Questions de Sécurité et de Vie Privée autour des Protocoles d'Identification de Personnes et d'Objets

Bruno Kindarji

Those who would give up Essential Liberty to purchase a little Temporary Safety, deserve neither Liberty nor Safety.

- - Benjamin Franklin

Foreword

The context.

Identification of people, devices, or patterns is a very broad subject, on which many research efforts are being deployed. By identification, we mean automatic identification by means of algorithms, protocols, making use of sensors and computers. In other words, the identification is a procedure in which the subject who needs to be identified does not directly provide its identity. This definition, intently vague and general, can be applied to many real-life situations. For example, a night guard at a military facility can recognize the high ranking officers who are granted access to the building. However, should there be a high turnover of either the officers or the night guards, then this security measure becomes inefficient. The officer now needs to present an identifying element in order to gain access.

This is the classical step to switch from a situation where the security is based on *who I am* to a situation where what matters is either *what I know* or *what I have.* The advantage of the first choice is that there is no need to carry a burdening equipment to enter; however, (classical) knowledge is transmittable, while a well thought design can make reproducibility of security elements hard or impossible.

About Biometrics and Privacy

The first research axis of this manuscript is to revert to what is commonly called the third factor (*what we are*) even though we saw that this is actually the *first* factor.

The global research effort on automatic identification is justified by the need to make a safer use of a whole range of technologies that were invented these last decades. From physical threats against strategic targets, to privacy risks when using a specific virtual service, the range of real-life menaces to take into account is wide. While the protection against physical violence is mostly oriented towards the verification of everyone's identity, it also requires everyone to disclose as much information as possible. However, more disclosure also means less privacy.

Privacy is the ability of an individual to seclude information about himself from the incursions of the others [188]. Defining this property is not an easy task for scientists; however some attempts were made in this direction. Once this was done, achieving privacy, while ensuring the maximal security and the minimal annoyance for the user, is a major scientific challenge.

The solution that we naturally considered for this purpose is the use of biometrics as a means of identification. Biometrics are fascinating elements of identification embedded in the human body; using them properly can lead to impressive levels of precision and accuracy. However, adding biometrics to the privacy equation raises even more challenges, as these identifying elements lead to identity theft, or more simply, disclosure of someone's habits. To prevent these risks, few solutions existed at the beginning of this thesis. A large part of this work consists in studying the implications of using biometrics in security protocol. In particular, the storage and transmission of biometric templates must be handled with specific precautions.

Existing Solutions for Template Protection We consider in this work that it is possible for the transmission of biometric templates to be made as secure as any transmission, as long as the adequate cryptographic infrastructure is chosen. The first part of this document studies the state-of-the-art solutions for template storage, which is the Achilles' heel of biometric systems. In particular, solutions such as Secure Sketches [27, 28] and Cancelable Biometrics [31, 32] are considered, in order to evaluate how good the associated algorithms behave on real-life data, and also how strong the underlying security is. We explicitly exhibit limitations that are inherent to these methods.

Propositions for Secure Biometric Identification In the second part of this document, we propose new methods for using biometrics in a secure setting, and more specifically, in the context of biometric identification. We propose a new cryptographic primitive, called *Error-Tolerant Searchable Encryption* [33], which is a generalization of Searchable Encryption. The application of Error-Tolerant Searchable Encryption to biometric data can lead to protocols that enable (secure-) biometric identification [34].

This aforementioned primitive was designed in the spirit of public-key cryptography. However, in order to preserve as much privacy as possible, the cryptographic requirements that were raised led to expensive computation on the database. In particular, the cryptographic operations that need to be done must be linear in the number of enrolled people; if this number is about the size of a national population, the computations are already too costly to be practical. Another line of work to solve this issue consists in applying symmetric cryptography, and different data structures for the storage [2]. The security properties that can be derived are different, and somewhat less protective. Nevertheless, the efficiency of this scheme is much greater, which makes that scheme interesting to consider.

Finally, we show how it is possible to investigate different strategies for biometrics by changing models. One example is to apply time-dependent functions to biometric templates [32]; if the function family is well chosen, then this can lead to an interesting application that we call *Anonymous Identification*. Another example is to dedicate biometric hardware to improve the accuracy of a biometric system; with a specific Match-on-Card technology, we are able to deploy a secure identification scheme that requires limited cryptographic requirements, and proves to be very efficient [30].

Solutions for Wireless Communication and Device Identification

The second research axis that motivated our study was the enforcement of privacy while being in a secure setting. Modern cryptography's characteristic is to provide computing primitives that enable people to transmit information in a secure way, and this can be both for military and civilian purposes. This thesis makes explicit several situations where security and privacy are desirable, and sometimes achievable. The link with identification will be made clear at that time.

Minimal-cost Identification Protocols Setting aside the noisy character of biometrics, one can ask the question of the overall complexity of identification protocols. In the case of noisy data, the overall performance can be expressed by the amount of data that did not perform well. With exact data, we can focus on the communication cost of identification protocols. We studied the costs of identification protocols [48], and most especially the League Problem: how many bits must me transmitted from one partner to the other, when there is prior – but non-shared – information on the data.

We then showed that it is possible to outperform the optimal solutions if we allow a small error-probability. This involves several techniques, such as deploying Identification Codes, a barely known though interesting coding primitive.

Private Interrogation of Devices Finally, identification codes can be used in a very different way, in order to beckon an element from a set of low-cost wireless devices. Indeed, there are situations in real-life where the question is not "who is in front of me?", but rather "Is Alice somewhere in the neighbourhood?". For that purpose, we deploy some identification codes, and we also state the cryptographic conditions for the privacy of these elements[35]. Here, the security rests on the unique ability of a wireless sensor to know the identity of the element interrogated; we show that a construction using coding theory over finite fields can be simple yet efficient.

In order to evaluate the security of that scheme, we use a common computational assumption, known as the Polynomial Reconstruction Problem. In order to test further the strength of this assumption in this context, we go further and look for the decoding possibility of Maximum-Distance Separable codes – such as Reed-Solomon codes. That lead us to the study of the threshold of these codes, closely related to their list-decoding capacity. After showing that the behaviour of q-ary codes around their threshold is the same as binary codes, we show how to explicitly estimate the threshold, asymptotically and for codes of finite yet reasonable length.

Overview

This work investigates several means for achieving identification. Authentication is usually preferred to identification because the paradigm is easier to accomplish; however, we discovered during the course of research that there can be efficient ways to deal with the difficulties of identification.

The main issue that is still to be improved is the error rate that is likely to arise as soon as probabilistic methods are used, be it for biometric applications, or for exact data that use probabilistic algorithms. However, while in the first case there is a natural limit that cannot be overcome without working on sensors and signal processing, there are also ways to improve significantly the results that would be obtained for identification in a naive way, and this is true for accuracy, computing-, and communicating costs.

Remerciements

> Obi-Wan is a great mentor, as wise as Master Yoda and as powerful as Master Windu.
>
> Anackin Skywalker

In French and in English, here are a few words, yet not enough to express my gratitude.

Je tiens tout d'abord à remercier Gérard Cohen, qui a dirigé mes recherches, pour son accompagnement et son soutien tout le long de ce doctorat. Je garderai un très agréable souvenir de mon passage à Télécom ParisTech, où les discussions scientifiques étaient parfois relayées par des considérations humoristiques, linguistiques ou culturelles. Toutes ces interactions m'ont bien rassuré sur le fait qu'on peut être scientifique sans pour autant perdre contact avec le reste du monde, bien au contraire! Un grand merci également à Gilles Zémor, également directeur de thèse, qui a toujours été d'excellent conseil. C'est Gilles qui m'a lancé dans l'aventure biométrique en mars 2006, en me mettant en relations avec Sagem Défense Sécurité. Un chapitre de ce document traite de comment transférer une information entre deux joueurs en minimisant le coût de la communication, et je dois dire que Gilles excelle dans ce domaine! Chacune de nos interactions m'a donné plus de recul sur mon travail, et m'a indiqué bien des directions. Gérard et Gilles m'ont apporté chacun une vision différente de la recherche théorique du domaine, visions complémentaires (mais non disjointes), d'une richesse scientifique considérable.

Une thèse CIFRE, c'est la rencontre entre les problématiques de l'industrie et les questions du monde académique. J'ai eu la chance d'être au coeur de questions ouvertes sur lesquelles de nombreuses personnes travaillent; et si les mathématiques sont théoriques (ça reste à prouver...), la biométrie, elle, est tout à fait concrète! Je remercie donc toutes les personnes de Sagem Sécurité qui m'ont donné l'opportunité de participer à ces réflexions. Je remercie tout particulièrement Julien Bringer et Hervé Chabanne qui m'ont initié à ces questions délicates ; s'ils n'avaient pas partagé leur expérience, le début de cette thèse aurait été bien plus difficile. Hervé m'a encouragé à présenter mes travaux sur quatre continents, et m'a incité à mettre en valeur les nouveautés abordées, en regardant le verre à moitié plein plutôt que la bouteille à moitié vide. Ca a par ailleurs été un plaisir de travailler quotidiennement avec Julien sur des problématiques très concrètes, et le point de vue d'un mathématicien, normalien de surcroit, sur des embûches posées par les données de la vraie vie, méritent le détour !

I want to thank my rapporteurs *Adam Smith and Boris Škorić, who agreed to review my dissertation manuscript and go through the arduous task of reading hundreds of pages of stumbling English. Their perspective greatly helped to improve the quality of this document.*

My thanks go also for the other members of the Graduation Committee, Professors Moti Yung and Jean-Pierre Tillich, who made me the honour of attending the defence of my thesis.

I cannot thank enough all my friends and family that have been so supportive of my work and tight schedule these last years; this goes especially for all those who took interest in my work, and even carefully read parts of my dissertation. I would like to thank more especially Samia, Nicolas, Lara and Meige for their invaluable comments on both substance and form. I must *also acknowledge the amazing work of my long-time friend Jad, who is the designer of many illustrations used in the dissertation and the defence of my thesis.*

Ces dernières années à Sagem Sécurité furent l'occasion de rencontrer une équipe avec lesquels il est agréable de travailler ou juste de passer du temps. Je remercie donc, dans un ordre pseudo-aléatoire, Vincent, Pascal, Mélanie, Hervé, Stéphane, Maël, Audrey, Pascal, Mickaël, Isabelle, Thanh-Ha, Yves, Fabien, Ludovic, Vincent, Julien et Benoît. Je remercie particulièrement Thomas pour lui avoir volé ses vacances en Australie, mais aussi pour avoir pu essayer avec lui toutes les manières de se divertir en réseau dans le RER.

On a different tone, I have learned that the doctorate is a path where you learn more than you find out, and a profession that is more than a job. Choosing your research topics is not as hard as choosing your way! My heart goes to all those who helped me discern when there were forks in my life, and especially Nicolas, Yvain, PL and Meige. Last and not least whatsoever, I am grateful for my family for supporting[1] *me on so many levels until today. This thesis is dedicated to all those I love, and you are at the forefront.*

[1] A good translation for this word would be *soutenir*, however *supporter* is accurate too.

Notations

Except when explicitly stated otherwise, the following notations will be used throughout the document.

Sets and Set Operations

- $\emptyset$ is the empty set;
- $\mathbb{N}$ is the set of natural integers;
- $\mathbb{R}$ is the set of real numbers;
- $\mathbb{F}_q$ is the Galois field of size q;
- For $p, q \in \mathbb{N}$, $p < q$, $[\![p, q]\!]$ is the set of all integers between p and q (inclusive).
- For $a, b \in \mathbb{R}$, $a < b$, $[a, b]$ is the set of all real numbers between a and b. $(a, b]$ and $[a, b)$ are the semi-open interval not containing a (resp. b) and (a, b) is the interval of real numbers strictly between a and b.
- If A is a finite set, $|A|$ is the cardinality of A.
- $A \subset B$ denotes "A is a subset of B". $A \subset A$ always hold.
- A^n is the set $A \times \ldots \times A$ (n times).
- $x \in A^n$ is the vector of coordinates $x^{(1)}, \ldots, x^{(n)}$.

Functions and Composition

- B^A is the set of all functions from A to B;
- $f : A \to B$ denotes $f \in B^A$;
- If $f : A \to B$ and $g : B \to C$, then $g \circ f : A \to C$ is the composite of f and g.
- $\delta_{x,y}$ is the Kronecker symbol, equal to 1 if and only if $x = y$, and to 0 otherwise.

Contents

Long Résumé

Ce mémoire de doctorat traite des problématiques de la vie privée et de la sécurité, appliquées à deux contextes très particuliers : la biométrie d'une part, et les communications sans-fil de l'autre. Ce résumé décrit les étapes-clé nécessaires à la compréhension de la thèse, et fait référence au texte anglais qui suit.

Cette thèse propose des contributions à différents aspects du respect de la vie privée et de la sécurité. Bien que l'aspect de l'identification ait été plus particulièrement approfondi, nous pouvons souligner les nouveautés suivantes:

- Améliorations de l'état de l'art en matière de *Secure Sketches* appliqués aux vecteurs binaires;
- Etude des solutions de type *Cancelable Biometrics*, et protocoles permettant de les exploiter;
- Travaux sur l'identification biométrique sur des bases de données chiffrées;
- Architecture d'identification biométrique locale sécurisée par du matériel dédié;
- Généralisation du problème de la ligue en autorisant une proportion d'erreurs asymptotiquement nulle;
- Protocoles d'interrogation d'éléments communicants sans-fil basés sur les codes d'identification;
- Calcul du seuil de codes séparables au maximum de la distance à faible taux.

Ces différents sujets sont résumés ci-après.

A propos de la Biométrie

Le terme “biométrie” désigne, de manière générale, un ensemble de mesures qu'on peut effectuer sur un être vivant. Si ces mesures sont si populaires, c'est parce qu'elles rendent possible la reconnaissance d'une personne (ou d'un animal) en les comparant avec des mesures de référence. La biométrie permet ainsi l'authentification ou l'identification automatique de personnes.

Cette propriété fait de la biométrie un élément un élément de sécurité très intéressant à déployer. Il fait en effet le lien entre une identité numérique (un nom, un numéro d'identification, un compte électronique) et l'identité *physique* de son propriétaire. Cette démarche s'oppose aux méthodes d'authentification plus classiques, à savoir l'authentification par la connaissance d'un secret, ou par la possession d'un élément. C'est d'ailleurs pour cette raison que la biométrie a percé dans ses applications policières (empreintes directes ou latentes, ADN).

Traits et mesures biométriques

Dans la suite du document, un trait biométrique est d'abord numérisé dans un élément b d'un ensemble de traits possibles $\mathcal{B}$. Deux traits biométriques b, b' sont comparés par un algorithme (dit de *matching*) dédié m, qui renvoie un score de similarité $\mathrm{m}(b, b')$. Plus ce score est important, plus les traits sont similaires. Deux traits biométriques provenant d'un même organe donneront donc des scores élevés, alors que deux traits provenant d'organes différents donneront des scores plus faibles.

Comme exemple de trait biométriques, on peut citer les empreintes digitales (qui sont représentées en un ensemble de points caractéristiques appelées les minuties) et l'iris (qui peut être numérisé notamment en IrisCode). Dans le cas de l'iris la numérisation se fait dans $\{0,1\}^n$ où $n = 2048$; le score de similarité s'obtient en calculant une distance de Hamming entre deux mesures. Par la suite, on supposera souvent qu'une similarité se calcule ainsi – un exemple de numérisation de l'empreinte en vecteur de bits est décrit en II.1.

Le processus de reconnaissance biométrique passe toujours par l'enrôlement, où un trait biométrique de référence est mesuré. Une base de données est ainsi constituée pour un système, où sont enregistrées les identités et données biométriques des utilisateurs. Le contenu de cette base est donc hautement sensible au vu du respect de la vie privée des différents utilisateurs, et il convient de déployer un certain nombre de mécanismes afin de protéger ces informations.

Etat de l'art sur la protection de traits biométriques

Le chapitre III présente certaines des techniques les plus réputées au sein de la communauté biométrique en ce qui concerne la protection de bases de données biométriques. L'accent est mis en particulier sur les approches issues des communautés biométriques (en particulier, les *Cancelable Biometrics*) et celles issues de la communauté du codage et de la cryptographie (*Secure Sketches* et *Fuzzy Extractors*). Ces approches sont résumées dans le tableau suivant.

Méthode	Encodage	Comparaison	Conditions de sécurité et commentaires
Match-on-Card	Stockage sur carte à puce	Algorithme biométrique dédié	Sécurité matérielle. Seule l'authentification est possible.
Cancelable Biometrics	Transformation de l'espace biométrique	Algorithme biométrique dédié	Transformation biométrique à sens unique difficile à obtenir.
Secure Sketches	Binarisation et masquage par mot de code	Décodage	Rétention d'entropie suffisante; la perte d'entropie est considérable. Pas d'application à l'identification.
Fuzzy Vault	Binarisation et ajout de bruit aléatoire	Reconstruction Polynomiale	Difficulté de la reconstruction polynomiale; utilisation unique. Pas d'application à l'identification.
Fuzzy Extractors	Création d'un secret et de données publiques auxiliaires	Comparaison exacte	Peu de schémas efficaces proposés. Perte d'entropie *via* les données publiques.
Comparaison des chiffrés	Chiffrement homomorphe d'un template binaire	Opération sur les chiffrés	Opérations coûteuses. Pas d'application à l'identification. Nécessite une représentation binaire des traits biométriques.

Améliorations de l'Etat de l'Art

Comme signalé précédemment, ce mémoire fait état de nouvelles propositions de schémas utilisant les outils cités dans la partie précédente. Dans ce résumé, nous soulignerons plus particulièrement une construction de Secure Sketches utilisant des codes produits, un protocole qui combine de manière naturelle les Secure Sketches et les Cancelable Biometrics, et enfin une utilisation astucieuse des Cancelable Biometrics qui permet d'éviter le rejeu.

Secure Sketches et Codes Produits

L'approche des Secure Sketches consiste à considérer les différences de capture entre deux traits biométriques comme des erreurs (au sens de la théorie des codes). En déployant une représentation des traits biométriques adaptée, il devient possible d'éliminer les erreurs d'une mesure sur l'autre. Cette technique permet en fait d'utiliser les grandes avancées de la théorie des codes correcteurs d'erreurs pour effectuer une comparaison biométrique. En contrepartie, il devient nécessaire de représenter le trait biométrique sous forme q-aire d'une part, et de rendre publiques des informations liées aux mesures biométriques de l'autre.

