ELITE: zEro Links Identity managemenT systEm

Tarik Moataz[1,2], Nora Cuppens-Boulahia[2], Frédéric Cuppens[2], Indrajit Ray[1], and Indrakshi Ray[1]

[1] Dept. of Computer Science, Colorado State University, Fort Collins, CO 80523
{tmoataz, indrajit, iray}@cs.colostate.edu
[2] Institut Mines-Télécom, Télécom Bretagne, Cesson Sévigné, France
{nora.cuppens, frederic.cuppens}@telecom-bretagne.eu

Résumé Modern day biometric systems, such as those used by governments to issue biometric-based identity cards, maintain a deterministic link between the identity of the user and her biometric information. However, such a link brings in serious privacy concerns for the individual. Sensitive information about the individual can be retrieved from the database by using her biometric information. Individuals, for reasons of privacy therefore, may not want such a link to be maintained. Deleting the link, on the other hand, is not feasible because the information is used for purposes of identification or issuing of identity cards. In this work, we address this dilemma by hiding the biometrics information, and keeping the association between biometric information and identity probabilistic. We extend traditional Bloom filters to store actual information and propose the SOBER data structure for this purpose. Simultaneously, we address the challenge of verifying an individual under the multitude of traits assumption, so as to guarantee that impersonation is always detected. We discuss real-world impersonation use cases, analyze the privacy limits, and compare our scheme to existing solutions.

1 Introduction

Increasingly, governments are moving towards using biometric based systems for national identity cards for their citizens. Examples of these are the proposed project of Carte nationale d'identité biométrique of the French government (see http://www.service-public.fr/actualites/002101.html), the AADHAAR project undertaken by the Unique Identification Authority of India (see http://uidai.gov.in) and many others countries. These governments have started creating large biometric database systems to issue social security cards, health insurance cards etc. The main objective is to efficiently provide citizen services via accurate identity verification. For this purpose, these systems typically maintain a database of sensitive personal information of the individuals, called the *identity database*, a *biometric database* containing the biometric information of individuals, and a *deterministic link* between the identity of the individual in the identity database and her biometric information in the biometric database. This link is important because it helps to ensure that any attempted impersonation is detected. Such impersonation occurs when an attacker tries to manipulate the system to obtain another identity illegally. In existing biometric information management systems, this link is public and not protected.

Impersonation can happen at many stages of the biometric card establishment and operation phases. We are interested in two specific situations, the so called "First application for biometric card" scenario and the "Renewal without a document/ ID card loss" scenario. These capture the main operations in the biometric system that we focus on in this article.

First application for biometric card : During the first application, the applicant goes through a verification step in order to determine if he is a legitimate applicant. If the biometric information does not exist in the biometric database, then this application is genuine. It is a case of *impersonation* when the individual provides an identity that already exists in the identity database.

Renewal without a document : During the renewal phase, the applicant has to go through the verification step in order to determine if he already exists in the system. Renewal can take place if and only if the relevant biometric information already exists in the biometric database. It is a case of impersonation when the individual gives a different identity that is not associated with his biometric information.

The deterministic link between biometric information and identity help detect such impersonations. However, it brings forth serious privacy concerns for the individuals. Biometric information is, surprisingly, easily available without consent. For example, fingerprints can be easily picked up from different surfaces. An attacker with access to some biometric information can then easily obtain sensitive information from the identity database. Consequently, individuals who are concerned about their privacy, may want these deterministic links between the identity database and the biometric database removed. Unfortunately, since removing the link is not an option, we investigate in this work if a *probabilistic link* between biometric information and identity can maintain user privacy and at the same time preserve all the functionalities of the system.

One possible solution to this problem is to keep the information in the biometric database encrypted, so that the identity is linked to encrypted biometric data. If the attacker cannot correlate the illegally obtained biometric information with the encrypted information stored in the database, the attacker will not be able to breach privacy of the individual. Several works aim to provide biometric privacy employing variants of this theme [17,14,11], although they do not address the same problem as ours. In fact, these techniques cannot be used as native constructions to solve our problem. Under these setups, proper identification would work only if an individual could be associated with only one stored biometric information. Unfortunately, owing to the vagaries of biometric capture devices, a physical biometric pattern can have several captured versions – the so called *multitude of traits* issue. These versions are not totally different and have considerable similarities. However, they are rarely exactly the same. An identification based on two similar pieces of biometric information is always possible with different levels of accuracy and efficiency. However, these similarities quickly vanish when the biometric information is encrypted. This is one of the reasons why biometric information is traditionally stored as plaintext, although the work of Bringer et al. [5] proposes an error-tolerant searchable encryption scheme that can be used to solve this problem [1] albeit at the expense of very high computational overhead making it impractical for large scale deployment.

