, 4)$. The matching templates are the points in the region delimited by the green circle. The adversary starts from the first component of the given matching template, the point $(6, 4)$, and increments it repeatedly by a factor 1. When rejected, on the point $(9, 4)$ denoted by the red bullet with a white cross, the attacker learns that the *previous* point is the last one inside the acceptance circle. The same strategy is repeated starting from the point b' and decreasing (by a factor 1 each time) the first component until rejection, and for the other component of the template. After discovering the coordinates of the four boundary points in the acceptance circle, the attacker can compute the coordinates of its centre, *i.e.,* find the digital representation of the biometric reference template.

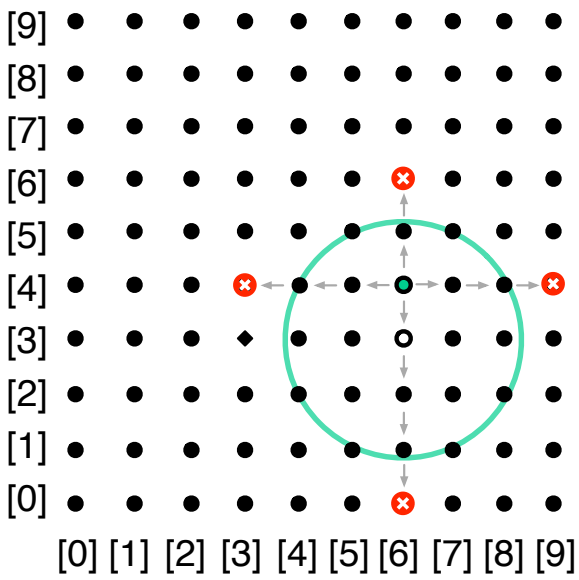

Figure 3: Example of a recovery template attack for a BAS with biometric traits represented as vectors in $\mathbb{Z}_{10}^2$ and with threshold $\tau = 2$. The values are chosen ad hoc to be able to picture the example in an easy and intuitive way and do not reflect the parameters used in real applications (usually, q is smaller than n and $n \gg \tau$ is in the order of 2048).

This reference recovery attack is very efficient as it only requires a number of authentication attempts that is linear in the length of the biometric template [29]. Moreover, it can be mounted against many biometric authentication systems (privacy-preserving or not), and even systems that employ secure multi-party computation techniques including somewhat homomorphic encryption [2].

Another, strategy to perform biometric reference recovery attacks is to gain access to the database and try to decrypt the target template. This approach, however, is way less successful since normally the employed cryptographic techniques used to protect the templates' privacy are proven to be secure.

3.3 User Traceability and Distinguishability

Generally speaking, attacks against the user's privacy (in the sense of traceability and distinguishability) do not aim at gathering information about the user's biometric credential in itself, but rather at profiling and identifying the target user among all the users of one or more biometric systems.

The main attack strategy to trace users in privacy-preserving BAS is the following. The attacker gets access to different databases (possibly in use by different biometric authentication systems) and successfully *traces* a user's authentication attempts, by checking which record of the database is queried (as match for the authentication). Note that the above approach does not require the attacker to know the user's credential, as long as the databases store the biometric credentials *in the same way* (*i.e.,* using the same encryption mechanism and the same secret key). Luckily, in real life, this is a very strong assumption which happens only seldomly [34].

In simple words, user distinguishability can be considered as user tracing over different authentication attempts in the same or different authentication systems. That is, the attacker can recognise the *target user* among the other users present in the the biometric authentication system. This attack is always successful if the attacker learns the mapping from the set of identities to the set of (encrypted) templates. In other words, an attacker can distinguish users if he learns that to a certain identity ID corresponds a certain (possibly encrypted) template b. A solution would be to keep the mapping $\mathsf{ID} \mapsto b$ secret, or to use a (secure) pseudo-random mapping. Another possibility is to ensure that the communication channels between the entities involved in the BAS are secure, or that the information transmitted is encrypted using chosen plaintext attacks (CPA)-secure systems.

We present more detailed explanations of methods to achieve user privacy in biometric authentication in the next section.