La technique "classique" (Juels et Wattenberg, [96]) consiste à représenter un trait biométrique b comme un élément de $\{0,1\}^n$. En choisissant un code correcteur d'erreur $C \subset \{0,1\}^n$, le *sketch* de b est donné par $(c \oplus b, h(c))$ où c est un mot du code C, et h une fonction de hachage. Lorsqu'un nouveau trait biométrique b' est mesuré, la comparaison s'effectue en décodant $(c \oplus b) \oplus b' = c \oplus (b \oplus b')$. Si b et b' sont proches (au sens de la distance de Hamming) alors le résultat du décodage est c; ceci est confirmé par la comparaison du haché avec $h(c)$.

Trouver des codes correcteurs d'erreurs bien adaptés à la biométrie est difficile. Il faut en effet trouver un compromis entre la taille du code (qui doit être grande si on veut pouvoir coder de nombreux éléments sans trop de fausses acceptances) et la distance minimale (qui indique la capacité de correction du code). Par ailleurs, il est important d'avoir accès à un algorithme de décodage efficace pour une application pratique.

Ce mémoire suggère en IV.1 l'utilisation de codes produits comme structure générale pour les Secure Sketches. Etant donnés deux codes correcteurs d'erreurs C_1 et C_2, respectivement $[n_1, k_1, d_1]_q$ et $[n_2, k_2, d_2]_q$, le code produit $C_1 \otimes C_2$ est le code $[n_1 n_2, k_1 k_2, d_1 d_2]_q$ dont les mots de code sont les matrices dont chaque ligne appartient à C_1 et chaque colonne à C_2

$$c = \begin{pmatrix} c_{1,1} & \dots & c_{1,j} & \dots & c_{1,n_1} \\ & & \vdots & & \\ c_{i,1} & \dots & c_{i,j} & \dots & c_{i,n_1} \\ & & \vdots & & \\ c_{n_2,1} & \dots & c_{n_2,j} & \dots & c_{n_2,n_1} \end{pmatrix}$$
$$\forall i \in [\![1, n_2]\!], (c_{i,1}, c_{i,2}, \dots, c_{i,n_1}) \in C_1$$
$$\forall j \in [\![1, n_1]\!], (c_{1,j}, c_{2,j}, \dots, c_{n_2,j}) \in C_2$$

L'utilisation d'un algorithme de décodage "Min-Sum" sur ces codes en permet un décodage itératif efficace qui décode souvent au delà de la distance minimale $d_1 d_2$. En particulier, de tels codes où les codes-lignes et -colonnes sont des codes de Reed et Muller permettent de construire des Secure Sketches efficaces pour l'IrisCode et l'empreinte digitale numérisée.

L'algorithme Min-Sum est un algorithme itératif qui permet de décoder un tel code produit en complexité $O(q^k)$ où $k = \max(k_1, k_2)$; si le code produit est équilibré, on peut ainsi décoder de manière exhaustive toutes les erreurs jusqu'à $\frac{d_1 d_2}{2}$ en complexité $O(q^{\sqrt{k_1 k_2}})$. Cet algorithme est décrit ainsi:

1. Définir pour tous les i, j $\kappa_{ij}^0(x) = 1 - \delta_{x,x_{ij}}$;
2. Itérer la boucle suivante un certain nombre maximal de fois:
   a) Pour $c \in C_1$, calculer le coût pour chaque ligne
   $$\kappa_i^{2\ell}(c) = \sum_{j=1}^{N_1} \kappa_{ij}^{2\ell}(c^{(j)})$$
   b) En déduire $\kappa_{ij}^{2\ell+1}(x) = \min_{c \in C_1, c^{(j)} = x} \kappa_i^{2\ell}(c)$;
   c) Pour $c \in C_2$, calculer le coût pour chaque colonne
   $$\kappa_j^{2\ell+1}(c) = \sum_{i=1}^{N_2} \kappa_{ij}^{2\ell+1}(c^{(i)})$$
   d) En déduire $\kappa_{ij}^{2\ell+2}(x) = \min_{c \in C_2, c^{(i)} = x} \kappa_j^{2\ell+1}(c)$
   e) Si vecteur v^ℓ défini par $v_{ij} = \left(\kappa_{ij}^{2\ell+2}(1) > \kappa_{ij}^{2\ell+2}(0)\right)$ appartient au code, renvoyer v.

Selon les motifs d'erreurs et d'effacement, cet algorithme permet de décoder bien au delà de la distance minimale du code. Il permet ainsi de construire un schéma de Secure Sketch dont les performances en termes de taux de Faux Rejets sont proches de la limite intrinsèque énoncée ci-après :

Théorème. *Soit $k \in \mathbb{N}^*$, et C un code binaire de longueur N et de taille 2^k. Soit m un message issu de la transmission d'un mot de code aléatoire de C, contenant w_n erreurs et w_e effacements.*

Si $\frac{w_n}{N-w_e} > \theta$, alors la probabilité de décoder m correctement est inférieurement bornée par $1 - o(N)$, où θ est le réel tel que la boule de Hamming de rayon $(N - w_e)\theta$ dans $\mathbb{F}_2^{N-w_e}$ contient 2^{N-w_e-k} éléments.

Ce théorème et le corollaire qui en découle (décrits en partie III.5) assurent un taux minimal de faux rejets pour une base de données biométriques fixée, si la comparaison est faite en décodant un mot de code correcteur d'erreurs.

Secure Sketches appliqués à la Cancelable Biometrics

La Cancelable Biometrics inventée dans [145], vise à ajouter de la sécurité au stockage des données biométriques sans renoncer aux algorithmes de matching dédiés. Une famille de fonctions f_i de $\mathcal{B}$ dans $\mathcal{B}$ décrit un schéma de *Cancelable Biometrics* pour l'algorithme de matching m si:

- Il est possible d'appliquer chaque fonction f_i à tous les templates possibles $b \in \mathcal{B}$;
- Si b, b' sont deux templates matchants pour m, alors $f_i(b)$ et $f_i(b')$ sont matchants;
- Inversement, si b et b' ne sont pas matchants, alors $f_i(b)$ et $f_i(b')$ ne sont pas matchants;
- Un template b transformé par f_i et f_j $(i \neq j)$ donnera deux images non matchantes;
- Un template b et son transformé $f_i(b)$ ne sont pas matchants;
- Il est difficile de retrouver par le calcul b à partir d'un de ses transformés $f_i(b)$.

En pratique, réaliser ces six conditions s'avère être une tâche très ardue, et il faut renoncer soit aux bonnes performances, soit au modèle sous-jacent, soit à la sécurité de la construction. Cependant, en supposant que de telles familles existent, nous donnons quelques exemples de protocoles possibles.

Les constructions de type Cancelable Biometrics et de type Secure Sketches sont deux constructions indépendantes, qui agissent selon deux méthodes différentes. En particulier, les Cancelable Biometrics utilisent une classe de fonctions de l'espace biométrique dans lui-même. On peut donc appliquer l'approche des Secure Sketches à l'ensemble des traits biométriques qui ont été modifiés par Cancelable Biometrics.

Suivant ce principe, pour un couple de fonctions $(\mathrm{Sk}, \mathrm{Rec})$ définissant un Secure Sketch et pour une fonction f de Cancelable Biometrics, on définit une fonction d'enrôlement et une fonction de vérification par:

- $\mathsf{Enrol}(b;\, f) = \mathrm{Sk}(f(b))$

- $\mathsf{Verif}(b;\, f;\, P) = \mathrm{Rec}(P,\, f(b'))$.

L'analyse de sécurité d'une telle construction est fournie en partie IV.2.

Identification Anonyme grâce aux Cancelable Biometrics

Un avantage de la construction "Cancelable Biometrics" est que l'espace d'arrivée après transformation est le même que l'espace des templates biométriques. La conséquence est qu'il est possible de transformer les templates plusieurs fois de manière consécutives, au prix d'une dégradation des performances (de l'information est perdue à chaque transformation).

La construction suivante, détaillée en IV.3, permet d'effectuer des requêtes d'identification biométrique qui ont la propriété d'intraçabilité. Un tel système empêche un attaquant de différencier deux différentes requêtes biométriques provenant de la même personne de deux requêtes biométriques provenant de deux individus.

Soit f, $g_{t,in}$, $g_{t,out}$ des fonctions de type "Cancelable Biometrics", et $h_{t,i}$ des fonctions bijectives indexées par t (un paramètre homogène à une mesure de temps) et i (homogène à l'identifiant d'un capteur). La construction proposée se tient entre un serveur et un terminal biométrique, et est la suivante:

- Tous les traits biométriques enrôlés sont stockés sur le serveur sous la forme de $f(b)$;
- Lors d'une requête d'identification:
  - Serveur et terminal s'authentifient mutuellement,
  - Le serveur envoie au terminal une fonction $\Phi_{in} = h_{t,i}^{-1} \circ g_{t,in} \circ f$;
  - Le terminal envoie au serveur le trait biométrique déformé $\Phi_{in}(b')$;
  - Le serveur termine la déformation en calculant $\Phi(b') = g_{t,out} \circ h_{t,i}\left(\Phi_{in}(b')\right)$;
  - Le serveur identifie alors $\Phi(b')$ en le comparant à l'ensemble de sa base.

Architectures pour l'Identification Biométrique Sécurisée

Les constructions décrites précédemment montrent leurs limites lorsqu'elles sont confrontées à des attaquants d'une part, ou lorsque les taux d'erreurs biométriques rentrent en compte. Les solutions que nous présentons ici visent donc à remplacer les méthodes venant de la biométrie par des méthodes venant du monde de la cryptographie. Ce faisant on peut assurer un niveau minimal de sécurité, en se basant sur des hypothèses calculatoires éprouvées par une grande communauté. Nous présentons donc des constructions dans le modèles de cryptographie à clé publique, cryptographie symétrique, et cryptographie matérielle.

Identification Biométrique sur Bases de Données Chiffrées

Présentée en V, la question de l'identification biométrique sur des données chiffrées se pose naturellement lorsqu'on pose le problème de la protection des données. Nous proposons de nous inspirer de primitives cryptographiques avancées, tel le chiffrement trouvable, pour obtenir le résultat escompté.

Chiffrement Trouvable Tolérant aux Erreurs

Effectuer une requête d'identification contre une base de données entièrement chiffrée revient à chercher le plus proche voisin d'une base de données sans avoir accès aux éléments de la base déchiffrés. Pour cette raison, nous introduisons la notion de *Chiffrement Trouvable Tolérant aux Erreurs* (Error-Tolerant Searchable Encryption).

Définition. *Une primitive de* Chiffrement Trouvable Tolérant aux Erreurs *paramétrée par* $(\epsilon, \lambda_{min}, \lambda_{max})$ *est donnée par trois méthodes probabilistes et exécutées en temps polynomial* $(\mathsf{KeyGen}, \mathsf{Send}, \mathsf{Retrieve})$:

- $\mathsf{KeyGen}(1^k)$ *initialise le système et génère un couple de clés publique et secrète* (pk, sk)*.* k *est le paramètre de sécurité;* pk *est utilisée pour envoyer des données sur un serveur, alors que* sk *sert à récupérer des informations du serveur.*

- $\mathsf{Send}_{\mathcal{X},\mathsf{S}}(x, pk)$ *est un protocole où l'utilisateur* $\mathcal{X}$ *envoie au serveur* S *l'élément* $x \in \{0,1\}^N$ *à enregistrer dans le système. A l'issue du protocole, un identifiant unique* $\varphi(x)$ *est associé à* x*.*

- $\mathsf{Retrieve}_{\mathcal{Y},\mathsf{S}}(x', sk)$ *est un protocole où l'utilisateur* $\mathcal{Y}$ *demande les identifiants de toutes les données stockées sur* $\mathcal{S}$ *qui sont proches de* x'*, en respectant les propriétés de* Complétude(λ_{min}) *et de* ϵ-Sûreté(λ_{max})*. Le résultat est noté* $\Phi(x')$*.*

Les conditions de complétude et de sûreté sont définies ci-après:

Définition. *Soient* $x_1, \ldots, x_p \in B = \{0,1\}^N$ p *vecteurs binaires différents, et* $x' \in B$ *un nouveau vecteur. En supposant que le système ait été initialisé, que tous les messages* x_i *aient été envoyés par l'utilisateur* $\mathcal{X}$ *au serveur* $\mathcal{S}$*, et associés respectivement aux identifiants* $\varphi(x_i)$*; en supposant que l'utilisateur* $\mathcal{Y}$ *ait obtenu par* $\mathsf{Retrieve}$ *l'ensemble* $\Phi(x')$:

- *Le système est* Complet(λ_{min}) *si la probabilité suivante :*

  $$\eta_c = \Pr_{x'}\left[\exists i \text{ s.t. } d(x', x_i) \leq \lambda_{min}, \varphi(x_i) \notin \Phi(x')\right]$$

  est négligeable;

- *Le système est* Sûr(λ_{max}) *si la probabilité suivante :*

  $$\eta_s = \Pr_{x'}\left[\exists i \in [\![1, p]\!] \text{ s.t. } \ d(x', x_i) > \lambda_{max}\ , \varphi(x_i) \in \Phi(x')\right],$$

  est plus petite que ϵ.

Ces deux conditions expriment le fait que le schéma est fonctionnel, c'est à dire que les bons identifiants seront retournés par la méthode Retrieve, et que des identifiants erronés apparaissent avec une probabilité faible.

Le respect de la vie privée des utilisateurs est en revanche défini par des conditions de sécurités nommées *Sender Privacy*, *Receiver Privacy* et *Symmetric Receiver Privacy.* De manière informelle, il y a Sender Privacy si $\mathcal{S}$ n'a pas d'information sur les données envoyées par $\mathcal{X}$; il y a Receiver Privacy si $\mathcal{S}$ n'a pas d'information sur les données réclamées par $\mathcal{Y}$; enfin, il y a Symmetric Receiver Privacy si $\mathcal{Y}$ n'obtient de $\mathcal{S}$ pas plus d'information que celles qui le concernent directement.

Des constructions réalisant ces schémas sont illustrées ci-dessous et présentées en détail en V.2.

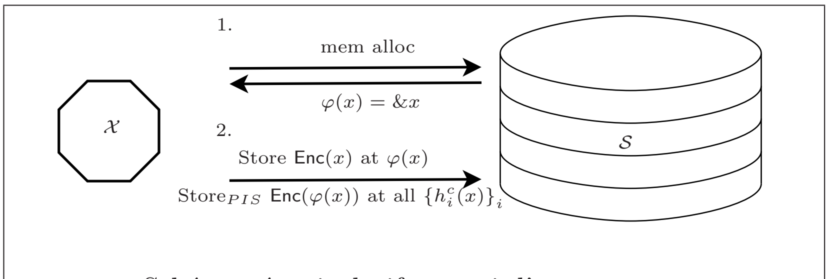

1.
mem alloc
$\varphi(x) = \&x$
$\mathcal{X}$
2.
Store $\mathsf{Enc}(x)$ at $\varphi(x)$
Store$_{PIS}$ $\mathsf{Enc}(\varphi(x))$ at all $\{h_i^c(x)\}_i$
$\mathcal{S}$

Schéma récapitulatif : envoi d'un message.

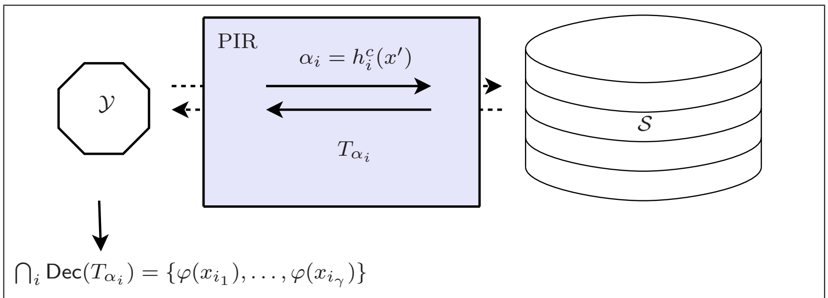

PIR
$\alpha_i = h_i^c(x')$
T_{α_i}
$\mathcal{Y}$
$\mathcal{S}$
$\bigcap_i \mathsf{Dec}(T_{\alpha_i}) = \{\varphi(x_{i_1}), \ldots, \varphi(x_{i_\gamma})\}$

Schéma récapitulatif : récupération de messages voisins.

Identification Chiffrée

A partir d'un schéma de chiffrement trouvable tolérant aux erreurs, il est possible de construire une architecture d'identification biométrique sur des données chiffrées, en procédant ainsi.

Pour enrôler un utilisateur, un capteur biométrique capture un trait b_i; l'enrôlement est effectué *via* $\mathsf{Send}_{\mathcal{X},\mathcal{S}}(b_i, pk)$.

Pour identifier un trait biométrique b', il suffit d'exécuter $\mathsf{Retrieve}_{\mathcal{Y},\mathcal{S}}(b', sk)$

Ainsi, l'élément $\mathcal{Y}$ qui exécute la requête obtient une liste de références aux candidats à l'identification de b'; on peut dans une étape ultérieure appliquer un protocole de matching sécurisé pour finaliser l'opération.

Identification Basée sur du Chiffrement Symétrique

Afin de préserver le caractère secret des données qui sont questionnées, la construction précédente utilise un PIR (protocole de retrait d'information privé). Ces protocoles sont très coûteux en terme de capacité de calcul, car un protocole effectue nécessairement un nombre d'opérations cryptographique linéaire en la taille de la base de données. Dans le but d'accélérer les calculs, nous présentons en partie VI un protocole qui se base uniquement sur de la cryptographie à clé secrète (ou cryptographie symétrique).