The Setbase approach proposed by Adi Shamir [18] was the first scheme to address the problem of converting a deterministic association between biometric information and identity into a probabilistic one. This scheme stores the biometric information as plaintext. The relation *one to one* is replaced by a relation n *to* n where n is the size of a subset of identities set. The idea centers around fixing a number m of subsets without fixing their size. This results in m subsets of identities, and their associated subsets of biometrics. Consequently, each identity is associated to many (the size of the subset) biometrics and, vice versa, each biometric information is associated with many identities. This concept makes the biometric-identity association private. The approach allows one to detect impersonation with a certain probability but not as accurately as a deterministic *one* to *one* mapping. There have quite a few works that explored the underlying principles of the Setbase approach and quantify various parameters [13,12]. However, the Setbase approach has several issues that prevent it from being adopted in practice. First of all, the association between a given subset of identities and the corresponding subset of biometric information represents a valuable information and should be kept secret. If there is an attacker (including insiders) who knows that a given person is in a subset S_i, it will enable the attacker to usurp this identity if the attacker and the individual share the same subset. For this reason, the association is encrypted using a probabilistic asymmetric semantically secure scheme that hides the association. Moreover, keys should be kept secret in order to protect the system while biometric information is stored as plaintext without any transformation, encryption or obfuscation. Finally, any deletion is impossible in the Setbase approach which makes the scheme not really scalable.

In this paper, we present a novel scheme called ELITE (acronym for zEro Links Identity managemenT systEm) to allow a probabilistic link between biometric information and identity. We describe an initial construction called ELITE-1 that introduces the main principles of the scheme. We then refine this to propose ELITE-2 – the second and main construction. ELITE-1 assumes that the biometric information stored in the database and the biometric information retrieved from the sensor during the verification are the same. We propose a novel probabilistic data structure called "Stored Object Bloom Filter" (SOBER, for short) for storing identities, which is based on the traditional Bloom Filters. We then adapt the Greedy algorithm proposed by Azar et al. [2] in the context of the *balls in bins* problem, to insert identities in the SOBER structure. We show how to determine if an individual is in the system during one of the two phases discussed earlier. For the second construction, we take into account the issue of *multitude of traits*. In ELITE-2, the storage and the creation of biometric templates is based on the scheme of R. Capelli et al. [7]. ELITE-2 ensures that even if the captured biometric information is considered different, we are able to store them in a secret way and at the same time preserve the ELITE-1 functionalities. Our proposed scheme has many advantages as compared to the Setbase approach [18] namely, (i) creating a probabilistic link between the biometrics and the identities while maintaining a high impersonation detection rate, (ii) better control over the privacy and impersonation detection dilemma, (iii) biometric information not stored in a plaintext, yet the multitude of traits issue addressed, (iv) more efficient scheme with a constant search complexity and reduced storage space while deletion can still be performed.

2 ELITE : zEro Links Identity managemenT systEm

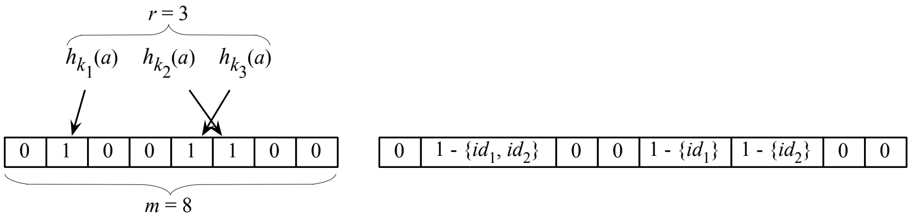
In realworld biometric system, there is always a phase during which the system enrolls new individuals and issues their biometric ID cards or passports. During this phase, a deterministic link between the identity and the biometric information needs to be maintained. We assume that the enrollment phase is not compromised and that this link is deleted directly after the issuance. The ELITE scheme deals with issues related to storing the biometric information in the database after the biometric ID card has been issued – the *storage phase* – and verifying whether an individual is in the system or not – the *search phase*.

We extend the classical Bloom filter data structure [3] to develop the ELITE solution. We call this data structure "Stored Object Bloom filtER (SOBER)". It enables the separation of the identity from the biometric information by making the link between biometric information and identity probabilistic. The scheme also hides the biometric information so that it is impossible to recover stored information. ELITE-1 employs a multiple choice identity allocation algorithm, **Greedy** [2] that has been proposed in the context of the *balls in bins* problem. It allows the insertion of a number of identities in the SOBER data structure. In the following, we first introduce the Stored Object Bloom Filter data structure, present the Greedy algorithm, and then discuss the construction of the ELITE-1 scheme. We present an example to discuss how our system works and then analyze the privacy features of the scheme.

2.1 Preliminaries

Stored Object Bloom Filter - SOBER Bloom filters [3] are probabilistic data structures that permit testing membership of an element in a group. Bloom filter as a structure does not allow the storage of the element, but only the membership verification of an element. The most important feature of a Bloom filter is that the time complexity for membership verification is constant i.e. $O(1)$. SOBER is a $\langle key - value\rangle$ data structure that has similar construction as a normal Bloom filter with the additional feature that the cells can contain extra information. In the following, we first discuss the construction steps of SOBER, then describe how to search for an element using this probabilistic data structure.

SOBER construction Briefly, a classical Bloom Filter works as follows. Let us consider a set of n elements $S = \{a_1, ..., a_n\}$ and r independent hash functions $h_{k_j} : \{0,1\}^* \rightarrow [\![0, m]\!]$ where m is the size of an array $\mathcal{A}$. We initialize all cells of $\mathcal{A}$ to zero. For each a_i in S and $j \in [\![1, r]\!]$, we compute the value $h_{k_j}(a_i)$. The output of each hash function represents the index to a cell in array $\mathcal{A}$ whose value will be set to 1. This is shown on the left side in Figure 1. To test whether an element a' is a member of the set S, we have to only calculate $h_{k_1}(a'), ..., h_{k_r}(a')$. If for each $j \in [\![1, r]\!], \mathcal{A}(h_{k_j}(a')) = 1$ then we can conclude that $a' \in$ S (with a very degree of certainty), otherwise $a' \notin$ S. Note that, in some membership verification cases, we can have a false positive for elements whose identities were not stored in the Bloom filter. However, the false positive rate can be arbitrarily reduced.