4 Challenges & Countermeasures

The main question that one needs to address when designing a privacy-preserving biometric authentication protocol is: *How to guarantee privacy-preservation without downgrading the accuracy of a biometric authentication system?*
Among the most challenging problems in designing efficient and privacy-preserving biometric authentication systems there are: (1) the resistance to impersonation attacks; (2) the irrevocability of biometric templates; and (3) guaranteeing that personal information remains private. In the following, we provide a list of methods that have been used to achieve privacy-preserving authentication, and we highlight the main advantages and disadvantages of each approach.

4.1 Biometric template protection

Most existing privacy-preserving biometric authentication approaches focus on storing and transmitting a modified version of the original biometric templates in order to avoid the danger of eavesdropping sensitive data or the case of compromised databases. One direction in order to combat the privacy issues associated with biometric authentication, is the employment of *biometric template protection* schemes such as *cancellable biometrics* and *biohashing*. Example of cancellable fingerprints were proposed by Ang and McAven [3], while Kanade and Dorizzi [26] proposed cancellable iris biometrics. Different biohashing schemes are presented in [33]. Although biohashing offers low error rates while guaranteeing a quick authentication phase, biohashing schemes are vulnerable to several attacks [27, 28].

4.2 Cryptographic Primitives

The direct employment of cryptographic primitives seems the most robust approach so far to tackle the challenging problem of privacy-preservation. Most of the state-of-the-art cryptographic protocols, however, were not designed taking into consideration the inherent variability of biometric data. In fact, cryptography tends to amplify small differences and it is not error-tolerant (*e.g.,* hashing, AES, RSA). The main cryptographic tools used to to combat the leakage of private information during biometric authentication are: *secure multi-party computation* (SMPC) [36], *Verifiable Computation* (VC) and *Bloom Filters* (see Box 2).

Secure Multiparty Computation in Biometric Authentication

Cryptographic primitives that are often employed in SMPC include: *Homomorphic Encryption*, *Obvilious Transfer* and *Garbled Circuits*, which will be presented shortly, and are often combined to obtain privacy-preserving BAS [18]. From a theoretical point of view, SMPC techniques allow to maximise the utility of information without compromising the user privacy. A more formal intuition on how SMPC works is given in Box 1.

Box 1: The general setting for SMPC is the following. The system is made up of N entities $p_1, p_2, \ldots, p_N$ that jointly compute some public function f based on some individually secret data $d_1, d_2, \ldots, d_N$, without revealing their private inputs to one another. In other words, SMPC allows the interactive computation among multiple parties in such a way that at the end of the process no participant p_i can learn more from f and the result $D = f(d_1, d_2, \ldots, d_N)$, than what p_i could learn from her own secret data d_i, $i = 1, 2, \ldots, N$. It is easy to see that SMPC can be very useful in privacy-preserving biometric authentication especially in the distributed scenario, where multiple entities are involved in the authentication process [4] (*e.g.,* a database $\mathcal{DB}$, an authentication server $\mathcal{AS}$ and a matcher $\mathcal{CS}$). In this case, the function f could be the distance between the fresh and the stored biometric template and the goal would be to guarantee the secrecy of the biometric templates (fresh and stored).

It is understood that SMPC is an incredibly useful tool for the design of privacy-preserving biometric authentication protocols. Multiple existing schemes, indeed, rely on SMPC [4, 20].

Homomorphic encryption (HE) is perhaps the most suitable cryptographic primitive (inside the SMPC framework) that can be successfully employed for privacy-preserving biometric authentication [5, 36]. Homomorphic encryption can be applied in a bit-by-bit mode making it possible to perform the matching process in the encrypted domain directly [36]. More formally, HE allows to translate operations on the encrypted data (ciphertext) to some useful operations on the corresponding plaintexts. In formulas:

$$\mathsf{Enc}_k(m_1) \circ \mathsf{Enc}_k(m_2) = \mathsf{Enc}_k(m_1 \times m_2),$$

where m_1, m_2 are plaintext messages, and Enc_k corresponds to an homomorphic encryption function under a public key k. If we consider that $m_1 = b'$ is the fresh template of a user ID and $m_2 = b$ is the stored template of the same user, then homomorphic encryption gives us the possibility to perform operations on the encrypted templates and compute the distance (*e.g.,* Hamming distance) between them. While HE protects biometric templates from user traceability attacks[1], it does not directly protect from other privacy attacks. For instance, Abidin *et al.* [2] exploit exactly the homomorphic property to show that the claimed privacy-preserving BAS in [36] is actually vulnerable to the biometric template attack. Another limitation to the employment of HE schemes is their computational cost, and limitations on the number of multiplications that can be performed between ciphertexts. Nevertheless, some recently-proposed schemes [12, 19] show promise regarding the efficiency of HE.