Le résultat de cette partie se résume dans la donnée du protocole d'identification (c'est à dire la donnée des trois méthodes `Initialize`, `Enrolment` et `Identification`) et dans les propriétés de sécurité qui y sont associées, à savoir la complétude, la sûreté, la confidentialité adaptative et l'indistingabilité non-adaptative. Toutes ces notions sont définies rigoureusement en partie VI; nous donnons ici la description des algorithmes d'enrôlement et d'identification:

```
Enrolment($b_1, \dots, b_N, ID_1, \dots, ID_N, \mathcal{K}$):

  • Initialisation:
      - construire $\Delta = \{(h_i(b_k), i); \quad i \in [\![1, \mu]\!], k \in [\![1, N]\!]\}$
      - pour chaque $\omega \in \Delta$, construire $\mathcal{D}(\omega) = \{ID_\omega^j\}_j$
        l'ensemble des identifiants des utilisateurs $\mathcal{U}_k$ tels
        que $(h_i(b_k), i) = \omega$
      - calculer max $= \max_{\omega \in \Delta}(|\mathcal{D}(\omega)|)$ et $m =$ max $\cdot |\Delta|$

  • construire l'index T:
      - pour chaque $\omega \in \Delta$
            pour $1 \leq j \leq |\mathcal{D}(\omega)|$
                définir T $[\pi_K(\omega \parallel j)] = \mathcal{E}_{sk}(ID_\omega^j)$
      - si $m' = \sum_{\omega \in \Delta} |\mathcal{D}(\omega)| < m$, remplir les $(m - m')$
        cellules de T restantes avec des valeurs aléatoires.

  • Retourner $\mathcal{I} =$ T
```

La procédure d'enrôlement.

$\underline{\texttt{Identification}}(\mathcal{K}, b')$:
Phase de recherche: Lorsqu'un utilisateur $\mathcal{U}$ veut s'identifier, le capteur mesure son trait biométrique b'. Il évalue alors chaque fonction LSH sur b' et obtient $\omega_i = (h_i(b'), i)$, $i \in [\![1, \mu]\!]$. Le capteur envoie alors au serveur les trappes:

$$T_{\omega_i} = \texttt{Trapdoor}(K, \omega_i) = (\pi_K(\omega_i, 1), \ldots, \pi_K(\omega_i, \max))$$

Le serveur exécute un algorithme SSE `Search` sur les différentes trappes T_{ω_i}, obtenant ainsi un ensemble de cellules du tableau `T`. Le serveur renvoie au capteur le tableau $\mathcal{ID}(b')$ qui agrège les résultats de recherche :

$$\mathcal{ID}(b') = \begin{bmatrix} \mathcal{E}_{sk}(ID_{k_1,1}) & \cdots & \mathcal{E}_{sk}(ID_{k_1,\texttt{max}}) \\ \vdots & \ddots & \vdots \\ \mathcal{E}_{sk}(ID_{k_\mu,1}) & \cdots & \mathcal{E}_{sk}(ID_{k_\mu,\texttt{max}}) \end{bmatrix}$$

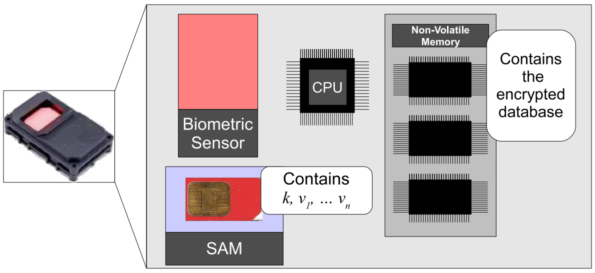
où chaque ligne est le résultat d'un $\texttt{Search}(T_{\omega_i})$.
Phase d'identification: le capteur déchiffre tous les identifiants reçus, et compte le nombre d'occurrence de chaque identité. Il renvoie alors la liste des identités qui apparaît plus de $\lambda\mu$ fois, c'est à dire les $\{ID(\mathcal{U}_l)\}$ telles que :

$$\sum_{i=1}^{\mu} \sum_{j=1}^{\texttt{max}} \delta_{ID(\mathcal{U}_l), ID_{k_i,j}} > \lambda\mu.$$

La procédure d'identification.

Identification Locale Sécurisée

Passer de la cryptographie à clé publique à la cryptographie symétrique permet de réduire les complexités calculatoires des primitives, mais se fait au prix d'un affaiblissement du modèle sous-jacent. Un des problèmes qui se présente alors est qu'il n'est plus possible, dans la construction proposée, d'enrôler les utilisateurs un à un. De plus, on renonce dans ce modèle à la *Symmetric Receiver Privacy.* Or, dans un grand nombre de situation, l'identification peut se faire uniquement sur une base de données locale (par exemple lorsqu'on considère l'accès à une zone sécurisée d'un bâtiment). Dans ce cas on peut baser la sécurité sur une architecture matérielle, suivant le schéma suivant.



Un terminal d'identification biométrique sécurisée.

Cette architecture utilise deux représentations différentes pour un même trait biométrique : d'une part une représentation dédiée à la comparaison sur une carte à puce (Match-On-Card), et d'autre part une représentation quantifiée qui permet d'effectuer des opérations de comparaison rapidement. Il va de soi que la seconde représentation est moins efficace que la première; lors du choix des fonctions de quantification, les paramètres importants pour une architecture efficace seront la taille des templates quantifiés (qui sont stockés sur la carte à puce, donc des templates plus petits impliquent de plus grandes bases de données) et leur précision, qui détermine le nombre d'opérations de Match-On-Card nécessaire pour identifier avec suffisamment de certitude le trait biométrique mesuré.

La sécurité de ce schéma repose donc d'une part sur une fonction de chiffrement, qui protège les templates dans une mémoire non-volatile, et d'autre part sur un matériel sécurisé de type carte à puce, qui contient les templates quantifiés.

Utilisation Cryptographique des Codes d'Identification

La partie 4 de ce mémoire traite d'utilisations différentes des codes d'identification. Dans ce résumé, nous donnerons la définition des codes d'identification, et présenterons les différents résultats obtenus dans les chapitres IX et X.

Définition et Constructions de Codes d'Identification

La définition des codes d'identification [3] suit.

Définition. *Soient $\mathcal{X}$ et $\mathcal{Y}$ deux alphabets. Pour $n, N \in \mathbb{N}$, $\lambda_1, \lambda_2 \in [0,1]$, , W^n un canal de $\mathcal{X}^n$ dans $\mathcal{Y}^n$, un $(n, N, \lambda_1, \lambda_2)$-code d'identification de $\mathcal{X}$ vers $\mathcal{Y}$ est donné par un ensemble de $\{(Q(\cdot|i), D_i)\}_{i \in [\![1,N]\!]}$, où, pour $i, j \in [\![1, N]\!]$, $i \neq j$:*

- *$Q(\cdot|i)$ est une loi de probabilités sur $\mathcal{X}^n$, appelée loi de codage de i;*
- *$D_i \subset \mathcal{Y}^n$ est l'ensemble de décodage de i;*

- *λ_1 et λ_2 sont les erreurs de premier et second type, définis par :*

$$\lambda_1 \geq \sum_{x^n \in \mathcal{X}^n} Q(x^n|i) W^n(\overline{D_i}|x^n)$$

et

$$\lambda_2 \geq \sum_{x^n \in \mathcal{X}^n} Q(x^n|j) W^n(D_i|x^n)$$

Un code d'identification sert ainsi à répondre à la question "est-ce que le message i a été envoyé?" et non pas à la question "quel message a été envoyé?". Les codes d'identifications sont une relaxation de la définition de code de transmission, et permettent donc d'envoyer un nombre de messages supérieur à capacité de canal constante.

Parmis les constructions de code d'identification existantes, nous étudions plus particulièrement dans ce mémoire celle proposée par Moulin et Koetter [124], qui utilise des codes de Reed-Solomon et que nous généralisons ici :

Définition. *Soit $C =$ un code correcteur d'erreur sur $\mathcal{X}$ de longueur n, taille N et de distance minimale (de Hamming) d. Notons le i-ème mot de code $c_i = (c_i^1, \ldots, c_i^n)$. Alors C induit un code d'identification de $\mathcal{X}^n$ vers lui-même défini par :*

- *D_i est défini par $\left\{(j, c_i^j)\right\}_{j \in [\![1,n]\!]}$;*
- *$Q(\cdot|i)$ est la loi uniforme sur D_i;*
- *$\lambda_1 = 0$ et $\lambda_2 = 1 - \frac{d}{n}$.*

Une instance particulière de cette construction est décrite par [124] lorsque $\mathcal{X} = \mathbb{F}_q$ et C est un code de Reed-Solomon q-aire de longueur n et de dimension $k = \log_q N$. Soit $F = \{\alpha_1, \ldots, \alpha_n\} \subset \mathbb{F}_q$ un sous-ensemble de n valeurs du corps $\mathbb{F}_q$. En associant à chaque $i \in [\![1, N]\!]$ un unique polynôme P_i de $\mathbb{F}_q[X]$ de degré strictement inférieur à k, l'ensemble de décodage D_i est défini par $\{(j, P_i(\alpha_j))|j \in [\![1, n]\!]\}$. De même, la loi de codage de i est la loi uniforme sur D_i.

Solutions au Problème de la Ligue

Le chapitre IX propose d'étudier un problème de théorie de l'information appelé le problème de la Ligue. Deux joueurs Alice et Bob ont des morceaux d'information différents, et l'objectif est de transmettre l'information entre les deux joueurs en communiquant le moins de bits possible. Par exemple, Alice s'intéresse au résultat d'un match entre deux équipes; elle connaît le nom des deux équipes à jouer. De son côté, Bob sait quelles équipes ont participé à la ligue entière, et connaît par ailleurs le nom de l'équipe gagnante. Comment faire en sorte que Alice puisse connaître le nom du gagnant de manière efficace?

Alors que les solutions classiques sont présentées, ainsi que leur caractère optimal, nous nous intéressons à une autre question qui avait été abordée par [133], qui est de savoir si on peut, dans ce cas, réduire le nombre de bits à communiquer tout en maintenant une probabilité d'erreur asymptotiquement nulle.

En utilisant dans une première partie des codes d'identification, et, dans une seconde partie, une construction explicite, on montre qu'il est possible d'obtenir de très faibles probabilités d'erreurs, avec un protocole en deux passes qui demande uniquement $O(\log\log\log n)$ bits de communication où n est le nombre d'équipes participant à la ligue.

Les résultats de ce chapitre sont exposés dans le tableau suivant.

	Une passe		Deux passes
Sans erreur	$\lceil \log n \rceil$	Codage entropique *optimal*	$1 + \lceil \log\log n \rceil$ *optimal [132]*
λ erreurs possibles	$\frac{\log\log n}{1-\epsilon}$ $\lceil \log\log n \rceil$	Codes d'Identification Section IX.3	$1 + \lceil \log\log\log n \rceil$ Section IX.3

Protocole d'Interrogation d'Etiquettes Electroniques

Ce mémoire se conclut sur la présentation d'un protocole d'interrogation d'éléments communicants sans fil. Le modèle est le suivant : une station de base interroge dans un ordre aléatoire qu'elle détermine elle-même, un ensemble d'étiquettes électroniques communiquant par ondes électromagnétiques. Ceci peut être fait pour un inventaire des étiquettes présentes dans un rayon, mais il peut également s'agir d'un réseau de capteurs mesurant une donnée quelconque (la température, l'humidité, la pression...) qui transmettent les informations *ad hoc* à la station de base à la demande.

La question de la sécurité se pose dans ce contexte : comment s'assurer que la station de base communique bien avec un élément sans fil. En d'autres termes, comment s'assurer qu'un imposteur ne prend pas la place de l'un ou de l'autre des partenaires? Par ailleurs, dans le cas de communications sans fil, le respect de la vie privée des éléments signifie qu'il est impossible pour une personne qui écoute toutes les communications de dire si une communication donnée émane d'une étiquette ou non.

La sécurité et la vie privée sont formalisées par un modèle mathématique à base d'oracles du à [182]. La contribution présentée ici propose d'appliquer les codes d'identifications à cette situation, et de démontrer, dans le cas des codes de Moulin-Koetter basés sur l'évaluation polynomiale, que sécurité et vie privée sont bien respectées pour une certaine classe de paramètres. Le schéma suivant résume le protocole dans le cas particulier des codes de Moulin-Koetter, et qui peut bien entendu être généralisé à toute famille de codes d'identification.

Etiquettes	paramètres	Vérifieur
Identifiants p, p'	$\mathbb{F}_q, (\alpha_1, \dots, \alpha_n)$	(l, p_l, p'_l)
	$\xleftarrow{(ACK, j, a=p_l(\alpha_j))}$	Choisis j
If $p(\alpha_j) = a$	$\xrightarrow{(ACK, b=p'(\alpha_j))}$	Vérifie l'égalité $p'_l(\alpha_j) = b$

Identification d'éléments sans-fil par les codes d'identification de Moulin-Koetter

Le chapitre X étudie les propriétés de sécurité et de vie privée de ce protocole. Nous y considérons tout d'abord le cas calculatoire, en supposant que le problème de reconstruction polynomiale est difficile. Puis dans un second temps, nous cherchons à avoir une estimation de la taille des paramètres qui assurent une sécurité inconditionnelle, c'est à dire en s'affranchissant d'hypothèses calculatoires.

Nous montrons dans ce chapitre que les propriétés souhaitées se ramènent directement au problème de la reconstruction polynomiale, dont la définition est donnée ici.

Définition. **[Problème de la Reconstruction Polynomiale]** *Soient n, k, t tels que $n \geq t \geq 1$, $n \geq k$ et $z, y \in \mathbb{F}_q^n$, avec $z_i \neq z_j$ pour $i \neq j$. Le problème noté $PR^z_{n,k,t}$, est défini comme tel :*

Calculer et renvoyer tous les (p, I) où $p \in \mathbb{F}_q[X]$, $\deg(P) < k$, $I \subset [\![1, n]\!]$, $|I| \geq t$ et $\forall i \in I, p(z_i) = y_i$.

L'algorithme de Guruswami-Sudan fournit une solution à ce problème lorsque $t \geq \sqrt{kn}$. En revanche, aucun algorithme connu ne résout ce problème en temps polylogarithmique lorsque $t < \sqrt{kn}$.

En notant M le nombre d'éléments sans-fils interrogés, et T le nombre maximal d'interrogations jouées, et en nous basant sur l'hypothèse que le problème est difficile lorsque $t < \sqrt{kn}$, nous démontrons les résultats suivants:

Proposition. *Sous la condition $\sqrt{q} \geq M \geq e\sqrt{\frac{n}{k}}$ et $T < M^2 k$, un adversaire ne peut se faire passer pour une étiquette électronique non-corrompue sans rejouer une communication existante qu'avec probabilité $\frac{1}{q}$.*

La proposition suivante utilise la notion de *weak privacy* définie en VIII.3.

Proposition. *Sous les mêmes hypothèses, le schéma est weak private.*

Ceci fournit donc toute une classe de paramètres n, k, q, T, M pour lesquels le schéma est utilisable. En pratique, sur des étiquettes électroniques, on choisira de faibles valeurs de $n \log_2 q$ en raison des limitations de mémoires sur des étiquettes électroniques destinées à être produites en masse.

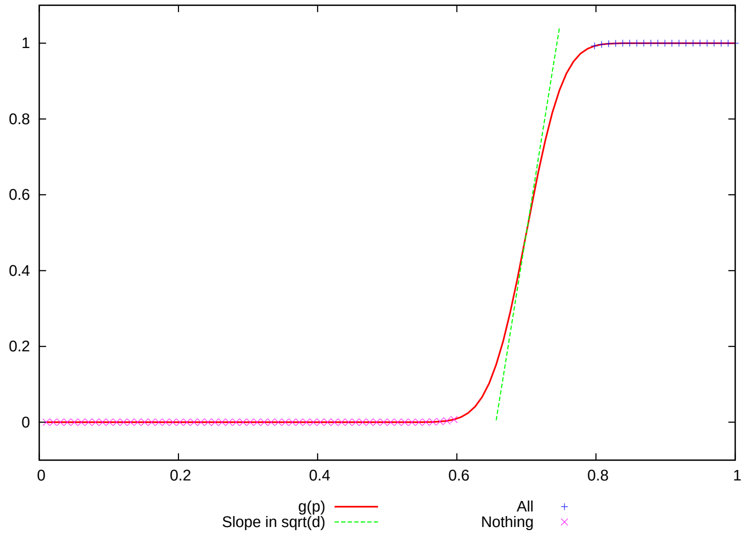
Le Seuil des Codes de Reed-Solomon

Une manière de s'affranchir de l'hypothèse calculatoire citée plus haut est de considérer le seuil des codes de Reed-Solomon. Nous montrons dans le chapitre X que tous les codes q-aires ont un seuil, c'est à dire que si le nombre d'erreurs reçues dépasse la valeur de ce seuil, il devient pratiquement impossible de retrouver le mot de code original. Cette propriété est atteinte en démontrant une version q-aire de l'identité de Margulis et Russo:

$$\frac{d\mu_p(U)}{dp} = \frac{1}{p}\int_U h_U(x)d\mu_p(x).$$

Ici U est un sous-ensemble de $\mathbb{F}_q^n$, et les fonctions μ_p et h_U sont définies dans la partie idoine.

Cette formule permet en particulier de calculer une formule qui majore la probabilité de décodage d'un mot de code avec erreurs par une fonction à seuil de telle sorte que la probabilité de décodage ressemble à la figure suivante :



Pour conclure, nous montrons qu'il est possible de situer le seuil des codes de Reed-Solomon de manière assez précise. En effet, les codes de Reed-Solomon sont séparables au maximum de la distance, et on connait donc la distribution des poids du code. Sous l'hypothèse que le code est "court" (c'est à dire que n est petit devant q), on peut alors approximer la probabilité de décodage correct $g(p)$ par :

$$\log_q g(p) \leq \max_{l\in[d,pn]} (1 + l - d - pn + n\mu(l, pn)) + o_q(1).$$

avec $d = n - k + 1$ la distance minimale du code, et μ une fonction liée à l'intersection de boules de Hamming définie ultérieurement.

En injectant ce résultat dans la construction précédente, on peut donc choisir deux modes de réalisation du protocole. Le premier est sûr de manière inconditionnelle; en revanche il autorise moins de communications avec le même élément. Le second est sûr sous l'hypothèse qu'il est difficile de reconstruire un polynôme avec moins d'information que la borne de Guruswami-Sudan. Alors que cette hypothèse parait plus faible, elle reste vérifiée aujourd'hui.