Bloom Filter construction with m = 8 and r = 3

Corresponding Stored Object Bloom Filter with id_1, id_2, and id_3

FIGURE 1: Relation between traditional Bloom Filter and SOBER

The proposed Stored Object Bloom Filter data structure can be seen as a combination of the Bloom filter with Storage (BFS) [4] and the classical Bloom filter. In BFS data structure, each cell contains a set of values depending on the output of the hash functions while in the basic bloom filter a zero or one value is stored (for fast membership detection). Each cell in the SOBER data structure can be considered as a $\langle key, value \rangle$ cell, where $key \in \{0,1\}$ and the value is a set of elements. This is illustrated on the right side in Figure 1.

Greedy algorithm for inserting identities into SOBER The insertion of identities in the SOBER data structures can be mapped to the classical *balls in bins* problem. In literature, there are many techniques that enable this kind of insertion, such as, the uniform insertion with a single choice insertion and multiple choice insertion. These schemes were constructed for different purposes but their main objective is to decrease the maximum load for every bin, that is the number of balls in any given bin. We are particularly interested in this article to the second type of insertion, namely, the multiple choice insertion. Moreover, we are interested in the special setting where the number of balls is larger than the number of bins but with a constant ratio. The selection of a multiple choice rather than the single choice scheme is mainly based on certain privacy issues that we are going to explain later on. We use the "Greedy" algorithm proposed in [2] for our purpose. Let l denote the number of balls and m the number of bins. **Greedy**$(\mathcal{U}, d)$ is the balls in bins insertion algorithm that places the next ball into the less loaded bin among d bins sampled uniformly at random from $\mathcal{U}$.

2.2 ELITE-1 scheme construction

Storage phase Let us consider the set of n biometric information $\mathbf{b} = \{b_1, \dots, b_n\}$ and the corresponding identity set $\mathbf{id} = \{id_1, \dots, id_n\}$.

1. Create an empty SOBER data structure (all cells initialized with value 0) with size equal to m and r independent hash functions $h_{k_i} : \{0,1\}^* \rightarrow [\![0, m]\!]$.
2. For $1 \leq i \leq n$, hash the biometric information b_i in the set by applying the r hash functions and store a 1 in each location of SOBER corresponding to the output of the hash function. If the cell location is already a 1, leave it as 1.
3. Insert one identity value id_i corresponding to b_i uniformly at random in one of the r cell locations identified by outputs of the hash functions when applied to b_i.
4. For $1 \leq i \leq n$, insert l times the same identity id_i associated with b_i in l positions in SOBER following the **Greedy**$(\mathcal{U}, d)$, regardless of whether the cell position is 0 or 1.
5. Create a look-up table $\mathcal{L}$ of n rows. Each row contains the identity id_i and the indexes of corresponding cells in SOBER where id_i has been inserted and the application of the hash functions on the corresponding biometric information b_i, resulted in a 1 in the cell. (Note that at the end of the previous step there can be some cells with a 0 but still containing id_i. Those cells will not be included in a row of $\mathcal{L}$.)

The search phase begins by taking a biometric information b_j of some individual and hashing it r number of times. If each of those r cells in SOBER indexed by the outputs of the hash functions contains a 1, then there is a match and we proceed with the verification of the identity.

Search phase Given the biometric information b_i and the identity id_i :

1. Create an empty set I.
2. Apply the r hash functions $h_i(b_i)$ for $1 \leq i \leq r$, if $h_i(b_i) = 1$ insert the associated identities in I.
3. If $id_i \in I$, then the individual is in the system.

2.3 Discussion

Let us consider a set of biometrics $\{b_1, b_2, b_3, b_4, b_5\}$ with corresponding identities $\{id_1, id_2, id_3, id_4, id_5\}$, a SOBER with size equal to 12, and 3 independent hash functions. Assume that l is equal to 2. A possible construction is shown in Tables 1(a) and 1(b).

Identity	Address in SOBER
id_1	9
id_2	2
id_3	2
id_4	12
id_5	7,12

(a) Look-up table of identities

0-$\{id_3\}$	1 - $\{id_3, id_2\}$	0-$\{id_3\}$	1-$\{id_1\}$
0	1-$\{id_5, id_2\}$	1 - $\{id_5, id_4\}$	0-$\{id_2\}$
1 - $\{id_1\}$	0-$\{id_4\}$	0-$\{id_1\}$	1 - $\{id_4\}$

(b) SOBER with $r = 3$ and $m = 12$. The cells are numbered 1 through 12 left to right from top to bottom.

Table 1: Possible construction of ELITE-1 system

The identity id_5 appears twice in the look-up table since tow of its identities replicas have been added in the 1's positions of the hash functions. On the other hand, some identities can coincide in the same positions which explain that id_5 appear just two times instead of 3 times (l+1) in the SOBER data structure.