Oblivious transfer (OT) (1-out-of-N) [31] enables one party the sender $\mathcal{S}$ to send one element out of N, to a receiver $\mathcal{R}$ in such a way that the sender does not know which element is received by $\mathcal{R}$. Furthermore, $\mathcal{R}$ does not find out anything about the other $N-1$ elements. If we consider the *elements* to be the stored (encrypted) biometric templates, we see that OT essentially allows one to *search* in the database, without revealing which item (*i.e.,* biometric template) is selected for the matching process. This is a very useful tool for privacy-preservation, and assures perfect resistance against user traceability and distiguishability [20, 9]. Similarly to HE, however, OT alone cannot prevent some template recovery attacks, since the best known strategy are based solely on the value returned by the BAS (essentially the acceptance/rejection message) which is not affected by the OT technique.

Garbled circuits are a cryptographic technique that enables two parties to compute a function (represented as a binary circuit) and learn only the output of the function and nothing else (*e.g.,* the other party's input) [35]. This approach combines OT and SMPC between two entities, and thus is quite relevant for achieving a privacy-preserving matching process in biometric authentication. Up to now, garbled circuits constitute the most promising cryptographic tool to prevent template recovery attacks. A detailed description of OT and garbled circuits in BAS can be found in [10].

[1] HE prevents user traceability given that different databases store different/independent encryptions of the same reference template.

Verifiable Computation in Biometric Authentication

Verifiable Computation (VC) techniques enable a client to outsource computations to a remote server in a secure way. After performing the calculations, the server returns to the client the result together with a proof asserting the correctness of the returned result (for the outsourced computation). The client only needs to check the proof to convince itself of the correctness of the returned output. At first it might appear that VC has little or no connections to biometric authentication, however the linking point lies in the need for outsourcing the matching process to a third party (*e.g.,* the computational server in the distributed architecture depicted in Figure 1). Incorporating VC in a BAS in a secure way, allows to speed-up the matching process, without introducing additional privacy leakage *e.g.,* it is harder to perform centre search attacks. Recently, Bringer *et al.* [8] showed how to apply the recent advances in verifiable computing to the main algorithms for biometric matching.

Box 2: *Bloom Filters in Biometric Authentication.* This method is the main alternative to the employment of leaking distances [6, 32]. Intuitively, a Bloom filter F_A is an N-bit string which represents a set $A \subseteq D$ (*e.g., A* the acceptance area, and D is the space of all biometric templates). The encoding of A into F_A is done using k independent hash functions $h_1, h_2, \ldots, h_k : D \to [0, N-1]$ in the following way. For each element $b \in A$, and for each $1 \leq i \leq k$, the bits at positions $h_i(b)$ of the Bloom filter F_A are set to 1 (the other bits are set to 0). To test if an element b' is in A using the bloom filter, it is sufficient to check whether the bits of F_A at positions $h_i(b')$ are equal to 1, for all $1 \leq i \leq k$. If this is the case, one can deduce that b' is in A with high probability, otherwise it holds $b' \notin A$. It is immediate to see that the employment of Bloom filters in the matching process directly mitigates any centre-search attack for template recovery.

4.3 Error Correcting based methods

The use of error correction codes is an attractive mitigation to the inherently noisy nature of biometric traits. Error correction, indeed, would automatically *decode* small perturbation of a template into the template itself, solving the problem of noisy data. In this way, the systems can get error-free biometric templates and thus successfully use cryptographic primitives that will not affect the matching biometric process. This is for instance the case for the *fuzzy commitment scheme* described by Juels and Wattenberg in [25]. The biometric template is used as a witness to commit to a secret codeword c. As long as the fresh witness provided by the client is *close* to the used one, it will *correct* to the same codeword c. The decoded codeword will then be used in the commitment scheme. Typically the witness is used as a key for the encryption/decryption and the user authentication. Such systems could handle efficiently the noisy nature of biometrics and subsequently cryptographic primitives (hashing and/or encryption) could be employed. From a theoretical point of view, these schemes are secure against biometric reference and sample template attacks. In order to recover either the biometric template or the key, an attacker should indeed know the user's biometric data. However, given that the biometric templates are not uniformly random, and practical error correcting codes do not have high correction capability, the theoretical security is not achievable in practice. It has been shown, indeed, that fuzzy commitment schemes leak private information [22].