Introduction

It is expected from a courteous man to present himself before addressing any question. This is the case for face to face communication (”My name is Bob, nice to meet you.”), but also in telephonic conversations, where the caller wants to make sure he got the right speaker, and the receiver wants to know who is calling him. To prevent people from lying about their identity, a country's authorities usually deliver identification documents (the ID card) to their inhabitants. Therefore, the identity of someone is first stated by the person himself, then proved by a document that carries the seal of a trusted party.

In a world where the communication is established from one computer to the computer, the question of ”who am I talking with?” is crucial. He who does not take this issue seriously is threatened by many attacks. For example, getting infected by malicious programs when clicking on a link that points to a server owned by a pirate, or being subject to phishing (confused a webpage with a more familiar one because they look alike). How does a server prove its identity to a computer, and to the end-user? Paradoxically enough[2], this question is answered by modern cryptography. Electronic signatures became the successors of wax seals, and protocols were constructed to prove someone's identity. Authentication is now a widely studied topic, and Identification follows closely.

In this document, we focus on identification of people through the use of their characteristics, as well as that of communicating devices, with wireless protocols. The elements that can identify someone are of three sorts: elements that the person *own*, information that the person *knows* and what the person *is*. That last element, biometrics, has most particularly been the subject of our attention in this document, along with other identifying elements.

Even though biometrics and cryptography are sciences studied for more than a century, there were until recently very few constructions that enabled to use biometrics in cryptographic protocols, or, to state it differently, to apply cryptographic primitives to biometric recognition. A reason for that is that there was little use of secure primitives for the historical uses of biomet-

[2] Cryptography was originally the art of keeping things hidden, but also serves as a way for people and devices to disclose information.

rics. Nowadays, biometrics can be the main tool for identification of people, and there is a serious need for cryptographic support in order to protect the databases that contain the personal data of enrolled users.

This thesis is organised in four parts.

Identification Elements. This introductory part presents the objects we deal with in the following sections. **Chapter I** explains the three varieties of identifying elements: biometrics, tokens and paswords. **Chapter II** introduces the different biometrics we will be using, most particularly the fingerprint and the iris. It also provides basic explanations on the other identifying elements. However, we do not go into a detailed taxonomy of all the possible elements. We essentially focus on the most useful ones for the constructions we provide.

Protecting Biometric Templates. Biometrics was quickly acknowledged as very sensitive data, but solutions for the protection of biometric elements did not appear as fast as one could expect. This part presents the most considered contributions to the protection of biometric databases. **Chapter III** is dedicated to Secure Sketches, Fuzzy Extractors and Cancelable Biometrics. It goes through a description of these methods and algorithms, to finally analyse the security of these constructions. As it appeared that the countermeasures introduced by these methods were not attack-proof, we extend their strength in some constructions in **Chapter IV**. Our first contribution is to provide a coding algorithm (a product-code with its iterative decoder) that is well adapted to biometric elements, as we show in our experiments. We also show how it is possible to combine Cancelable Biometrics with Secure Sketches, in order to keep the advantages of both methods, and, even better, to sum up their security. Finally, we show an exotic construction that uses Cancelable Biometric as the building block. This construction mimics the One-Time Password behaviour in the biometric domain.

Cryptography Applied to Biometric Data. Setting aside biometric methods, we focus on applied cryptography, and show how it is possible to design identification protocols that are well adapted to biometrics. We provide three constructions. In **Chapter V**, the first one uses public-key cryptography, and is designed to respect the users' privacy as much as possible - and indeed, the privacy requirements that the scheme offers define a high standard. The second construction, main topic of **Chapter VI**, uses Symmetric Searchable Encryption instead of asymmetric cryptography. It is a computationally more efficient method, but unfortunately, the security model is weaker. Nevertheless, the privacy properties that are achieved by this method are very interesting in a realistic model. In **Chapter VII**, the last construction uses hardware-based security to identify a user's biometric template out of a local database.

Identifying Wireless Devices. Finally, the problematic of identifying elements is raised in this last part. Focusing in **Chapter VIII** on the specificities of wireless communication, we enumerate some specific threats against the security of the communication and the privacy of the elements. This part uses the notion of Identification Codes, a coding structure seldom used in the literature, to identify elements in wireless protocols. With these codes, it is possible to achieve in **Chapter IX** very low communication costs. Moreover, as their design is based on a probabilistic encoding, **Chapter X** shows that they have very good properties from the privacy and security point of view, in an interrogation protocol between a verifier and wireless devices. The range of parameters within which the protocol is secure is thoroughly studied in the case of Reed-Solomon based identification codes.

For the sake of completeness, we added to these parts appendices that recall classical elements of Information Theory (Appendix A), Coding Theory (Appendix B), and Cryptography (Appendix C). Appendix D presents a construction that uses these three aspects of computing and communication science for a key establishment protocol. The characteristics of this protocol is that it is adapted to low-cost wireless devices, and is resistant against active adversaries.

This work make contributions to different fields of computer science; these fields may seem distant but they share a common application: the secure identification of people and devices. We hope that putting together these different results will encourage researchers of the community to work jointly for a better security and privacy.

Part 1

Biometrics, Templates and Physical Elements

Chapter I

Identification Elements

> As far as the laws of mathematics refer to reality, they are not certain; and as far as they are certain, they do not refer to reality.
>
> Albert Einstein

This chapter will present the different techniques that have been used to identify people - and, as a logical extension, objects. It happens that throughout time, mankind switched from using one of the most secure forms of identification (biometrics) to one of the less secure ways of identifying people, which is passwords. One of the goals of this thesis is to provide the tools for going back from secret keys to using biometrics, *i.e.* renew the link between a man's virtual and physical identity[1].

I.1 "Natural" Identification

Biometric recognition is the most natural security measure, and the one that human beings are the most at ease with. Indeed, this is the most natural way of recognizing people, as the brain is trained even before the birth to recognizing sounds, images (static or in motion), and every other input that is received, sometimes unconsciously. This intuition is confirmed by experiments made on cats and kittens [68], showing that the cerebral response to a natural image is much higher than the response to noise or to geometric shapes uncorrelated to a natural context.

In a more practical way, the human mind is used to biometric recognition, as it was trained to recognize faces, voices, even smells and gaits. All the senses contribute to a very solid identification of people - after a reasonable

[1] A man's physical identity is defined by his body; his virtual (or social) identity consists in names, pseudonyms, *etc.* deriving from conventions.

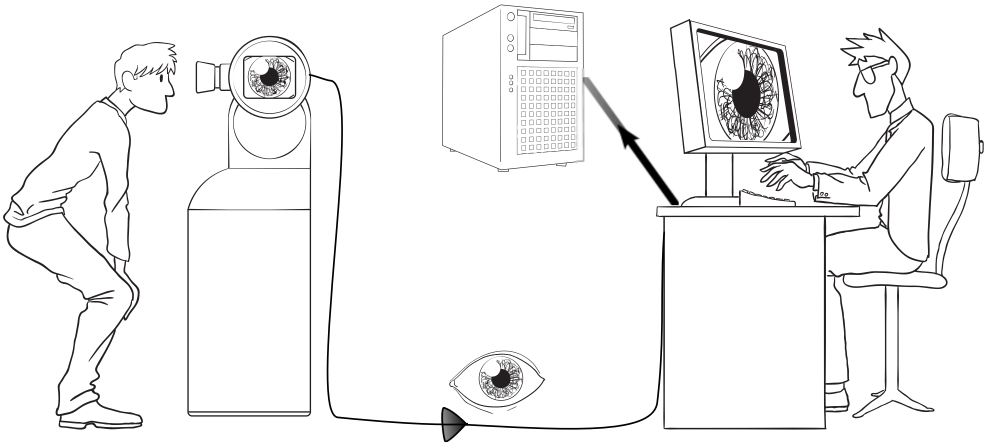


Figure I.1: Enrolment process of a user

period of training. Reproducing and automatizing this ability is not an easy task, and the elements that work the best for biometric identification are not the ones mentioned above, but elements that human take much more time recognizing (*e.g.* fingerprints).

In a practical biometric system, users first need to be registered by providing a measurement of their biometric trait. This is called *enrolment* (see figure I.1). After enrolment, the system is able to recognize the users from a fresh biometric capture.

Throughout this document, we will focus essentially on two varieties of biometrics, the fingerprint and the iris. However, other biometrics are worth noting, and will be briefly introduced.

The fingerprint

Fingerprints are the most ancient biometric elements used for document authentication. They are composed of ridge lines at the surface of the finger, and are modelled *in utero*. The pattern is the same from childhood to old age; when a finger is hurt, scars alter the ridge flow slightly, but minor injury do not impede the biometric matching algorithms.

It is believed that each fingerprint is unique. Indeed, each finger as a significantly different fingerprint, and even homozygous twins do not have the same fingerprints. So far, the most convenient way to distinguish fingerprints is to compare the general shape of the fingerprints, and then their *minutiae*, which are singular points of the ridge flow (points where a ridge line ends, or where a ridge line is separated into two different lines). There are also characteristic points called *cores* and *deltas*. These are singularities of the ridge flow field, a core is the centre of a loop-type singularity, and a delta, of a triangle-type. However, it is not sufficient to only use cores and deltas to recognize biometrics, since their detection and comparison is a tricky part; besides, not all the fingerprints possess these points.

While it is usually enough to use the general shape, cores, deltas, and minutiae, to compare two fingerprints, it can also be useful to take into account more detailed information, like the precise design of the ridge lines, or even the pores (captured with high-definition sensors).

Difficulties of fingerprint matching

Fingerprints are unique, but so are the measurements of the same fingerprint. Indeed, a fingerprint is measured by pressing the finger against a surface, and, depending on the dryness of the finger, the temperature, the angle and strength of pressure applied, the position of pressure, or even the cleanness of the sensor, the measurement will be different. This means that to match two minutiae map, we need to take into account:

- the elasticity of the skin, and the distortion map induced;
- the presence or absence of minutiae from one capture to the other;
- a limited overlap between two templates;
- the time-variability due to burns, scars, or any kind of damage;
- *etc.*

Even with these constraints, fingerprint matching is still pretty efficient, with the modern matching algorithms, due to advances in computer science, industrial competition, and the exciting challenge of reliably identifying people among huge databases.

Fingerprint databases

In order to compare the fingerprint matching algorithms, it is necessary to establish some reference databases. For this, the Fingerprint Verification Competition (**FVC**) was established in 2000 [70], organised by four different universities that established four sets of databases in 2000, 2002, 2004 and 2006. Each year, they assembled three databases of "real" fingers captured with different sensors, and a fourth database generated synthetically. These data sets are public, and they are pretty interesting to compare the performances (in terms of error rates) of the different algorithms published in the literature. These data sets are made of 800 pictures of 100 different fingers, 8 pictures per finger.

Another known dataset is the **MCYT** (Ministerio de Ciencia y Tecnología) Fingerprint database. This database is a bimodal collection of fingerprints and signatures collected by several Spanish universities. For our experiments, we used only a subset made of 10 captures of 100 fingerprints.

Some papers publish their results on private datasets. Doing this, they are able to enhance the results by adapting the algorithms to the specificities of the private database that they could be aware of. This practice also makes the results non-reproducible, as we have to reimplement the methods described in a paper if we want to compare the performances to other constructions.

The Iris

The iris is the eye muscle that can be seen between the pupil (black) and the sclera (white). It is the coloured part of the eye. Its function is to expand or reduce pupil surface so that the eye adapts to the light. The iris pattern is a mosaic of thousands of pigmented cells, and this pattern is believed to be unique to the eye. As for the fingerprint, the iris pattern is made *in utero* and differs from the right to the left eye. Twins may also have similar eyes, but they have independent iris patterns.

Remark 1. The iris recognition technology is different from the retinal scan. The iris is the visible and coloured part of the eye while the retina is a tissue made of neural visual cells, at the back of the eye. A simple (infra-red) picture suffices for the recognition of the iris, but a retinal scan requires the user to put his eye against a scanner's eyepiece.

Matching Irises

The mainstream (and historical) technology used for iris recognition is the IrisCode, detailed in section II.1. This is however not the only method available, and one can cite, among others, [62, 169], as different methods used for matching irises.

Iris Databases

As was the case for fingerprints, there are few public iris databases of sufficient size. However, the two sets on which we made experiments were the ICE set [128], gathered by the US National Institute of Standards and Technology (NIST), and the CASIA set [47], collected by the Chinese Center for Biometrics and Security Research (CBSR).

- The ICE 2005 database: [114, 128].

  It contains 2953 images coming from 244 different eyes. It is taken without modification but one slight correction: the label of the eye 246260 has been switched from left to right. In this dataset the number of images for each eye is variable.

- The CASIA database: [47].

> This is the first version of the Chinese Academy of Science public iris database. It contains 756 pictures of 108 different eyes, with 7 pictures per eye.

Note that other datasets exist, as well as other versions of the CASIA and ICE datasets. Using a dataset or another is an arbitrary choice made for the purpose of some experiment.

Other biometrics

Though our work was essentially aimed to match the fingerprint and the iris, biometrics methods are pretty general and can be applied to any kind of "fuzzy data". The problematic is essentially to find methods to integrate the noise issue to the different protocols at stake.

Among other biometrics on which the methods that are described in the rest of this manuscript apply, we can cite face, voice, palmprint, shape of the hand, gait, vein network recognition.

This whole range of biometrics is being investigated world-wide; some of the features are more interesting than others. For example, the face requires minimal material, and leaves no trace behind. On the other hand, it is not as unique as the fingerprint, as twins - or even members of the same family - do resemble each other. However, biometric fusion enables us to take into account multiple biometrics and to answer accurately by fusing the multiple matching scores.

Remark 2. While the classical issues in cryptography are the key size, encryption and decryption speed for a given level of security, the main concerns with biometrics are the error rates and matching speed.

Throughout the document, these topics will appear. A system will be more attractive if the number of matching that can be done in a second is high, and the error rates, low. Biometric error rates are defined in Section II.2.

I.2 Physical Tokens: What we Own Identifies us

Using an artificial security element is also a common practice. For example, wax seals used to authenticate a document as coming from an authority; the signet ring itself was an element of enough sophistication to identify his owner. More recently, these devices became less elegant, but also less reproducible.

In this document, we shall often refer to nomad devices that can be used in several protocols. This short section describes the context in which they can be used.

Smartcards, chips on plastic

A smartcard is a device made of a processing chip, small enough to be embedded into a plastic card. Examples of smartcards can be found in SIM cards (Subscriber Identity Module) used in mobile phones, but also on many credit cards.

Smartcards have actually the same architecture as any processing unit: they have a central processing unit, a memory, input and output pins, and, depending on the card functionality, specific processors. Smartcards can be contact devices (to be used *within* a terminal) or contactless; in the former case, the electric power comes from a terminal; in the latter, electricity is produced from electromagnetic induction. It is also possible to have an integrated battery.

While the general architecture of a smartcard is of little interest to the rest of this document, we however retain some key facts:

- while a smartcard is a classical computer, because of its limited size, it obeys to much stricter memory constraints;
- it is much harder to get the information that is used in a computation inside a smartcard than inside a desktop computer;
- cryptographic smartcards are equipped with a cryptoprocessor that ensures a minimal level of tamper resistance.

Biometric Smartcards A recent use of smartcards appeared with biometrics. Biometric systems are often reproached the presence of a central database that contains all the biometric templates of the users. **Match-on-Card** is a decentralized model of biometric comparison that implies working with a smartcard, a smartcard reader and a biometric sensor.

Whenever a user wishes to have access to a service, he presents his smartcard and his biometric feature to the sensor. The sensor extracts the biometric template out of the feature, communicates it to the smartcard through the smartcard reader, and the smartcard compares it to the referenced template.

The comparison is made directly in the embedded processor, and the original template is read from the smartcard memory. This removes the need for an external database, and also minimizes the risk of knowing who has access to the system. The main drawback of this method is the need for everyone to carry their reference-template on a physical token.

The use of a smartcard also enables to use off-the-shelf protocols for mutual authentication between the smartcard and the smartcard reader; in which case, the owner of the smartcard can be reassured on the security of the biometric terminal, and the system can trust the smartcard and the computations made in it.

Radio-frequency Identification

Another kind of (embedded) physical element interesting in our work is Radio-Frequency Identification (**RFID**) tags. It is also an embedded device, which function is mainly to communicate with other wireless elements.

As for the smartcard, RFID tags are micro-computers equipped with a CPU, some memory, and sometimes, dedicated processors. Their characteristic is to communicate using Radio-frequencies, and are therefore equipped with an antenna. RFID tags cover a wide range of applications, from very low-cost identification tags designed to replace barcodes, to more complex elements like microchips capable of sophisticated cryptographic operations in modern electronic passports.

RFID tags present new challenges to cryptographers, as they are activated by electromagnetic induction. It is therefore possible for anyone in the neighbourhood to communicate with an element. Moreover, as in most cases the components are not sophisticated enough to implement high level communication protocols such as TLS/SSL, the communication protocols with these devices is still an open subject.

I.3 Passwords: Identification Based on a Knowledge

Finally, the most classical identification element, still universally used, is the password. Its advantages are numerous: it can be chosen freely by the user, can be replaced at will, requires nothing but some (brain) memory to carry securely, and can be used with hardly other hardware than a keyboard.

The classical use of passwords is illustrated in Figure I.2. The principle is that each user has a login (usually his name, or some pseudonym) associated to a password, known by himself only. Authentication consists in sending the login and a proof of knowledge of his password (for example, a cryptographic hash of the password concatenated with a random binary string, or the encryption of a binary string using the password as the key, along with some salting).

However, it is obviously still the weakest and more sensitive point of entrance of all security applications. Indeed, a password is easy to remember only if it is easy for the user to find it again; in other words, making it easy for the user also makes it easy for an attacker. This can be quantized with the classical notion of *entropy*, which provides a measure of the variety of the password that are used, or, in an equivalent way, a measure of the hardness for the adversary to break into a password-protected system. The formal definition of entropy follows.