Observations In our approach, the system does not store the plain text of the biometric information of registered users. The information stored in SOBER will be used to verify the existence of the user's biometric information. Further, the *look-up table of identities* acts as a proof of whether the user is registered in the system without leaking any information about his biometric information. This look-up table also allows easy deletion of an identity from the information base. Since ELITE-1 knows the positions of the cells in SOBER that contain the identity, we can delete it without altering the other identities or any information in the data structure. Finally, even if the system has knowledge of the type of hash functions, it is very difficult to restore the real biometric information. This is because even using brute force many biometrics can give the same result (the result of a hash function is equal to 1 in the same identity position). We will see in the privacy and computational analysis that the search is constant in time owing to the Bloom feature of SOBER, and storage complexity is far below the Setbase approach. This makes real world deployment practical.

Fraud cases discussions Based on the example, we now discuss the fraud use cases presented earlier in Section 1. We address the first application use case first. Suppose that there is an applicant who comes to an agency for the first application. The first phase to perform is the verification of his biometric information. This verification consists of a search step on the SOBER biometric base. Referring to the SOBER base given in Tables 1a and 1b, let b be the new biometric information of this applicant id. We calculate the output of the three hash functions $h_{K_i}(b_{new})$. We can have the following results :

- If $\exists i \in [\![1, r]\!]$ such that $h_i(b) = 0$ and $id \notin \mathcal{L}$ then neither the biometric information nor the identity exist in the system. In this case, the individual is truly a new user.
- If $h_1(b) = 4, h_2(b) = 2, h_3(b) = 6$ then the biometric information exists in SOBER. If $id \in I = \{id_1, id_2, id_3, id_5\}$ such that the given identity of the applicant was on the identity base, the system then assumes that the applicant has made an error to be addressed for a first application service.
- The biometric information exists in SOBER but the identity does not exist in the look-up table $\mathcal{L}$. This is an attempt of identity theft or impersonation.

For the second use case (renewal without document), the applicant wants to renew his ID card when the biometric information already exists in the system. Suppose that $h_1(b) = 4, h_2(b) = 2, h_3(b) = 6$, $id \notin I = \{id_1, id_2, id_3, id_5\}$ and $id \in \mathcal{L}$. In this case, the applicant is not the person that he claims to be. So this is an impersonation case. In fact, the system proves this by showing that the biometric information provided by the individual exists in the SOBER, and the identity is not in the set I. Note that for this specific example, it is easy for the fraudster to usurp the system since we deal with a small set of identities. However this task is going to be more difficult in real world deployment since this set will be much larger. The size of the identity set I has to be parametrized by the administrator so that the privacy and the fraud detection can follow the administrator's expectation of the system.

2.4 Privacy analysis

The construction phase reveals that the size of the identity set I is crucial for privacy and for reliable impersonation detection. In the following section, we present an analysis that aims to determine the appropriate values that will allow us to create a reliable system. The analysis is dependent on four variables : m the size of the SOBER structure (the number of cells, that is), k the number of hash functions, n the number of identities and l the number of random insertions into SOBER for each identity. We first define the degree of privacy and the probability of fraud detection.

Definition 1. *Let I be the set of retrieved identities during the search phase. The degree of privacy p_P for any individual is a ratio equal to : $p_P = \frac{1}{|I|}$, where $|I|$ denotes the size of the set I.*

Definition 2. *Let I be the set of retrieved identities during the search phase. The probability of fraud p_F for the two use cases defined in Section 1 are as follows :*

$$p_F = \begin{cases} \frac{1}{|I|} & \text{use case -- first application for biometric card} \\ \\ \frac{|I|}{n} & \text{use case -- renewal without a document} \end{cases}$$

The above definitions capture the fact that the size of identity set I controls the rate of fraud and at the same time the degree of privacy of individuals. In fact, the probability to detect fraud in our first use case is equal to $\frac{1}{|I|}$, where I is the number of unique identities in the identity set I. On the other hand, for our second use case, the probability p_F that a fraudster gives a different identity in the same identity set I is equal to $\frac{|I|}{n}$. Moreover, if the size of the identity set increases the privacy level of users also increases. Thus, a small privacy degree p_P reflects a high privacy level.

It is clear that there is a trade off between the privacy level that the system offers, the reliability of fraud detection and the difficulty to mislead the system. Indeed, if the set of unique identities gets larger, the fraudster gets a higher chance of cheating the system ; on the other hand, if the set of unique identities gets larger, users get better privacy. The degree of privacy is based on how evenly we distribute the identities over all positions of the SOBER data structure. The best scenario will be a case where every position stores exactly $\frac{l \cdot n}{m}$ identities. However, a random insertion of identities cannot guarantee this result. Thus, we have used the Greedy algorithm to decrease the maximum load of every position in order to be as close to the ideal situation as we can. Moreover, decreasing the maximum load will increase the minimum load (i.e. the minimum number of identities in any cell). One may be led to believe that a deterministic identity insertion will be better in our scenario. However, it is not the case from a security perspective. This is because having a deterministic insertion algorithm will leak information about the strategy of identity insertion. Consequently, for any internal adversary the task of identity deletion will be straightforward.