4.4 Other non-cryptographic Approaches

Given that OT is a well-established countermeasure against user traceability and distinguishablility attacks, most non-cryptographic tools for privacy-preserving BAS focus to combat template and sample recovery attacks.

For instance, [29] suggests to combat *centre search* attacks by using weighted distances to compare the fresh template with the stored one, and to keep the weights secret and different for each user. This procedure is adopted by the biometric authentication protocols that employ the normalised Hamming distance [15] or the weighted Euclidean distance [38]. Even though the centre search attack might still be feasible also in these scenarios, it will only lead to the recovery of a subset of the components of the stored biometric template.

Another alternative is to generalise the comparison process to include multiple distances. More precisely, if the matching process relies on such a mechanism that, at each authentication attempt, a distance is randomly selected from a pre-defined set of distances. Thus, the attacker could not

gain any information about the stored template without knowing first which distance has been used.

Similarly, changing the value of the threshold τ used for the matching process at each authentication attempt, renders harder the implementation of the centre search attack. However, such approaches may have a negative impact on the accuracy of the biometric authentication and may increase the false acceptance and/or false rejection rates.

Finally, one could consider to combine Differential Privacy (DP) [17, 16] with biometric authentication, in order to achieve privacy preservation. Intuitively, DP allows users to query a database and receive *noisy* answers, so that no information in leaked about the data stored in the database. Although this combination of DP with biometric authentication could possibly give an end to template recovery attacks (*i.e.,* centre search attacks), it could also have an impact on the accuracy of the authentication process and thus, a more detailed analysis of the achieved utility (accuracy) and privacy-preservation needs to be performed.

5 Conclusions

This article discusses challenges in biometric authentication, with a particular focus on privacy-preserving ones. We highlight the main advantages of biometric authentication as well as the risks that it brings along. We then list the most dangerous threats against privacy-preserving BAS and discuss possible attack strategies to undermine the privacy of a BAS. Finally, we identify possible directions to mitigate the highlighted threats, providing both the advantages and the disadvantages of the proposed methods. The practicality of privacy-preserving biometric authentication systems is by itself a great motivation for finding solutions to the security and privacy challenges connected to the employment of biometrics in authentication systems.

References

[1] A. Abidin and A. Mitrokotsa. Security aspects of privacy-preserving biometric authentication based on ideal lattices and ring-LWE. In *Proceedings of the IEEE Workshop on Information Forensics and Security*, pages 1653–1658, 2014.

[2] A. Abidin, E. Pagnin, and A. Mitrokotsa. Attacks on privacy-preserving biometric authentication. In *Proceedings of the 19th Nordic Conference on Secure IT Systems (NordSec 2014)*, pages 293–294, Tromso, Norway, October 2014.

[3] R. Ang, R. Safavi-Naini, and L. McAven. Cancelable key-based fingerprint templates. *ACISP*, pages 242–252, 2005.

[4] M. Blanton and M. Aliasgari. Secure computation of biometric matching. *CSE Technical Report TR 2009-03, University of Notre Dame*, April 2009.

[5] M. Blanton and P. Gasti. Secure and efficient protocols for iris and fingerprint identification. In *European Symposium on Research in Computer Security*, pages 190–209. Springer, 2011.

[6] B. H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970.

[7] J. Bolling. A window to your health. *Jacksonville Medicine, Special Issue: Retinal Diseases*, 51, 2000.

[8] J. Bringer, H. Chabanne, F. Kraïem, R. Lescuyer, and E. Soria-Vázquez. Some applications of verifiable computation to biometric verification. In *Information Forensics and Security (WIFS), 2015 IEEE International Workshop on*, pages 1–6. IEEE, 2015.

[9] J. Bringer, H. Chabanne, and A. Patey. Shade: Secure hamming distance computation from oblivious transfer. Cryptology ePrint Archive, Report 2012/586, 2012.

[10] J. Bringer, H. Chabanne, and A. Patey. Privacy-preserving biometric identification using secure multiparty computation: An overview and recent trends. *IEEE Signal Processing Magazine*, 30(2):42–52, 2013.