Definition 1. Let X be a random variable over the finite set $\mathcal{X}$, with probability law p_X. With these notations:

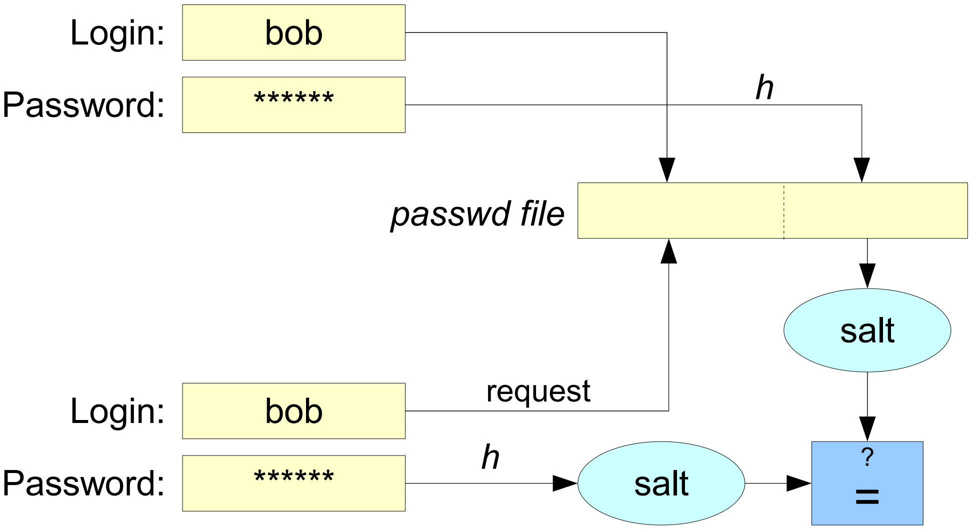


Figure I.2: General Scheme for Classical Password Authentication

- The information associated to the event $X = x \in \mathcal{X}$ is

$$I(x) = -\log p_X(x);$$

- The entropy of the source X is the average information associated to X, in other words:

$$H(X) = \mathbb{E}(I(X)) = \sum_{x \in \mathcal{X}} -p_X(x) \log p_X(x)$$

The main problem of passwords is their low entropy. Indeed, many password systems are broken with a dictionary attack. Even worse; many systems ask their users to choose their password, but these systems do not put any security measure on the password database, which is stored in clear text in the file system. And, as a lot of systems require passwords, few people derive different passwords for each system.

The result is that low-entropy password is one of the least secure ways to perform authentication or identification, without using any additional factor. That is why even though there exist secure password protocols, password-based authentication is still the weakest link in penetration tests.

I.4 Identification and Authentication

To conclude this introductory chapter, we state the difference between authentication and identification. We actually distinguish Authentication, Identification and Authorization.

Authentication consists in proving one's identity by providing an identifying element. It is the classical login/password paradigm, where the knowledge of the password validates the claimed identity (the login).

Identification is much easier for a user, as he just has to present a single element for the service to find his identity. Actually, the process of identifying consists in looking into a database for the entry that is the most likely to correspond to a queried element. The system then returns the identifier of the person in the system.

Authorization is a procedure that is looser than identification. It is used when there is no need to find a user's identity, only to know that the owner of the data belongs to the group of all authorized people.

Of these three kinds of methods, the most difficult to achieve is Identification, as it is the one that requires in the same time to authorize a user, and to provide his identity. Part 3 describes ways to achieve secure biometric identification, and part 4 focuses on wireless devices.

Chapter II

Biometric Templates

> Man is still the most
> extraordinary computer of all.
>
> John F. Kennedy

II.1 A Model for Biometrics

As it is mentioned earlier in chapter I, biometrics is a very powerful means of recognizing people, using some characteristics that are specific to each individual, but found in everyone. The study of these characteristics leads to methods and algorithms that compare biometrics measurements, and output a matching score. Unlike the intuitive recognition of person, these methods can be implemented into machine-readable code; the main advantage is that it is now possible to perform recognition of people automatically.

The basic tools to perform biometric identification are:

A sensor A device that captures a physical element and transcribes it into a binary representation. It often produces a picture or a sound recording.

An encoder An algorithm that takes as input the measurement of the sensor, and outputs a template, *i.e.* a data format assembling the discriminative information used in biometric comparison.

A matcher An algorithm that takes two biometric template and outputs a matching score. As we will see later, this matching score can either be a resemblance score (the bigger, the more likely the two templates come from the same physical element) or a distance-type score.

A database An element that stores the templates of *enrolled* people, those who can be identified by the system.

A service provider This is the component that just assembles the different pieces to perform identification.

Let us now review the state of the art for fingerprints and irises.

The fingerprint

Representations of Fingerprints

The first representation of a fingerprint is - of course - its picture as captured by the sensor. However, the best results are seldom obtained by processing image-based matching functions, and the first classical step for automated fingerprint recognition is to perform some signal processing operation. More details on how to transform a fingerprint picture into an interesting format can be found in [117], and we will here only retain the final result.

The Minutiae Set A minutiae is characterised by three (sometimes more) important features:

- The two coordinates (x, y), which represent the position of the minutiae in a coordinate system.
- The orientation θ, which represents the direction of the ridge flow at the location of the fingerprint, with respect to a fixed axis.

Most often, the origin (and axis) of the system varies from one capture to another. Indeed, as it is difficult to "absolutely" align a fingerprint, *i.e.* translate and rotate the picture so that two different measurements of the same fingerprint could be almost superimposed, the position (x, y) and the orientation θ will not precisely match for each fingerprint.

Sometimes, it is possible to add to the representation, for each minutiae:

- Its type: is it a ridge ending, a ridge bifurcation, ...
- The quality of the estimation: how likely is it that there really is a minutiae at this position?
- The quality of each parameter measured: a standard deviation for the position if it was not easy to locate the minutiae, a standard deviation for the orientation for the same reason, or a probability table for the type.

Depending on the state of the art and the types of information used in a matching algorithm, more or less information will be noted in the template. Even with more details, this still makes this representation very light, as there are rarely more than some hundreds minutiae in a fingerprint.

The Ridge Flow Matrix (RFM) This representation is, for a set of positions in the fingerprint, the orientation of the ridge flow at each point. It is represented as a matrix, where each component is the angle of the local ridge.

It is measured by applying directional- and frequency-filters, such as Gabor filters[1] to the picture at each point; the maximal response to the filters gives the local orientation.

This representation has the major advantage of being of fixed size. The issue is that while the RFM is pretty discriminative, it is not discriminative enough to be used as the only matching element.

Binary Sparse Representation Traditional fingerprint matching is made thanks to minutiae extraction [117] and comparisons of unordered sets $\mathcal{E}$, $\mathcal{E}'$ of variable length. Using the characteristic function $\chi_{\mathcal{E}}$ – as done in [60, 64] – is a way to translate minutiae into a binary vector of fixed length. The size corresponds to the number of values the coordinates could take. From a set of minutiae, the idea is to construct a vector with all coordinates equal zero except those which are associated with the position of one minutiae.

As we will see later on, the problem is that this representation is not well-suited for binary secure sketches. Indeed, the metrics associated to the set representation is the symmetric set difference, which does not take into account local distortion due to elasticity of the finger skin.

An attempt to model the distortions of the fingerprints was however made in [172, 171], using this representation and a graphical model for the translations, removal and insertions of minutiae from one capture to the other. This research direction was not successful enough to inspire more practical constructions.

Binary Quantization by Selecting Components We here present a representation of fingerprints inspired by [176], in the line of the previous works [112, 177]. This representation, published in [31], differs from the original as it is designed in the spirit of coding theory, allowing errors *and* erasures. It has common roots with Fingercode-type algorithms, and consists in three phase. First, compute a set of real values of fixed size from the fingerprint. Then, process all this data on a training database, in order to select which components are reliable and which one are just noisy. Finally, for a given fingerprint, compare each component with the threshold, and output the corresponding bit array.

The main idea is to deal with fingerprint patterns rather than minutiae. It makes use of core-based alignment techniques and pattern features linked to

[1] The Gabor Transform is an extension of the Fourier Transform, dedicated to Time-Frequency analysis. The principle is to pre-filter the signal by a Gaussian window centred around a space position, then compute the frequencies that contribute to the signal in this window.

directional fields, thanks to the techniques described in [11, 10, 9]. Moreover, to increase the stability of the vectors, the binary fixed-length strings are generated following some statistics by using several images per user at the enrolment. This method is detailed in the following paragraphs.

First, to compute a n-real vector v out of a fingerprint picture I (with $n = 5n'$):

Algorithm 1 Generate a fixed-length vector v out of a picture I

```
• Select $n'$ fixed positions in the picture frame;
• Realign the fingerprint with respect to an absolute point (see section II.1).
• For each of the position $j$ out of the $n'$ present in the picture:
    – For $k \in [\![0, 3]\!]$, $v^{(k \cdot n' + j)}$ is the response of a Gabor Filter around the position $j$, of orientation $\theta = k\frac{\pi}{4}$;
    – $v^{(4n'+j)}$ is the value of the fingerprint Ridge Flow Matrix at the position $j$.
```

Remark 3. Due to the realignment algorithm, there are some components that will not have any value in the array v. This is not a perfect situation, but it is still manageable to use the vectors v; the problem of array cells that are empty has been addressed by the IrisCode technology (see also section II.1); such a cell position is called an *erasure*, and we note $v^{(k)} = \star$[2].

In the matching process, it is however reasonable to assume that not too many erasures happen, or the matching score becomes irrelevant.

The second step consists in computing elementary statistics on the vectors, in order to select reliable components. We then suppose that there are ${}_jN$ users, who provide m fingerprint pictures each I_i^j ($i \in [\![1, N]\!], j \in [\![1, m]\!]$). For such i, j, I_i^j is quantized into a vector v_i^j following algorithm 1 we note $\mu^{(k)}$ the average value of the array coordinate $k \in [\![1, n]\!]$ and $\mu_i^{(k)}$ the average value of the array coordinate *k for the user i*. The same goes for the standard deviations, noted $\sigma^{(k)}$ and $\sigma_i^{(k)}$.

[2] In the literature, an erasure is sometimes noted ϵ; we avoid this notation as ϵ is a widely-used character.

$$
\begin{aligned}
\mu_i^{(k)} &= \frac{\sum_{j:(v_i^j)^{(k)}\neq\star}(v_i^j)^{(k)}}{|\{j:(v_i^j)^{(k)}\neq\star\}|}\\
\mu^{(k)} &= \frac{1}{N}\sum_{i=1}^{N}\mu_i^{(k)}\\
(\sigma_i^{(k)})^2 &= \frac{\sum_{j:(v_i^j)^{(k)}\neq\star}\left((v_i^j)^{(k)}-\mu_i^{(k)}\right)^2}{|j:\{(v_i^j)^{(k)}\neq\star\}|}\\
(\sigma^{(k)})^2 &= \frac{\sum_{i,j:(v_i^j)^{(k)}\neq\star}\left((v_i^j)^{(k)}-\mu^{(k)}\right)^2}{|\{j:(v_i^j)^{(k)}\neq\star\}|}
\end{aligned}
$$

Remark 4. It is possible to select the reliable components by picking, at this step, the coordinates that give the best Signal-to-Noise ratio, as we shall do later; however, a component that is reliable with respect to this measure is not necessary stable after quantization. Therefore, the Signal-to-Noise ratio will be used as a secondary comparative measurement to select the most stable components.

The third step is the quantization into bits. Given a vector v, and the mean vector μ, it becomes easy to obtain a binary vector $x = \left(x^{(1)},\ldots,x^{(n)}\right)$, defined by:

$$
\begin{aligned}
x^{(k)} &= 1 \quad \text{if } v^{(k)} \geq \mu^{(k)}\\
&= 0 \quad \text{if } v^{(k)} < \mu^{(k)}
\end{aligned}
$$

The fourth step is the selection of reliable components. For this, the basic idea is to look for components that will be the least noisy, and the most discriminative.

A component is noisy if its value is not always the same. In other words, when its intra-class entropy is high. The first goal of the component selection is to pick components with the lowest intra-class entropy; it needs only selecting first those that always take the same value; then those for which there was one measurement that produced a different bit, and so on.

A component is discriminative if its value has a high entropy over the whole population: indeed, if we collect ℓ independent bits of high entropy (the entropy of each bit being about 1), the entropy of the whole vector is the sum of the ℓ separate entropies, and is near to ℓ. However, in practice, the components are *not* independent and the entropy of each bit is not that easy to evaluate. We then approximate the Signal-to-Noise Ratio with the estimate: $SNR_j^{(k)} = \frac{\sigma^{(k)}}{\sigma_j^{(k)}}$.

This provides Algorithm 2, that select the n_1 most reliable components with these approximations.

Algorithm 2 Selection of the most reliable components for the user i

```
On input: $v_i^1, \dots, v_i^m, \mu, \sigma$; $n_1$ the size of the final binary vector.
On output: A subset $W \subset [\![1, n]\!]$ of size $|W| = n_1$.
  • For each $v_i^j$, compute the binary vector $x_i^j$;
  • For each $k \in [\![1, n]\!]$, compute $\sigma_i^{(k)}$
  • Define the vector $y_i$ such that $y_i^{(k)} = \left( |\sum_{j=1}^{m} (x_i^j)^{(k)} - \frac{m}{2}|, \frac{\sigma_i^{(k)}}{\sigma^{(k)}} \right)$;
  • Sort $y_i$ by the first coordinate in decreasing order, then by the second coordinate by increasing order;
  • Return the $n_1$ first coordinates of this reordering.
```

Remark 5. The entropy of a single-dimension Gaussian source $\mathcal{S}$ of mean μ and standard deviation σ is well known, and is $h(\mathcal{S}) = \ln\left(\sigma\sqrt{2\pi e}\right)$. Finding the minimal ratio $\frac{\sigma_i}{\sigma}$ is a way of finding the component of minimal entropy with respect to the whole population, *i.e.* a component that has a low entropy for a user j and for which the inter-class entropy is high.

Remark 6. [100] suggests another way of selecting the components by using a Signal-to-Noise ratio defined as $SNR_j^{(k)} = \frac{1}{2}\left(1 + \log\left(\frac{|\mu_j^{(k)} - \mu^{(k)}|}{\sqrt{2}\sigma_j^{(k)}}\right)\right)$. This normalizes the selection of the components, taking into account not only the standard deviation of the values, but also the gap between the intra-class and extra-class behaviours.

At the end of this procedure, we have, for each user, a binary vector x_i of length n_1, and an index W_i of the components to be selected for a future quantization. It is then possible to reproduce a fresh vector x_i' of length n_1 from a new image I' by applying Algorithm 1, then quantizing by the threshold μ each component of W_i.

However, this implies that for each user, the index W_i must be stored at an interesting place; as W_i is likely to leak information on I, special care must be done for this.

In practice, on the database FVC 2000 (Db. 2), this algorithm enables to extract a 2048-bit long vectors at the verification step, along with a mask of the same size, among which there are 1984 bits of information. The mask tells when a coordinate is a $\star$ or not. This algorithm requires a multiple enrolment (which means that a person needs to provide multiple fingerprints to be enrolled); in our experience, we tried 6 fingerprint for the enrolment on this database, resulting in a 2048-bit long enrolled vector. The performances of this quantization algorithm along with the distance-based matching function, are represented in Section III.5.

Algorithm 3 Binary quantization of a fingerprint

```
On input: Fingerprint picture $I$; length $n_1 \in \mathbb{N}$, positions $W_i$, mean vector $\mu$
On output: A binary template $x \in \{0,1\}^{n_1}$.
  1. Compute the real vector $v \in \mathbb{R}^n$ following Alg. 1
  2. For each $k \in [\![1, n_1]\!]$, compute $x^{(k)}$ by:
      • $x^{(k)} = 0$ if $v^{(W[k])} < \mu^{(W[k])}$
      • $x^{(k)} = 1$ if $v^{(W[k])} \geq \mu^{(W[k])}$
      • $x^{(k)} = \star$ if $v^{(W[k])} = \star$.
```

Fingerprint alignment

A usual problem in fingerprint comparison is the alignment of two templates. As it often happens that a person presses his finger on different positions of the sensor, the fingerprint is likely to be translated, and it is probable that the overlapping of two measurements is not perfect.

This raises the problem of fingerprint realignment, which is not an easy one. If the two fingerprints are available for the matching, then it is possible to realign the pictures with many algorithms; in our case, we shall face situations where only one fingerprint is available, and the other is protected and cannot be recovered for the alignment.

A possible way to deal with this issue is to realign the picture with respect to a predetermined point. The classical idea is to locate the fingerprint core, or its centre if the fingerprint does not have a core, and to impose the central position for this point. It solves many cases, but it remains difficult to precisely locate the core of a fingerprint, and sometimes, the measurement is so little that this method is no longer possible.

Recalling that this problem is not perfectly solved yet, we still made algorithms that use an “absolute alignment” prior to the encoding.

The Iris, and the IrisCode

The first study and use of the iris as a biometric feature were made by John Daugman [55],inventor of the IrisCode. It is a binary representation of the iris, that makes the comparison of two iris a fast and easy operation.

A picture of the iris is taken with a near-infrared camera; the boundaries of the pupil and the iris are detected, and the picture is then represented in polar coordinates. This gives a band-like picture (see Fig. refIris), on which 2D filters can be applied. Daugman suggests the use of Gabor filters, whose phase is then quantized, depending on its sign. The picture is normalized onto its polar representation, then divided into areas of regular size. The amplitude information is discarded and the actual bits are the phase quantization of this Gabor-domain representation of the iris image. The ordering of the bits is directly linked to the localization of the area. In practice, the iris code can be represented by an 2D bits-array.

The result is a 2048-bit vector; some of the components of this vector may be masked, because of eye occlusions, reflections, or difficulties to accurately separate the iris from the rest of the picture. In other words, the IrisCode is made of two 2048-bit vectors I and M; I contains the information bits, and $M[t]$ says whether the bit $I[t]$ is relevant or not.

The matching is made on two IrisCodes by computing a Hamming-like distance:

$$d_I\left((I_1, M_1), (I_2, M_2)\right) = \frac{w((I_1 \oplus I_2) \odot M_1 \odot M_2)}{w(M_1 \odot M_2)} \tag{II.1.1}$$

where $w(x)$ is the Hamming weight of x, $\oplus$ is the bit-wise exclusive-or operator and $\odot$ is the bit-wise product ("and") operator.