Essentially, we want to have a SOBER data structure where empty positions are very rare, almost non-existent. Empty positions refer to those positions in the SOBER data structure that do not contain any identity. We employ a classical problem known as the *the occupancy problem* [8], that gives us the exact probability of finding an empty position. We will show that, using the Greedy algorithm, we can control the minimum and maximum load of every cell in SOBER datastructure and consequently disperse the identities in a uniform manner throughout the entire data structure.

Decreasing the false positive rate in SOBER SOBER is a probabilistic data structure that involves some false positives. Let p_f denote the probability of a false positive. We first determine the appropriate values to minimize the false positive rate. Let us consider a SOBER with a size equal to m associated with k hash functions. We have n entries. Each insertion in the SOBER will imply insertion of $l+1$ same identity values uniformly at random. We consider hash functions as random functions. We can show that for $k = \frac{m}{n}\ln(2)$, the probability of false positives is the minimum and is equal to $p_f = 2^{-k}$.

Probability of an empty cell in the SOBER data structure Let us assume that insertion of identities are made uniformly at random with a single choice, i.e. every identity has one random choice to get into a given cell. We have $l \cdot n$ identities and m cells. We denote by $X_{n.l,m}$ the number of empty cells after all insertions. We can show that the probability that all cells contain at least one identity is equal to : $\Pr(X_{n.l,m} = 0) = \sum_{i=0}^{n}(-1)^i\binom{m}{i}(1 - \frac{i}{m})^{n.l}$. This formula can be approximated [9] to : $\Pr(X_{n.l,m} = 0) \simeq e^{-\lambda}$, where $\lambda = m.e^{-\frac{n.l}{m}}$.

Minimum/Maximum Load of any cell The Greedy algorithm ensures with a high probability [20] a maximum load equal to $\frac{n \cdot l}{m} + \sqrt{\frac{n \cdot l \cdot ln(m)}{m}}$ in the case where $n \cdot l > m \cdot ln(m)$. While the maximum load defines the upper bound of the number of identities by cell, the minimum load is very important as well, since it controls the minimum size of the set I in the worst case. The following theorem gives the behavior of the minimum load of the *Greedy*($\mathcal{U}, 2$) algorithm.

Theorem 1. *Let $n \cdot l$ be the number of identities, m the size of the SOBER data structure and $d = 2$ the parameter of the Greedy algorithm. Let p be a positive real number. Then, we have with a probability at least equal to $1 - \frac{n \cdot k}{ln(2)} \cdot e^{-\frac{l \cdot ln(2) \cdot p^2}{2k}}$ the minimum load of any bin to be larger or equal to : $\frac{(1-p) \cdot l \cdot ln(2)}{k}$*

A direct consequence of Theorem 1 is that the number of identities in the worst case with a probability equal to $1 - p(l, k, n, p)$ is equal :

$$|I| = (1 - p) \cdot l \cdot ln(2)$$

where $p(l, k, n, p) = \frac{n \cdot k}{ln(2)} \cdot e^{-\frac{l \cdot ln(2) \cdot p^2}{2k}}$. We can control the minimum load by choosing a proper value of l for a fixed number of hash functions as well as a fixed population. This implies that the administrator can control the privacy of fraud p_F as well as the degree of privacy p_P. We should emphasize that the bigger the set I the more private the individual's biometric is but with lesser fraud detection ratio. This latter ratio should be carefully selected by authorities for a fair use of the system. We refer the reader to the full version of the paper for all the proofs of the previous theorems as well as a discussion on how the number l can be chosen.

3 ELITE-2 solution for multitude of traits issue

The ELITE-1 scheme assumes that the user is associated with only one biometric information that has an exact match during the verification phase. This, however, is not true in real life [10]. In fact, we should differentiate between the biometric information as a physical characteristic of the individual, and the numerical biometric information after being captured by an image sensor. Note that, a physical biometric information can also have several versions. However, these versions are not totally different and have some similarities. So an identification of two similar biometrics can always be possible ; only the accuracy and the efficiency are the main issues. This identification is mainly based on how the biometric information is digitized, and how robustly a biometric information can be represented such that similar biometrics will match even if they are distorted.

In literature, there are several biometric indexing techniques that can be variously classified depending on the features used [18]. Examples are global features such as the average of ridge-line frequency over the whole biometric information, local ridge-line orientations, minutiae and other features obtained from the biometric pattern. In the following, we are interested in the minutiae indexing technique presented in [6,7]. This technique introduces a biometric information indexing based on Minutiae Cylinder-Code, MCC. We provide in the following the details of the MCC approach.

3.1 Minutiae Cylinder-Code overview

The MCC representation is a fixed-radius approach relying on minutiae features of the biometric information. MCC involves three dimensional representations of minutiae into cylinders. Each physical biometric information β can be seen as a set of minutiae that represents a template $\mathcal{T}$ of β such that $\mathcal{T} = \{m_1, \cdots, m_n\}$. Each minutia m_i is defined by its location (x_{m_i}, y_{m_i}) and its orientation in the space θ_{m_i}. The MCC transformation associates each minutia with a local space (cylinder) that encodes spacial and directional relationship with the neighboring minutiae. Each cylinder is divided into multiple cells and each cell contains a value depending on the neighboring minutiae. We will not go into the details of MCC. We describe next the verification steps done using the locality-sensitive hash functions [15].