[11] L. Chen. New analysis of the sphere covering problems and optimal polytope approximation of convex bodies. *Journal of Approximation Theory*, 133(1):134–145, March 2005.

[12] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachne. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. 2016.

[13] J. Daugman. How iris recognition works. *IEEE Transactions on Circuits and Systems for Video Technology*, 14:21–30, 2004.

[14] R. Derakhshani, S. A. Schuckers, L. A. Hornak, and L. O'Gorman. Determination of vitality from a non-invasive biomedical measurement for use in fingerprint scanners. *Pattern recognition*, 36(2):383–396, 2003.

[15] A. K. Dewangan and M. A. Siddhiqui. Human identification and verification using iris recognition by calculating hamming distance. *International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231*, 2, May 2012.

[16] C. Dwork. Differential privacy. In *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*, pages 1–12, 2006.

[17] C. Dwork, F. McSherry, K. Nissim, and A. D. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, pages 265–284, 2006.

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

[19] C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Proceedings of CRYPTO 2013*, volume 8042, pages 75–92. Springer Berlin Heidelberg, 2013.

[20] Y. Huang, L. Malka, D. Evans, and J. Katz. Efficient privacy-preserving biometric identification. In *Proceedings of NDSS*, 2011.

[21] D. J. Hurley, M. S. Nixon, and J. N. Carter. Force field feature extraction for ear biometrics. *Computer Vision and Image Understanding*, 98(3):491 – 512, 2005.

[22] T. Ignatenko and F. M. J. Willems. Information leakage in fuzzy commitment schemes. *IEEE Transactions on Information Forensics and Security*, 5(2):337–348, 2010.

[23] A. Jain, K. Nandakumar, and A. Nagar. Biometric template security. *EURASIP Journal Advances Signal Proceedings*, pages 1–17, 2008.

[24] A. K. Jain, A. Ross, and S. Prabhakar. An introduction to biometric recognition. *IEEE Trans. Circuits Syst. Video Technol.*, 14:4–20, 2004.

[25] 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, 1999.

[26] S. Kanade, D. Petrovska-Delacrtaz, and B. Dorizzi. Cancelable iris biometrics and using error correcting codes to reduce variability in biometric data. *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, pages 120–127, June 2009.

[27] K. Kommel and C. Vielhauer. Reverse-engineer methods on a biometric hash algorithm for dynamic handwriting. *Proceedings of the 12th ACM workshop on Multimedia and security*, pages 67–72, 2010.

[28] A. Kong, K. H. Cheung, D. Zhang, M. Kamel, and J. You. An analysis of biohashing and its variants. *Pattern Recogn.*, 39:1359–1368, July 2006.

[29] E. Pagnin, C. Dimitrakakis, A. Abidin, and A. Mitrokotsa. On the leakage of information in biometric authentication. In *Proceedings of Indocrypt 2014*, LNCS, pages 265–280. Springer, 2014.

[30] L. Penrose. Dermatoglyphic topology. *Nature*, 205:544–546, February 1965.

[31] M. O. Rabin. How to exchange secrets with oblivious transfer. *IACR Cryptology ePrint Archive*, 2005:187, 2005.

[32] C. Rathgeb, F. Breitinger, C. Busch, and H. Baier. On application of bloom filters to iris biometrics. *IET Biometrics*, 3(4):207–218, 2014.

[33] C. Rathgeb and A. Uhl. Iris-biometric hash generation for biometric database indexing. *Pattern Recognition, International Conference on*, pages 2848–2851, 2010.

[34] K. Simoens, J. Bringer, H. Chabanne, and S. Seys. A framework for analyzing template security and privacy in biometric authentication systems. *IEEE Transactions on Information Forensics and Security*, 7(2):833–841, 2012.

[35] A. C.-C. Yao. How to generate and exchange secrets. In *Foundations of Computer Science, 1986., 27th Annual Symposium on*, pages 162–167. IEEE, 1986.

[36] 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 Inf.*, volume 8128 of *LNCS*, pages 55–74, 2013.

[37] N. Zaeri. Minutiae-based fingerprint extraction and recognition. In *Computer and Information Science - Artificial Intelligence - "Biometrics"*, June 2011.

[38] P.-F. Zhang, D.-S. Li, and Q. Wang. A novel iris recognition method based on feature fusion. *In Proc. Int. Conf. on Machine Learning and Cybernetics.*