This distance is approximately equal to $\frac{1}{2}$ when the two IrisCodes come from different eyes, and is significantly lower when the originating iris is the same. Indeed, comparing IrisCodes is just a test of independence of two measurements.

Remark 7. In practice, the distance used to compare Irises is not d_I but a derivative normalized distance

$$d_n = 0.5 - (0.5 - d_I)\sqrt{\frac{n}{911}}, \tag{II.1.2}$$

where $n = ||M_1 \cap M_2||$ is the number of bits effectively compared, and 911 is the average number of bits compared. This formula is used to rescale the scores, to have them balanced around 0.5; using d_n instead of d_I significantly improves the results; however, d_I still provides a good matcher that can be used in security applications.

Note that the iris template as computed by this algorithm has a specific structure: [56] reports 249 degrees-of-freedom within the 2048 bits composing the template. This observation was obtained by comparing the non-matching score distributions to a normalized binomial distribution with 249 trials.

Other biometrics

Finally, a commonly found biometric representation is called the *eigenface*. Used for face recognition, it is a geometric approach that uses a fixed set of images (the eigenvectors E_i of some $N \times N$ matrix) as reference points in a multidimensional space. The representation of a fresh image I is simply the collection of the projections of I over each of the eigenfaces E_i. In other words, the representation of I is the vector $v \in \mathbb{R}^N$ of coordinates $\langle E_i | I \rangle$.

As the approach is geometric, the matching of two templates v_1 and v_2 is simply done by computing the Euclidean distance of v_1 and v_2, and comparing it to a threshold.

The elegance and simplicity of this construction makes it a widely studied representation of biometrics; it has also been applied to other kind of biometrics, such as palm, hand, footprint, and iris recognition.

In the literature, a wide ecosystem of biometric modalities and matching algorithm has been existing for several years; an algorithm emerges from time to time above the others, and provide an alternative way of dealing with biometrics. However, due to our concerns on template protection, we will essentially concentrate on the representations of fingerprints in a Hamming space, *i.e.* $B = \{0,1\}^N$ with the Hamming distance as the base element for a matching operation. For this reason, we shall concentrate our efforts on the fingerprint and the iris.

II.2 Some Limits to the Use of Biometrics

Even though cinematography and literature have been full of reference to biometric systems for the last 20 years [1, 129, 159, 168], we still do not live in a world where police cars sweep the street with a camera that automatically recognize people. This is because there are several limitations that need to be taken into account when we deal with biometrics.

Privacy Issues Arising from Biometrics

Biometric features are elements that do identify people with little forensic traces, which raises many problems linked to the end-user's privacy.

Personal Information, Computers and Privacy First of all, the fact that personal elements, even not biometric ones, are stored in a database inspires a lot of concern. Indeed, there is a great number of selling companies, websites, mail systems, that require the user to provide their personal details in order to get registered. Even though a growing proportion of these systems have a privacy policy quite reassuring for the final user, the insurance that your private information is secure is pretty thin. Indeed, the fine prints of the *End-User Agreement* are almost always more protective of the company than of the so-called End-User.

A direct threat to such information is its misuse by the system administrator, to fulfil the company design or his own. Databases that are constructed within years are worth a lot of money, and human liability cannot be ruled out. Databases are now easily transferable, and no longer require a cumbersome medium. Therefore, the possibility that the information is passed from a database to another exists, this can also be the case when different companies merge to make a larger one.

Another example is the voluntary communication of “anonymous data” from one system to the other as an agreement in order to improve their systems. This happened in the case of the TomTom IQ Routes$^{\text{TM}}$system: since 2008, some cell-phone companies send to TomTom the density of users in each cell, so that the GPS systems can deduce which roads are jammed. This should not be a privacy issue to users, but in the same time, it shows how the data gathered is invaluable.

Another threat is the e-mail communication; in most cases, e-mail is not secured, and the information that transits from the POP server (Post Office Protocol) to the client is generally in cleartext.

Finally, even international companies and associations are subject to the law of their hosting country, which is a parameter that is not easy to take into account. That means that at any time, the private information stored in a system can be demanded by the authorities[3].

At the end of the day, when a systems administrates a database in which sensitive information is stored in a way that the user cannot control, then there is a potential privacy threat to these data.

Biometrics, Sensible Information The case for biometrics is quite different from that of more traditional information. Indeed, biometrics are personal information - as they are believed to be unique - *and they are not renewable.* Since they are used as identifying elements, the way they are stored is critical to their deployment. Indeed, while there are clever methods of storing classical authenticating data as passwords, the 2007 state-of-the-art was pretty thin on storing biometric data while preventing its use by the administrator without the user’s agreement.

It is worth noting that biometrics also have the annoying characteristic (depending on the point of view, of course) of leaving traces. The historical use of fingerprints is for police forensics, in order to help solving crimes. In the same way, they search for DNA which they can use in criminal investigations. While the people “without fear and without reproach” are not likely to be falsely incriminated, it remains that with only traces of data, the imprecision of biometrics can lead to misidentification [122] (see also section II.2 for more details on error rates).

Biometrics can also leak information that go beyond the usual “name and surname”. DNA can (obviously) reveal genetic disorders, and information not only on the user, but also on his family. Face recognition is efficient, but discriminatory with respect to the color of the skin (and pictures stored reveal that information too). It is even possible, from a picture of the eye, to spot traces of diseases.

All these reasons make biometrics a sensitive target, that require a special protection system.

[3] That is, depending on the hosting country’s law.

A non-vanishing Error Rate

Biometric systems are subject to many kind of failures, at each step of the verification / authentication / identification process. The most commonly known - and the ones of interest to us - are the False Rejects and the False Accepts.

To define these notions, it is preferable to provide a formal background to biometrics and matching functions. This will be used throughout the rest of the manuscript.

Users, templates and matching functions When considering a biometric system, we need to determine a set of N "legitimate" users $\{\mathcal{U}_1,\mathcal{U}_2,\dots\mathcal{U}_N\}$ that are likely to use the system. In addition to these N people, there are N' "illegitimate" users $\{\mathcal{U}'_1, U'_2,\dots,\mathcal{U}'_{N'}\}$ that could use the biometric sensor without being enrolled.

A user $\mathcal{U}$ is a source for biometric templates. In other words, each measurement of a user's biometrics provides a template b; this operation will be noted $b \leftarrow \mathcal{U}$ (as it is sometimes written in the literature). The set of all biometric templates will be noted $\mathcal{B}$, and $\mathcal{U}$ can be viewed as a random variable over $\mathcal{B}$, and b is a realisation of $\mathcal{U}$. In this model, the random variable has a distribution law that is centred around a mean value, and the realisation b is equal to this value, up to some noise.

A first cause of biometric failure is the impossibility for the system to obtain a biometric template from a user. This can happen for many reasons (for example, a fingerprint is too dry, or someone's eye is hidden by glasses...). This is referenced as a Failure to Acquire error.

Biometrics use matching functions that output *true* if the two templates are similar enough, and *false* otherwise. Failure to execute the matching algorithm outputs the $\perp$ symbol.

Definition 2. Let $\mathcal{B}$ be the set of all biometric templates; a matching function over $\mathcal{B}$ is a function $\mathrm{m} : \mathcal{B} \times \mathcal{B} \to \{0, 1, \perp\}$.

Two templates $b, b' \in \mathcal{B}$ such that $\mathrm{m}(b, b') = 1$ are called *matching* templates. They are *non-matching* otherwise.

False Accepts, False Rejects A *False Accept* happens when two templates coming from different users are matching. This is bad when the biometric system is deployed in order to enhance access control; in practice, raising the matching threshold can reduce the number of False Accepts.

A *False Reject* happens when two templates coming from the same user are not matching. This provides discomfort and frustration to the user, as he needs to process his biometric trait once again, or to provide another element to prove his identity.

False Accepts and False Rejects happen statistically over a set of users. In practice, it is possible to measure the False Accept Rate (FAR) as the ratio of False Accepts to the number of impostor matchings, and, similarly, to measure the False Reject Rate (FRR).

If the threshold is increased, then the FRR increases, and the FAR decreases. That is why biometric systems are characterised by the Detection Error Trade-off (DET), a curve that plots the FRR with respect to the FAR. An equivalent representation is the Receiver Operating Characteristic (ROC) curve, in which the FAR and FRR are plotted as function of a score threshold. It is worth noting that up to now, *no biometric system is error-proof.* Depending on the context, an application will encourage either False Rejects or False Accepts. To significantly decrease error rates, a solution would be to use multiple biometrics to identify one person. This is called multi-biometrics, or biometric *fusion* [111]. From a theoretical point of view, there is no significant difference between one biometric trait and multi-biometrics; however, in practice, as the matching functions are not necessary the same for all biometric traits, this induces problems when integrating biometrics with cryptography, as we will see later on.

A Psychological Unease

Last but not least, there is another barrier to a wider use of biometrics. Indeed, there are technical and technological limitations on error rates and performances that will most likely be reduced in the next years; the psychological context is however an aspect where sheer science can merely intervene.

Since a huge historical literature weighs over the use of fingerprints, their deployment does evoke police and forensics applications in the collective mind. This is reinforced by the growing reputation of DNA analysis and facial recognition software that are executed on closed-circuit television feeds. All these elements provide - rightly - reasons of concern to the final user of biometrics. Indeed, the question "*what will be the uses of my biometric trait?*" is raised as soon as a user is asked to enrol into a system.

In this perspective, clarity and transparency might be the safest way to deal with reluctant users; a sound scientific reputation can be the triggering element for the public trust - as it was for Internet shopping [59].

II.3 Authentication and Identification

We here revisit section I.4 in order to outline what makes biometric identification a sensitive topic.

Different Modes for Different Levels of Comfort

The most secure way to use biometrics is also the least convenient. Biometric *Authentication* requires a user to state his identity, then to provide a biometric measurement. The comparison of the new biometric template with one that is specified in the prover's reference data is an additional guarantee of his identity. That means the person to be authenticated needs to tell his name or record number, or to show an element that contains these information (that includes the Match-on-Card paradigm presented in Chapter I.2).

In biometric *Identification*, the user only needs to have his biometric trait captured to be identified; the same goes for *Authorization*.

These methods are ordered from the least convenient to the least secure. Indeed, it is much easier for an intruder to be identified, and even more, to be authorized, into a system where there are multiple people enrolled. For the least, the chances of entering such a system are multiplied by n, the number of entries in the database.

Even though it is not possible to easily derive the error rates for an identification system from the associated authentication system, the following section provides elements to understand the behaviour.

Identification Error Rates

Let us formalize the system settings. n users are enrolled in a system, and their template b_i are stored in a central server. The goal for an identification procedure, is, on input b', to output the ι such that b' and b_ι come from the same user, and $\perp$ if no such template exists.

The easiest way to do this is to compute the matching score $s_i = \mathrm{m}(b', b_i)$ and to output the ι that maximizes the s_i, if $\max\{s_i\} \geq \tau$ where τ is a confidence threshold.

Now, we model the score distribution as follows: if b and b' are matching (resp. non-matching), then $\mathrm{m}(b', b)$ follows the probability distribution function f (resp. g), with cumulative distribution $F(x) = \int_{-\infty}^{x} f(t)dt$ (resp $G(x)$), as illustrated by Figure II.1. We also suppose the matching of templates coming from different users gives independent results.

Assume that there is indeed a matching element b_ι in the database. With these elements, the result of the identification is correct if $s_\iota \geq \tau$ and $\forall i \neq \iota, s_\iota > s_i$. The probability p of misidentification is then:

$$\begin{aligned} p &= 1 - \Pr\left[s_\iota \geq \tau \text{ and } \forall i \neq \iota, s_\iota > s_i\right] \\ &= 1 - \Pr\left[s_\iota \geq \tau\right] \cdot \prod_{i \neq \iota} \Pr\left[s_\iota > s_i\right] \\ &= 1 - (1 - F(\tau)) \cdot \left(\Pr\left[s_\iota > s_i\right]\right)^{n-1} \end{aligned}$$

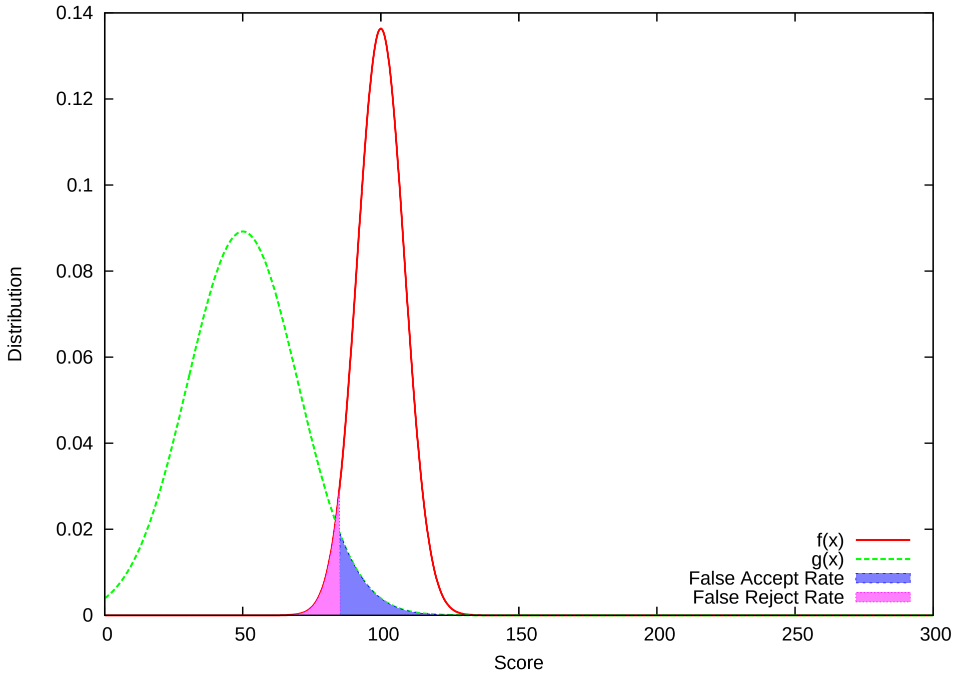


Figure II.1: Illustration of the score distributions for matching and non-matching pairs. The error rates (here $FAR = FRR = 4\%$) are the filled areas.

The probability of a genuine record having a better score than another template can be written as:

$$\begin{aligned} p_1 = \Pr\left[s_\iota > s_i\right] &= \int_{\mathbb{R}} f(s) \Pr\left[s > s_i\right] ds \\ &= \int_{\mathbb{R}} f(s) G(s) ds \\ \text{thus} & \\ G(\tau)(1 - F(\tau)) \quad \leq p_1 \leq & \quad 1 - F(\tau)(1 - G(\tau)) \end{aligned}$$

This gives an estimation of p (under the hypothesis that the scores are random and independent):

$$1 - (1 - F(\tau)) \left(1 - F(\tau)(1 - G(\tau))\right)^n \leq p \leq 1 - G(\tau) \cdot (1 - F(\tau))^n$$

This shows indeed that, even when f and g are well separated (which leads to small FAR and FRR), the probability of misidentification remains considerable when the system contains many users. The minimal and maximal misidentification probability are drawn in Figure II.2.

Remark 8. It is interesting to notice that there is an exponential gap between identification and authentication. This can be manifested here in terms of

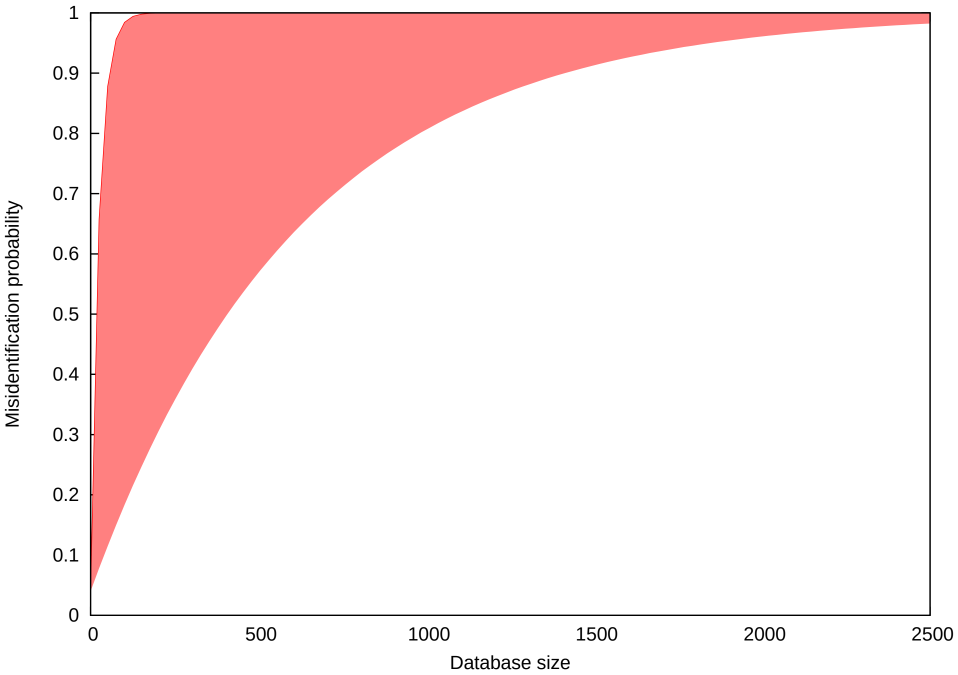


Figure II.2: Minimal and maximal misidentification probability for a 4% EER

error rates, for fixe-size data; we shall see in Chapter IX that a similar exponential gap exists, this time in term of size, with constant error probability.

Part 2

Protecting Biometric Templates

Chapter III

The State-of-the-Art

> Books serve to show a man that those original thoughts of his aren't very new at all.
>
> Abraham Lincoln

Looking at biometrics using a channel model suggests the use of all the range of information- and coding-theoretic tools developed over the last decades. At enrolment and verification, a user provides a biometric template; from this, we deduce that the database and the client share a secret (the biometric template) that went through a channel. How to reconcile this secret without publishing it was investigated in [61].