We represent each biometric information as a set of binary vectors B. Each binary vector b_m corresponds to a MCC transformation of a given minutia m in the template of the biometric information $\mathcal{T}$ such that, $B = \{\mathbf{b_m} \mid m \in \mathcal{T}\ and\ MCC(m) = \mathbf{b_m}\}$.

For two biometrics β_1 and β_2 having respectively the templates $\mathcal{T}_1$ and $\mathcal{T}_2$, we generate the binary vector sets for both templates B_1 and B_2. A similarity measure between these two biometric information can be done using Hamming distance [19] such that, $hds(\mathcal{T}_1, \mathcal{T}_2) = \frac{\sum_{\mathbf{b}\in\mathbf{B_2}} max_{\mathbf{b_j}\in B_1}(1-(\frac{d_H(\mathbf{b},\mathbf{b}_j)}{n})^p)}{|B_1|}$ where n represents the size of each binary vector, p a parameter controlling the shape of the similarity and d_H the Hamming distance, with $hds(\cdot)$ near to 0 means no similarity, and a $hds(\cdot)$ near to one means a maximum of similarity. We have to point out that this similarity measure may not be the best choice for MCC comparison, and there are many other more suited measures discussed in detail in [6]. For the sake of simplicity we have chosen the hamming distance similarity measure.

At this point, we cannot directly integrate the MCC transformation in our ELITE solution, since we cannot apply a hamming computation over hashed values of biometrics templates if we store them in our SOBER. In fact, using MCC representation, we can avoid computing hamming distance and replace it by locality-sensitive hash function (LSH) [7]. LSH can be viewed as projecting a n size vector into h size vector where $h < n$. The idea behind the use of LSH is that similar n size vectors still remain similar by projecting them into h size vectors.

The LSH approach consists of selecting k hash functions f_{H_1} defined by randomly choosing k arrival position subsets H_1, H_2,...,H_k. In order to compute the projection of a vector, we apply the k-hash functions ; the output of each hash function is a binary vector with a size equal to h. Thus, using LSH, the Hamming distance similarity can be estimated [16] such that :

$$hds(\mathcal{T}_1, \mathcal{T}_2) \cong \frac{\sum_{\mathbf{b}\in\mathbf{B_2}} max_{\mathbf{b_j}\in B_1}(C(\mathbf{b},\mathbf{b_j}))^{\frac{p}{h}})}{|B_1|.k^{\frac{p}{h}}} \qquad (1)$$

where $C(\mathbf{b}, \mathbf{b_j}) = \sum_{i=1}^{l} \delta[f_{H_i}(\mathbf{b}) - f_{H_i}(\mathbf{b_j})]$ and δ is a Dirac symbol equal to 1 in case of equality and zero in the other case. We refer the reader to [7] to the experimental results on multiple well known biometrics databases.

Summing up, since we do not require computing Hamming distance for an identification, the MCC representation and the multiple LSH solution can be integrated to the ELITE scheme in order to handle identification under the multitude of traits issue, while at the same time providing a probabilistic link between individuals and their biometric information. In the following we describe the solution ELITE-2.

3.2 ELITE-2 construction

Let us consider a set of n biometrics considered as n templates, where each template is a set of minutiae, and the associated set of n identities $\mathbf{id} = \{id_1, \cdots, id_n\}$. After applying the MCC transformation, the result will be a set of n binary vectors such that $\mathbf{B} = \{B_1, \cdots, B_n\}$. Let us consider s locality-sensitive hash functions (LSH) defined by a random sampling of arrival spaces such that $H_1, \cdots, H_s$, the size of each arrival space being equal to h. We should underline the fact that the size h will determine later the size of the SOBER filters. In the following, we describe the storage phase as well as the search (i.e. verification) phase.

Storage phase First, we create s SOBERs with the same set of r independent hash functions $h_1, \cdots, h_r$, where each SOBER has a size equal to 2^h. Each cell in each SOBER will be divided into two lists. The first list will contain the identifier couples of each minutia transformation – for example $(1, 2)$ denotes the second minutia of the first biometry (instead of containing one or zero value) – and the second list contains the identities. Let us consider the MCC transformation of the i^{th} biometric information $B_i = \{b_{i,1}, \cdots, b_{i,t}\}$ and the associated identity id_i. For each $b_{i,j} \in B_i$, the algorithm proceeds in these steps :

1. For each $1 \leq k \leq s$, apply the r independent hash functions $h_1, \cdots, h_r$ such that $\{SOBER_k[h_1(f_{H_k}(b_{i,j}))] = (i, j), \cdots, SOBER_k[h_r(f_{H_k}(b_{i,j}))] = (i, j)\}$,
2. For each $1 \leq k \leq s$, insert the identity id_i in l cell of each SOBER$_k$ following the Greedy algorithm,

3. Insert only one value of id_i in only one SOBER uniformly at random in the positions where the r hash functions outputted has an outputted result.
4. Create a row in the look-up table $\mathcal{L}$ which contains the identity id_i and the corresponding positions in the random selected SOBER where id_i belongs to cells where hash functions outputted a 1's result for the biometric information B_i.

We reiterate these steps for all binary vectors B_i. At the end we will output s filled out SOBERS which represents the new biometric database.

Search phase The input of this phase is a scanned physical biometric information and an identity id. We want to verify whether the biometric information exists or not in the biometric database. The first step is to transform the scanned biometric information using the MCC representation. The output of the MCC representation is a binary vector set B. In order to perform the verification we follow these steps :

1. Create t empty collusion sets $C_1, \cdots, C_t$ and t empty identity sets $(I_1, \cdots, I_t)$,
2. For $1 \leq k \leq s$, for $1 \leq i \leq t$, compute the value $h_1(f_{H_k}(b_i)), \cdots, h_r(f_{H_k}(b_i))$ and retrieve from $\{SOBER_k[h_1(f_{H_k}(b_{i,j}))], \cdots, SOBER_k[h_r(f_{H_k}(b_{i,j}))]\}$ the couple (or couples) existing in all the corresponding positions as well as all the corresponding identities. Store them respectively in C_i and I_i.
3. For $1 \leq i \leq t$, rearrange the list C_i such that for each couple we give a score that represents the number of images of the corresponding couple in C_i, that is, the number of hash functions that collide between the new entry and existing biometric information(s).
4. Based on $C_1, \cdots, C_t$, select the maximum number of occurrences that belongs to the same template, then calculate the similarity based on the equation 1. If the similarity is bigger than a minimum that the administrator defines, the biometric information exists.
5. If $id \in I = \{I_1, \cdots, I_t\}$, conclude that the identity belongs to the system.

We should point out that the size of I is very important since it represents the parameter of privacy that we have explained in previous section (see Definition 1). In addition, the size of the SOBER is 2^h, which is equal to all the possibilities of hash function space H_i. On the other hand, the number of entries for each SOBER is equal to $n \times t$ where n is the number of biometrics and t the number of minutiae in each biometric information. (In practice $t \cong 70$ [7]). In order to have the minimum false positives and decrease the collision in the same SOBER for different binary vectors b, we should verify the following equation : $r = \frac{2^h}{n.t} ln(2)$.

In addition, tuples stored in the SOBER cells do not disclose any information about the identity. A number of these couples exist in the construction so as to maintain a link between minutiae and not to individuals' identity. This link allows one to determine the number of collusions for each stored minutiae in relation to others in the same template, as shown in the search phase.

From privacy perspective, the analysis can be done following the same steps as ELITE-1. The only main difference is the number of SOBERs (the number of minutiae associated with each biometric information is considered as a single entry).

3.3 Complexity analysis

The storage complexity of ELITE-1 is dependent on the number of instances of identities, which is equal to l for each identity. If n represents the number of identities, the storage complexity is equal to $\mathcal{O}(n \cdot l)$. ELITE-2 represents a solution that can accomodate the multitude of traits issue, which ELITE-1 cannot, while maintaining the advantage of the basic ELITE-1 scheme. ELITE-2 derives its power from the constant search time of SOBER. However since the ELITE-2 construction requires the use of s bloom filters, the search time is equal to $\mathcal{O}(s)$. On the other hand, the use of s Bloom filters increases the storage complexity to $\mathcal{O}(s \cdot n \cdot l)$. (Here we do not take into account the constant factor of number of minutiae, which is in the order of ~ 70 minutiae per biometric information). ELITE-2 takes into consideration the multitude of traits for deletion of biometrics while keeping identities unlinked to their hidden biometrics. Table **??** presents a functional and computational comparison between the Setbase approach and the two ELITE solutions.

ELITE-2 has many privacy advantages compared to the Setbase approach, specially with regards to the flexibility it offers for the choice of the rate of impersonation detection. In the Setbase approach, the rate is equal to m/n, where m is the size of each subset and n the size of whole population. Since the number of subsets are fixed, the m factor increases, which decreases linearly the detection rate. In ELITE-2, the randomization factor can be dynamically changed depending on the authorities' expectations. In ELITE-2, fixing the size of the SOBER is mandatory before inserting elements. This can be a shortcoming to overcome if the distribution of population growth is not pre-determined (which is typically the case). However, even if we made the assumption of an unknown population growth, a good way to proceed is to divide each SOBER into chunks (each chunk is a different SOBER with its own hash functions) that we fill up depending on the population growth.

Scheme	Search complexity	Storage complexity	Multitude of traits	Hidden biometric information	Delete operation
ELITE-1	$\mathcal{O}(1)$	$\mathcal{O}(n.l)$	no	yes	yes
ELITE-2	$\mathcal{O}(s)$	$\mathcal{O}(s.n.l)$	yes	yes	yes
Setbase approach	$\mathcal{O}(n)$	$\mathcal{O}(n\|B\|)$	yes	no	no

Table 2: Comparison between ELITE-(1,2) and Setbase approach

4 Conclusion

One of the biggest concerns of biometric based systems such as the ones used for issuing biometric based identity cards is that the systems include a deterministic link between the biometric information of an individual and her identity. Since the system also contains sensitive private information, such deterministic links can cause identity thefts for an individual when an attacker misuses a externally obtained biometric information to impersonate a registered user. In this work, we presented two constructions, ELITE-1 and ELITE-2, that render the association between the biometric information and the identity probabilistic. ELITE-2 improves upon ELITE-1 to address the challenges posed by the multitude of traits issue. We provide a theoretical analysis of the privacy guarantees of the ELITE scheme. We discuss how real-world impersonations can be detected. Finally, we provide analytical results of the storage and search complexities of the two schemes.