Since Davida *et al.* wrote [57], *Secure Sketches* are still the most studied way to protect biometric templates. However, there are two other methods in vogue, namely *Cancelable Biometrics* and *Fuzzy Extractors*. This chapter is dedicated to define these three models, how they work, and what are their weaknesses.

The different constructions that follow have unequal security models and assumptions; the aim of presenting the State-of-the-Art is essentially to provide an understanding of the different approaches used in biometrics, and as such, the following sections are not as detailed as they could be.

III.1 Secure Sketches

The idea that leads to Secure Sketches is to get rid of the noise that is inherent to biometrics, using Error Correction. In the meanwhile, the format of the data to be stored is modified in such a way that a significant amount of entropy is now hidden. The definition for "min-entropy", along with notions of information theory, is provided in Appendix A.

Definition 3 (Secure Sketches, [60])**.** Let $\mathcal{M}$ be a metric space with associated distance function d, S an image space. A $(\mathcal{M}, m, m', t)$-*Secure Sketch* is a randomized map $\mathrm{Sk} : \mathcal{M} \to S$ with the following properties:

1. *Recovery*: there exists a recovery function $\mathrm{Rec} : S \times \mathcal{M} \to \mathcal{M}$ that enables recovery of $m \in \mathcal{M}$ given its sketch and a m' near to m, *i.e.* such that

$$\forall m, m' \in \mathcal{M} : d(m, m') < t \Rightarrow \mathrm{Rec}(m', \mathrm{Sk}(m)) = m;$$

2. *Entropy Retention*: If W is a random variable over $\mathcal{M}$ with min-entropy at least m, then the average min-entropy of W given $\mathrm{Sk}(W)$ is greater than m', *i.e.*

$$H_\infty(W) \geq m \Rightarrow \overline{H}_\infty(W|\mathrm{Sk}(W)) \geq m'.$$

Applying Secure Sketches to biometrics is pretty straightforward. The main difficulty is to make sure that the templates belong to a metric space.

Instead of storing the biometric template b, its sketch $\mathrm{Sk}(b)$ is stored in the database. At the authentication step, the fresh template b' and the sketch $\mathrm{Sk}(b)$ are used together in the recovery function to reconstruct b. If we are not certain - as is often the case with biometrics - that the new template b' is at distance at most t from b, then the only step left to do is to check that the reconstructed b is indeed the correct one. This can be done by storing a (cryptographic) hash $h(b)$ of the original template together with the sketch $\mathrm{Sk}(b)$.

The Code-Offset Construction

There exists several propositions for Secure Sketches, but the most commonly known is Juels and Wattenberg's Code-Offset construction [96], also known as the Fuzzy Commitment Scheme. This construction uses error-correcting codes for sketching and reconstructing.

Initialization We suppose that $\mathcal{M}$ is the Hamming space $\{0, 1\}^n$.

- Choose a $[n, k, d]$ linear error-correcting code C with decoding capacity at least t.
- Choose a hash function h.

Sketching To sketch, a codeword c is selected and added to the template b.

- Randomly select a codeword $c \in C$,
- The sketch Sk is equal to $b \oplus c$;
- The verification hash is $h(b)$.

Recovering Recovering the template from the sketch consists in decoding the sketch added with the new template.

- Compute $\tilde{c} = b' \oplus \mathrm{Sk}(b) = c \oplus (b \oplus b')$;
- Decode $\tilde{c}$ into the codeword c_1;
- If the decoding is successful, output $b_1 = c_1 \oplus \mathrm{Sk}(b)$. Otherwise, output $\perp$.

If b and b' are close enough (*i.e.* $d(b, b') \leq t$, then $b \oplus b'$ is a noise vector of Hamming weight at most t, and it is possible to reconstruct c out of $\tilde{c}$. This property ensures that the scheme has the "Reconstruct" property.

Remark 9. The verification hash enables to check that the decoding was successful, by testing whether $h(b) = h(c_1 \oplus \mathrm{Sk}(b))$. The hash function is useful to know that the recovery was done, but is not part of the Secure Sketch $(\mathrm{Sk}, \mathrm{Rec})$.

If the input b' is always to be close to b (at a distance less than t) then there is no need of h. However, if the contrary is possible, then the hash check enables to test the success of the operation.

Entropy loss

The entropy loss of this construction is easy to evaluate. Suppose that the biometric templates have minimal entropy m. Then the information on b leaked from $\mathrm{Sk}(b)$ is about $n - k$, *i.e.* the redundancy left by the code.

Proposition III.1. *The code-offset construction is a*

$$(\{0,1\}^n, m, m - (n-k), t)\text{ -Secure Sketch.}$$

The proof is available in [60].

Remark 10. The "entropy retention" of Secure Sketches ensures that there is still enough entropy to prevent *the reconstruction* of the biometric template. Nevertheless, if someone has access to the Secure Sketch, the "entropy loss" aspect can lead to privacy issue, especially if the entropy loss is significant.

The binary Singleton bound states that for a binary linear error-correcting code, $d \leq n - k + 1$. Moreover, for such a code, the theoretical decoding capacity is at most $\frac{d-1}{2}$. This means that it is not possible to always decode more than half the entropy loss; for a noisy source, in order for the code-offset construction to be efficient, the entropy loss will be large[1].

This is a first step for showing that Secure Sketches alone are not a viable solution for the protection of biometrics.

[1] Using codes that allow to decode beyond the $\frac{d-1}{2}$ threshold is a first way to improve the behaviour of Secure Sketches, that will be used in chapter IV.1

Secure Sketches and Syndrome Coding

A construction similar to the code-offset construction is the one based on syndrome coding. It was proposed by Davida *et al.* [57] to reconstruct the biometric q-ary template b using $n - k$ redundancy bits $H.b$ where H is a parity matrix. In this case, the information stored is the syndrome $H.b$ of the template b for the parity matrix.

This approach is the dual of the code-offset: instead of recovering a codeword, the goal is to use the redundant information to find a coset of the code, *i.e.* to find the translation from the code to the template b. Note that when the sketch stored is the sum $c \oplus b$, then applying the parity-matrix H of the code leaves only the syndrome $H.b$.

Here, the entropy loss is bounded by the syndrome length; it must be large enough so that, on input b' near to b, it is possible to reconstruct b from b' and $H.b$. From the point of view of information theory, there are q^k possible elements of $\mathbb{F}_q^n$ that have the same syndrome (actually, the translation of the code C by b), so the security is based on the difficulty to recover an element from its syndrome.

Initialization $\mathcal{M}$ is the Hamming space $\mathbb{F}_q^n$.

- Select a parity matrix $H \in \mathbb{F}_q^{n\times(n-k)}$;
- Initialize an algorithm sDecode taking as input $y \in \mathcal{M}$ and $s \in \mathbb{F}_q^{n-k}$ and outputs $x \in \mathcal{M}$ such that $H.x = s$ and $d(x, y) \leq t$.

Sketching To sketch b, compute and output $H.b$.

Recovering Recovering b from $H.b$ and b' consists in applying sDecode.

This approach was also used on fingerprints by [172, 171]; in this case, the underlying code is a LDPC, and the decoding is adapted to the fingerprint structure. Their work is based on a model of noise on the minutiae set. The authors reported good results on a - unfortunately - private database, but no independent team was able to reproduce and publish such experiments.

The Fuzzy Vault

Another well known and elegant Secure Sketch construction is that of Juels and Sudan: the Fuzzy Vault [95]. It is a construction that is dedicated to minutiae-based fingerprints, and assumes that minutiae can appear and disappear.

The principle of the Fuzzy Vault is to hide a Reed-Solomon codeword in some random noise. Based on the hardness of finding a polynomial of a given degree when there are more points than the degree, it uses a finite field $\mathbb{F}$ and a decoding function. Here are the Sketching and Recovery Functions:

Initialization A minutiae is given by two position and an angle, *i.e.* a triplet from $[\![1, L_x]\!] \times [\![1, L_y]\!] \times [\![1, L_\theta]\!]$ where L_x, L_y and L_θ are the number of quantifying steps of the positions and angle respectively[2].

- Select a finite field $\mathbb{F}$;
- Select an injective map $\sigma : [\![1, L_x]\!] \times [\![1, L_y]\!] \times [\![1, L_\theta]\!] \to \mathbb{F}$.

Sketching The element to be sketched is a minutiae-set $M = \{m_1, \dots, m_n\}$.

- Select a secret polynomial $P \in \mathbb{F}[X]$ of degree $k - 1$.
- Associate with each minutiae m_i the value $x_i = \sigma(m_i)$.
- Deduce the $y_i = P(x_i)$, and select chaff values y' for the coordinates x' that were not present in the set $\sigma(M)$. The sketch $\mathrm{Sk}(M)$ is the reunion of all the (x_i, y_i) together with the chaff (x', y').

Recovery The recovery is made on a minutiae-set $M' = \{m'_1, \dots, m'_{n'}\}$, of cardinal n'.

- Select all the $(x_i, y_i) \in \mathrm{Sk}(M)$ such that $x_i = \sigma(m'_i)$. This provides a n'-vector $((x_1, y_1) \dots, (x_{n'}, y_{n'}))$.
- Decode this vector with the Reed-Solomon decoder, to obtain a polynomial P'.
- Select all the coordinates of $\mathrm{Sk}(M)$ such that $y = P'(x)$, to recover a minutiae-set M''.

This scheme is efficient if the intersection of M and M' contains enough points, *i.e.* if $\frac{|M \cap M'|}{|M'|}$ is greater than the decoding proportion of the decoder.

Remark 11. The assumption that a minutiae's position are not modified from one shot to the other is very strong. This means that the Fuzzy Vault is not that well adapted to fingerprints. It implies that the position noise will be modelled as minutiae insertions and deletions, and so, the decoding capacity of the code needs to be high, once again, and the dimension of the code needs to be low...

Uludag and Jain [180] did obtain 128-bit keys on a private database, with a False Reject Rate of 15%. This was done with expert-found minutiae, and previously aligned fingerprints, which are two hard-to-achieve-in-an-automated-way conditions.

Other constructions that

[2] The best case occurs when L_x, L_y and L_θ are both a power of the same prime p.

III.2 Fuzzy Extractors

Also known as "Biohashing", or sometimes "biometric key extractors", another technique to integrate biometrics into cryptographic protocols is the Fuzzy Extractor. This name is also defined in [60] as follows.

Definition 4 (Fuzzy Extractors, [60])**.** A $(\mathcal{M}, m, l, t, \epsilon)$-*fuzzy extractor* is given by two procedures Gen and Rep such that:

- The generating procedure Gen is a randomized application from $\mathcal{M}$ to $\{0,1\}^l \times \{0,1\}^\star$. On input w, it extracts an l-bit string R together with a helper string $P \in \{0,1\}^\star$.
- The reproduction procedure Rep, from $\mathcal{M} \times \{0,1\}^\star$ to $\{0,1\}^l$, takes as input another data w and the helper string P, and reproduces an extracted string R', such that if P was produced by Gen on input w, and if $d(w, w') \leq t$, then $R' = R$.
- The security requirement on Gen is that if W is a random variable on $\mathcal{M}$ of min-entropy (at least) m, then the produced string R is nearly uniform, with $SD((R,P),(U_l,P)) \leq \epsilon$ where SD is the statistical distance and U_l is the uniform distribution over $\{0,1\}^l$.

This tool is interesting in order to produce keys out of biometric templates. What makes biometrics difficult to use in cryptographic protocols is the fact that two measurements of the same template always output significantly different results, while cryptographic functions expect precise inputs. Even the slightest difference leads to completely different results.

This explains the use of a fuzzy extractor: to produce a string of bits out of a random source. This requires a helper string to be stored on a given medium. The "security requirement" stated above ensures that the key cannot be deduced from that helper string.

However, as for the Secure Sketches, the requirement for the min-entropy of the source to be sufficient, since it is very difficult to estimate the statistical law of the biometric source.

Practical Fuzzy Extractors

There has been some constructions for Fuzzy Extractors that provide an insight on the general construction of such elements. [42, 41] provide several fuzzy extractor constructions that are based on real values, the key element being the quantization of the continuous data into a finite set. [183] go further: they show how the intrinsic noise of a source imposes boundaries on the length of the extracted key, and they provide a geometric construction for noisy continuous sources.

Remark 12. The design of a fuzzy extractor is narrowly linked to Rate-Distortion Theory [15], whose goal is to downsample a given source, while keeping the possibility of reconstruction of the signal. A fuzzy extractor can, in this optic, be considered as a downsampling algorithm, that keeps side information in order to reproduce the downsampling on similar data.

III.3 Cancelable Biometrics

In 2001, Ratha *et al.* [145] proposed to change views on the secure storage of biometric templates. Instead of trying to derive cryptographic elements from biometrics, they suggest to directly distort the templates in the biometric space.

The *cancelable biometrics* approach consists in doing the verification using a biometric matcher. The principle is to replace a biometric template by a revocable one, through a kind of one-way transformation.

A *cancelable biometrics* system is defined through a family of distortion functions $F = \{f_i\}_i$. The functions $f_i : \mathcal{B} \to \mathcal{B}$ transform a biometric template b into another biometric template $f_i(b)$.

The distortion functions f_i and the matching function m must verify the following properties [147]:

Condition 1 (Registration)**.** It should be possible to apply the same transformation f_i to different measurements of the same biometric trait b_1, b_2.

Condition 2 (Intra-user variability tolerance)**.** Two matching biometric traits should also match after a distortion f_i, *i.e.*

$$m(b_1, b_2) = 1 \Rightarrow m(f_i(b_1), f_i(b_2)) = 1.$$

Condition 3 (Entropy retention)**.** Two non-matching biometric traits should not match either after distortion, *i.e.*

$$m(b_1, b_2) = 0 \Rightarrow m(f_i(b_1), f_i(b_2)) = 0.$$

Condition 4 (Transformation function design)**.** This condition is made of three points:

1. **Distortion.** A biometric trait b and its distorted version $f_i(b)$ should not match: $m(b, f_i(b)) = 0$.

2. **Diversity.** Two different distortions of the same biometric trait should not match: $m(f_i(b), f_j(b)) = 0$ $(i \neq j)$.

3. **Cancelability.** It should be computationally hard to retrieve the original biometric trait b from one of its distorted versions $f_i(b)$.

The first 3 conditions enable the system to be practical, *i.e.* identification of a genuine template succeeds almost all the time whereas the identification of a non-registered biometric data leads almost always to a negative answer. In that case, the system is said to be complete. Note that, in practice, one can expect the error rates to slightly raise after the distortions; see for instance [147, Fig. 7a].

The last condition expresses a security requirement: a distorted template must have been distorted indeed (part 1), and it should not be computationally feasible to revert to the original template (part 3). Moreover, it should be possible to derive multiple different distorted templates from the original one (part 2).

There are numerous examples of cancelable biometrics systems. Some of them are depicted in the following paragraphs.

Cancelable Irises

The iris is the coloured part of the eye, between the pupil and the sclera. A deformation on the iris might be to part the iris into several angular and radial sectors, then deform each sector by enlarging or contracting its content. This gives a deformed image of the eye, that can be used in the matching algorithms, such as Daugman's IrisCode [55, 56].

This simple procedure is repeatable only after the iris is vertically aligned, so that the same deformation is applied to both templates.

More generally speaking, a good way to achieve cancelable biometrics on the iris is to apply the same picture effect on the iris picture.

Cancelable Fingerprints

A fingerprint is usually recognized thanks to its minutiae. Cancelable distortions exist at this minutiae-level. The basic idea is to displace the same minutiae at the same place from one transformation to the other. Such transformations are introduced in [147], which we sum up here:

1. The "Cartesian transformation" parts the minutiae space into rectangles $R = \{R_k\}$, and chooses a non-injective function from the rectangle set R into itself. Each rectangle R_k has now an image rectangle R'_k, and the cancelable template $f(b)$ induced by this transformation is obtained by moving each minutiae from R_k into R'_k (cf. Figure III.1).

2. "Polar transformation": A similar transformation is possible in choosing to part the minutiae space, not into rectangles, but into angular and radial sectors. In both cases, the irreversibility depends on "how much" the function is non-injective (cf. Figure III.2).

3. "Functional transformation": Another method that seems to work well consists in applying a continuous vector field $v(\vec{x,y})$ to the minutiae

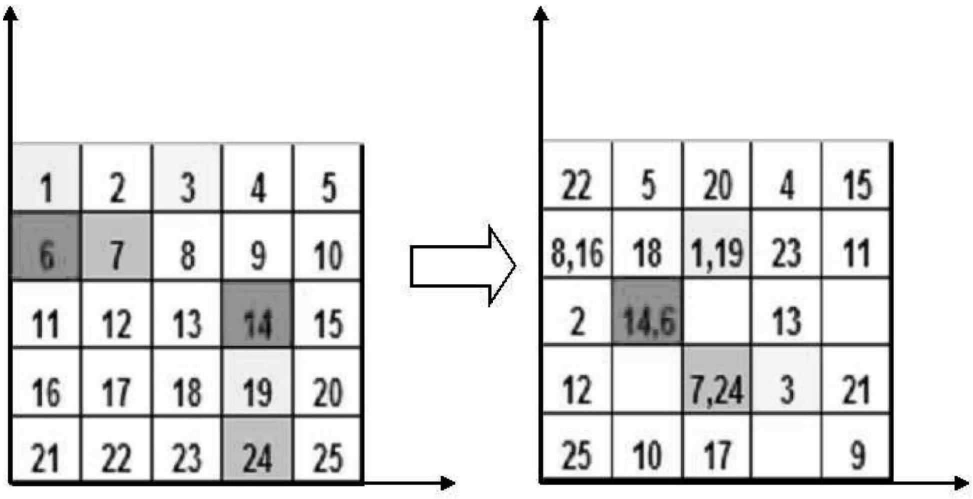


Figure III.1: Cartesian transformation from [147]

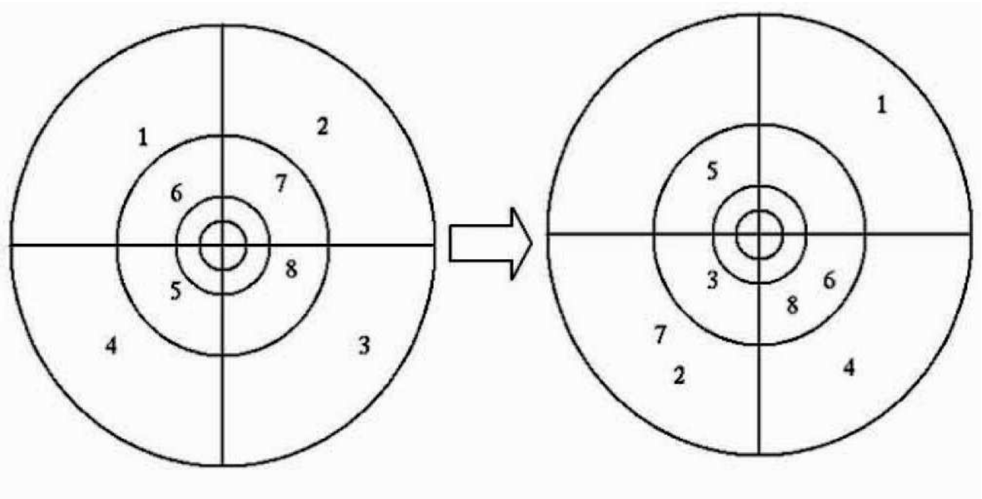


Figure III.2: Polar transformation from [147]

set: if a minutiae was at the position (x, y), it is moved to position $(x, y) + \vec{v(x, y)}$. Here again, the irreversibility depends on the vector field $\vec{v(x, y)}$ (cf. Figure III.3).