Future work involves a thorough new stateful algorithm that takes dynamically into consideration the distribution of identities during the storage phase in order to have more precise control over the probabilistic parameters. In addition, we plan to investigate how our scheme ELITE-1 can be applied to other applications, such as, keeping the relationship between an individual and his genetic information secret.

Acknowledgement

Références

1. M. Adjedj, J. Bringer, H. Chabanne, and B. Kindarji. Biometric Identification over Encrypted Data Made Feasible. In *Proceedings of the 5th International Conference on Information Systems Security*, volume 5905 of *Lecture Notes in Computer Science*, pages 86–100, Kolkata, India, December 2009.
2. Y. Azar, A. Z. Broder, A. R. Karlin, and E. Upfal. Balanced allocations. In *Proceedings of the 26th annual ACM Symposium on Theory of computing*, pages 593–602, Chicago, Illinois, USA, May 1994. ACM.
3. B. H. Bloom. Space/Time Trade-offs in Hash Coding with Allowable Errors. *Communications of the ACM*, 13(7) :422–426, 1970.
4. D. Boneh, E. Kushilevitz, R. Ostrovsky, and W. E. Skeith III. Public Key Encryption that Allows PIR Queries. In *Proceedings of the 27th Annual International Cryptology Conference*, volume 4622 of *Lecture Notes in Computer Science*, pages 50–67, Santa Barbara, California, USA, August 2007.
5. J. Bringer, H. Chabanne, and B. Kindarji. Error-Tolerant Searchable Encryption. In *Proceedings of IEEE International Conference on Communications*, pages 1–6, Dresden, Germany, June 2009.
6. R. Cappelli, M. Ferrara, and D. Maltoni. Minutia Cylinder-Code : A New Representation and Matching Technique for Fingerprint Recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 32(12) :2128–2141, 2010.

7. R. Cappelli, M. Ferrara, and D. Maltoni. Fingerprint Indexing Based on Minutia Cylinder-Code. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 33(5) :1051–1057, 2011.
8. W. Feller. *An Introduction to Probability Theory and Its Applications : Volume One*. John Wiley & Sons, 1968.
9. L. Host. Some asymptotic results for occupancy problems. *The Annals of Probability*, 5(6) :1028–1035, 1977.
10. A. K. Jain, R. M. Bolle, and S. Pankanti. *Biometrics : Personal Identification in Networked Society*. Springer, 1999.
11. A. K. Jain, K. Nandakumar, and A. Nagar. Biometric Template Security. *EURASIP Journal on Advances in Signal Processing*, 2008, 2008.
12. B. Justus, F. Cuppens, N. Cuppens-Boulahia, J. Bringer, H. Chabanne, and O. Cipiere. Define Privacy-preserving Setbase Drawer Size Standard : A ϵ-closeness Perspective. In *Proceedings of the 11th Annual International Conference on Privacy, Security and Trust*, pages 362–365, Tarragona, Catalonia, Spain, July 2013.
13. B. Justus, F. Cuppens, N. Cuppens-Boulahia, J. Bringer, H. Chabanne, and O. Cipiere. Enhance Biometric Database Privacy : Defining Privacy-Preserving Drawer Size Standard for the Setbase. In *Proceedings of the 27th Annual IFIP WG 11.3 Conference, Data and Applications Security and Privacy*, volume 7964 of *Lecture Notes in Computer Science*, pages 274–281, Newark, New Jersey, USA, July 2013.
14. T. A. M. Kevenaar, U. Korte, J. Merkle, M. Niesing, H. Ihmor, C. Busch, and X. Zhou. A Reference Framework for the Privacy Assessment of Keyless Biometric Template Protection Systems. In *Proceedings of the Special Interest Group on Biometrics and Electronic Signatures*, pages 45–56, Darmstadt, Germany, September 2010.
15. E. Kushilevitz, R. Ostrovsky, and Y. Rabani. Efficient Search for Approximate Nearest Neighbor in High Dimensional Spaces. In *Proceedings of the 30th Annual ACM Symposium on the Theory of Computing*, pages 614–623, Dallas, Texas, USA, May 1998.
16. S. Mimaroglu and D. A. Simovici. Approximate Computation of Object Distances by Locality-Sensitive Hashing. In *Proceedings of the 4th International Conference on Data Mining*, pages 714–718, Las Vegas, Nevada, USA, July 2008.
17. G. J. Schmidt, C. Soutar, and G. J. Tomko. Fingerprint Controlled Public Key Cryptographic System. Patent #US5541994 A. Mytec Technologies Inc., July 1996.
18. A. Shamir. Adding Privacy to Biometric Databases : The Setbase Approach. Presentation at the 31st International Conference of Data Protection and Privacy. Available from `http://www.privacyconference2009.org/program/Presentaciones/common/pdfs/adhi_shamir_madrid.pdf`, 2009 (Last accessed September 23, 2013).
19. A. M. Steane. Error Correcting Codes in Quantum Theory. *Physical Review Letters*, 77(5) :793, 1996.
20. K. Talwar and U. Wieder. Balanced allocations : The weighted case. In *Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing*, STOC '07, pages 256–265, New York, NY, USA, 2007. ACM.