This list does not intent to be exhaustive. One could also cite [146, 7] as references for fingerprint-based cancelable biometric systems; however, this enumeration provides an overview of the basic ideas behind these works.

Applying a cancelable transformation to biometrics is bound to degrade the performances of the matcher. In term of DET curve, this means that

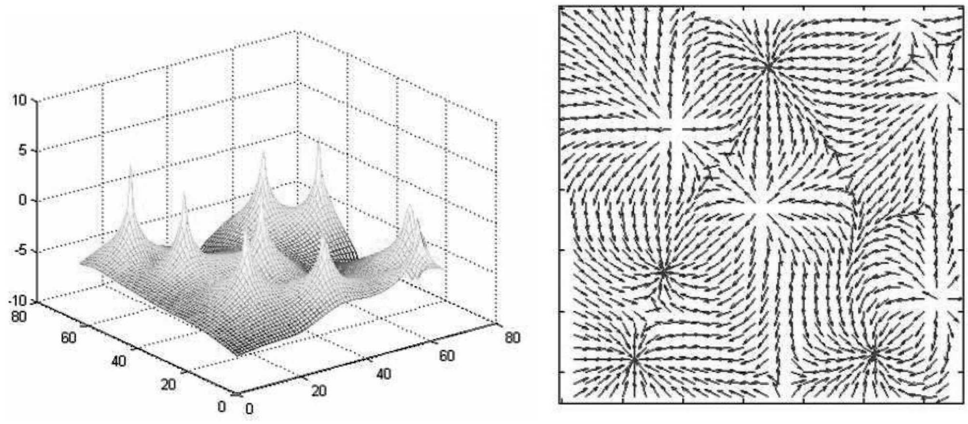


Figure III.3: Functional transformation from [147]

the new curve is “lower” than the original one. [147] presents results on a proprietary database that go in this way. However, the loss in term of FR rate at a given FA rate is still acceptable for a real-life system.

We conducted experiments on our own, on the FVC 2000 Db1 database, with a loss of performance of the same order of magnitude as [147]. We used for that the NIST open source matching algorithm, to compare original scores with scores after transformations. The cancelable functions involved are the cartesian transformation of Figure III.1, a morphing-based functional transform similar to that of Figure III.3 and a folding-based transform inspired by [7]. Figure III.4 sums up the results, and will be refered to later in this document to invoke the feasibility of Cancelable Biometrics.

Remark 13. Once more, the curve depicted in Figure III.4 only refers to the conditions 1,2 and 3 that characterize the interface of a Cancelable Biometric System; however it does not provide anything on the “cancelability” of the transformations.

III.4 Security Analysis of these Designs

In order to evaluate the security of these template protection schemes, we first describe a very general model and two typical attacks against biometric systems. We will then show how the state-of-the-art construction resist to these attacks. This document does not intend to list all the threats against a biometric system; that has already been done in *inter alia* [18, 151].

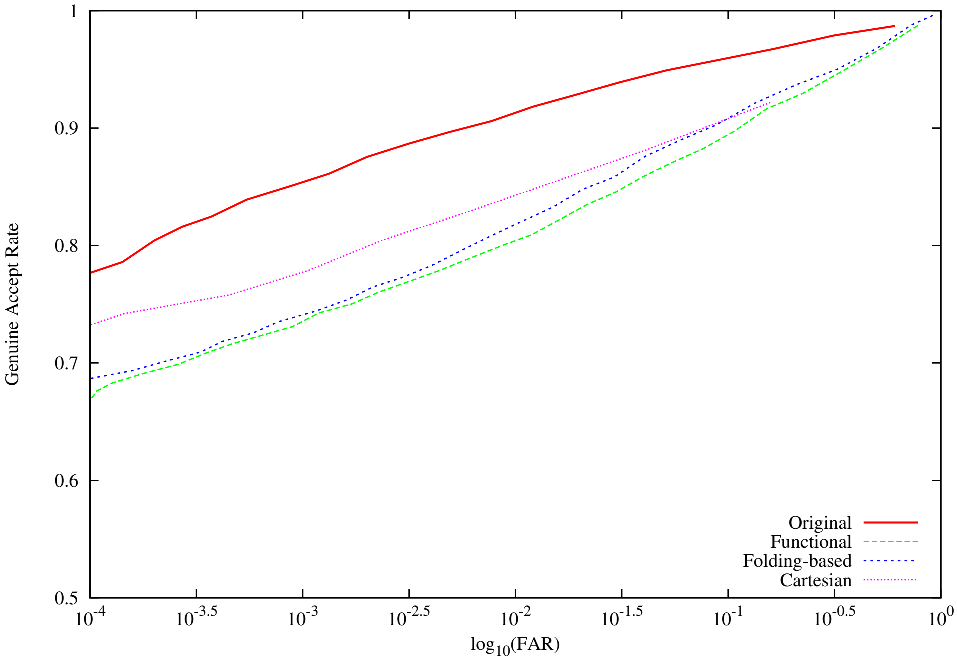


Figure III.4: Detection Error Trade-off curves for experiments on FVC 2000 Db1

General Security Model

The deployment of a template protection scheme depends strongly on the context and few general hypothesis can be made on the architecture. Here are the few guidelines that we believe make the whole “Template Protection” system relevant.

- The fundamental hypothesis is that if there is a database in the system, then its content is public.
- Following Kerckhoffs’ principle, a template protection system should be based on public functions that can be indexed by a secret key.
- Depending on the application, an adversary may make queries to the actual system with biometric templates of his choice.

The security model of such schemes does not aim at filling each security hole of a multi-terminal system; though they are possible, we will not consider classical attacks such as buffer overflows, denial of service, query injection, *etc* as they are way out of the scope of this document.

Without loss of generality, we suppose that N users are enrolled in the system, and provided their biometric template b at enrolment. The goal of an attacker is either to break the security of the system (be wrongly authenticated), or to gain information on the enrolled users, thus breaching their privacy. There is no *a priori* limitation on the possibilities of such an attacker - though we are only interested in feasible attacks. In particular, we want to accomplish these three goals:

1. An attacker should not be able to get authenticated if he is not registered;

2. An attacker should not be able to distinguish two different records, *e.g.* tell whether a record comes from a user on whom he focused.

3. An attacker should not be able to reconstruct a user's biometric template.

We will describe two attacks that fit in this model : Hill-Climbing attacks and False-Accept Attacks.

Hill Climbing Attacks

The goal of hill climbing is to determine a template that is close enough to a reference one so that it is accepted (as a FA) by the system.

If a score function is available, the principle is to execute a walk in the template domain in the way of a gradient ascent, and to finally reconstruct a close enough template. This technique should not be possible with only a "yes / no" answer (in which case, walking to a "yes" region relies on sheer luck).

This attack is easy to imagine in the case of vectors of fixed length over $\mathbb{R}$ or an alphabet $\mathcal{X}$ (the classical gradient ascent algorithm suffices); it is also doable with minutiae-based fingerprints, as was shown in [179].

The False-Accept Attack

The attack is the following: instead of presenting a fresh and genuine template b', the sensor receives one after the other a set of templates $b_1, \ldots, b_m$. If m and the FAR are large enough, then it is very probable that one of these b_i is accepted. The attacker then knows that both the original b and b_i are recognized by the system, and this information can be very revealing.

This attack is very easy to achieve. Indeed, assuming a False Accept Rate of ϵ, the probability that all m templates are correctly rejected is

$$\Pr\left[\forall j, b_j \text{ is rejected }\right] = \prod_{j=1}^{m}(1-\epsilon) = (1-\epsilon)^m$$

From this formula, we see that a database of size $\Omega(\frac{1}{\epsilon})$ is sufficient to get a False Accept with very high probability. This attack is similar to a dictionary attack used on passwords; as it was underlined by Plaga [144], this attack shows that $\log_2(\epsilon)$ is a "moral" upper-bound on the length of keys that can be extracted out of practical biometric templates.

Remark 14. ϵ is not a negligible probability. For practical systems, False Accept Rates greater than 1% are common.

Secure Sketches

We showed in Remark 10 that as soon as as the biometric source is sufficiently noisy, there can be enough information - in terms of entropy loss - to discriminate users based on the Secure Sketch available on the storing media. To be more precise, we discuss here the case for code-offset based secure sketches, as they are the simplest and only practical construction available in the literature at the time of writing.

In the code-offset Secure Sketch design, we suppose that the code C and the hash function h are public. In other words, an attacker has access to C, h, and the $\mathrm{Sk}(b_i)$ of the enrolled templates b_i.

Hill Climbing and False-Accept Attacks

Generally speaking, the hill climbing attack does not apply to code-offset based Secure Sketches. The output of the recovery is either the correct codeword (in which case the final hash test is positive) or another codeword that is at distance at least d (the minimal distance of the code). The output can also be $\perp$ which means that the recovery was not successful. Starting from a random point in the biometric space, and trying several directions to "climb the hill", one often needs to jump to a distance at least $\frac{d}{4}$ from the original point to see any difference in the output - and even that will not give information on the correctness of the jump.

On the other side, the False Accept Attack almost always "breaks" this construction. Applying this scheme to the Secure Sketch, we see that if an attacker is in possession of a $(h(b_i), c_i \oplus b_i)$, then he can compare the sketch $c_i \oplus b_i$ with each template from a - say, public - dataset $b'_1, \ldots, b'_m$. For each b_i, he decodes $c_i \oplus b_i \oplus b'_j$ into c'; whenever $h(c' \oplus c_i \oplus b_i) = h(b_i)$, a False Accept did occur. In this case, the attacker is able to reconstruct $b_i = c' \oplus c_i \oplus b_i$ without other prior knowledge than the stored record.

The success of this attack is most probable. Indeed, assuming a False Accept Rate of ϵ, the probability that all m templates are correctly rejected is

$$\Pr \forall j, b_j \text{ is rejected } = \prod_{j=1}^{m} (1-\epsilon) = (1-\epsilon)^m$$

From this formula, we see that a database of size $O(\frac{1}{\epsilon})$ is sufficient to get a False Accept with very high probability.

Remark 15. ϵ is not a negligible probability. For practical systems, False Accept Rates greater than 1% are common.

The Case for Fuzzy Vaults

Fuzzy Vaults are a special case of Secure Sketches, that differ from the code-offset construction. However, the False Accept attack works exactly in the same way for this construction. Even though [180] report 0 False Accepts in their experiments, the size of the database, and the setting of the experiment do not provide convincing arguments against this attack.

Moreover, the Fuzzy Vault construction would be efficient only if the structure of the fingerprint was random; in practice this is not the case, and the number of polynomials to be considered is much less than the worst case brute-force estimation of [180, 95]. Indeed, it is well known that fingerprints patterns can be classified among 5 general families (loosely speaking, loops, archs and whorls of different orientation). Depending on the family of a pattern, the minutiae that are to be found in the fingerprint will not be random at all; for example, minutiae along the same slope are pretty likely to be found at the same time.

Boyen [23] also took interest in the security of the Fuzzy Vault, and showed that a major downside of this kind of construction is that using it in two different systems exposes the original polynomial to an attacker who would have access to two Fuzzy Vaults based on the same biometric trait. [158] provides a formalization of this attack, among other attacks to biometric systems.

Hong *et al.* [91] also showed that using a Fuzzy Vault with a key that would not be perfectly random – such as a password, which was a countermeasure proposed by [126] to reinforce the Fuzzy Vault's security – is not secure either.

From these elements (and the other weaknesses of secure sketches), it transpires that the Fuzzy Vaults are not a good solution for storing biometric data either, without further protection.

Cancelable Biometrics

The security of cancelable biometrics is not an easy thing to achieve. The ideal situation would be for the system to be secure "as is", which means that the storage of the cancelable templates is sufficient to achieve the previously stated goals.

However, in practice, things are not that easy. Indeed, the proposed cancelable transformations are either too brutal or too soft to really achieve both security and Intra-User variability tolerance. To the best of our knowledge, no reasonable trade-off was proposed in the literature yet.As a sketch example, let us take a look at the proposed fingerprint transformations: image-block permutations (that includes cartesian and polar transformation), and functional distortion.

Image-block Permutations

The security of this transformation is difficult to quantify. Indeed, the minutiae map is split in blocks, and there are n^n possible functions that can reorder the blocks (as the function is not necessary bijective), where n is the number of blocks. This means that a brute-force reconstruction of the minutiae set would be very costly as soon as n is large enough.

On the other side, it is not necessary to reconstruct the minutiae set to be able to recognize someone. Finding a specific minutiae configuration in a block can be discriminative enough. Moreover, the minutiae set is not uniformly random in the set of all possible minutiae, and some patterns can help reconstruct the minutiae set up to a certain level of precision - think of a puzzle: if we take a n-pieces puzzle, it is almost impossible to reconstruct it using bruteforce reordering of the pieces. However, trying to fit the pieces together makes it - literally - a child's game.

Moreover, this family of transformations is extremely brutal and does not take into account the specificities of fingerprint distortions. Indeed, due to the elasticity of the skin - among other factors - it happens very frequently that a minutiae is translated from a few pixels. It makes the matching of cancelled fingerprints very difficult if the minutiae is transported from one block to the other, as these blocks will most likely not be adjacent any more. This leads to bad matching performances.

Among this family of cancelable functions, we can cite [7], in which the minutiae map is separated in two by a line, and all minutiae “above” the line are folded into the bottom part, by symmetry. This transformation is one of the easiest to attack, as an adversary can do all sort of attacks to fully reconstruct the data. For example, given two cancelled templates, the attacker can match two templates to deduce which minutiae match (those are the minutiae that were not folded), and then unfold the non-matching ones!

Functional Distortions

Functional Distortions suffer from another kind of weakness. This family provides the illusion that it is not possible to find correlations between two fingerprints because they do not match. However, being somehow clever enables to deduce which parts of the minutiae map were distorted. When that happens, it becomes pretty easy to correct these parts and have an approximated value of the original fingerprint.

This error comes from the fact that it is easy to underestimate a determined attacker and believe that the security is as hard as for a novice attacker. A novice would use the only tool at his disposal, which is a given matching function. This matching function would tell that the cancelled fingerprint and the original one are not from the same finger, and this is the security criterion put forward by many papers. Analysing in details the fingerprint repartition enables to have much more information than what one matching function provides.

Hill Climbing

False Accepts do not really threaten cancelable biometrics as does the attack described previously. On the other hand, as we work in the biometric domain, with biometric matching functions, Hill Climbing attacks can be achieved. Indeed, from conditions 2 and 3, one can tweak a first false accept b_0 by twisting it in several directions to locate an area where all the $f(b_0 + \delta)$ and a reference $f(b)$ match. The "center" of this area (depending on the function f) should not be far from b_0... in other words, it is hard to maintain condition 4.3.

Security-Performances Trade-off

Cancelable biometrics is the paradigm of the security-performances trade-off. Indeed, it is impossible to have the same level of performances with a good matching function, after a transformation.

Assuming the function f with the matching function m *produces a better ROC curve than just* m*, then* $\mathrm{m} \circ f$ *is also a matching function which behaves better than* m*. And if* $\mathrm{m} \circ f$ *behaves as well as* m*, then either f is a permutation of $\mathcal{B}$, or* m *does not take into account all the elements modified by f. In which case, another matching* m' *which takes into account these elements, should perform better than* m*.*

This trade-off is, as was showed, still unfavourable. Other works [107, 141, 18] investigated the security of such schemes, with similar warnings; as a conclusion, another kind of protection should be more efficient.

Fuzzy Extractors

The general construction of Fuzzy Extractors does not suffer from the same weaknesses as the Secure Sketches and the Cancelable Biometrics. Yet, one needs to be cautious about the instantiation of this model. For example, Secure-Sketch–based Fuzzy Extractors as stated in [60] are not a viable implementation, as they require to use the Secure Sketch as public data.

In the same fashion as [166], [40] experiments a similar distinguishing attack on Continuous Fuzzy Extractors. The authors underline the facts that 1. the Fuzzy Extractors constructed in [41] are as much subject to this attack as code-based Secure Sketches, and 2. the better the Fuzzy Extractor's error rates are, the more the attack is successful. This shows that Fuzzy Extractors are a tool that must also be used as input to further protection.

Fuzzy Extractors are still one of the most promising way to deal with biometrics. The main problem is that it is hard to build a $(\mathcal{M}, m, l, t, \epsilon)-$scheme that resists well to the fuzziness of biometrics (*i.e.* with t large enough), while in the same time having satisfying security properties: small ϵ, large m.

III.5 Inherent Limitations to ECC-based Secure Sketches

We did show in the previous section that a binary Error Correcting Codes-based Secure Sketch is not sufficient to ensure the protection of the templates by itself. This section goes one step further, and shows that Shannon theory implies a decoding threshold for biometric databases. For a given biometric system, and a given code size, there will be a minimal FRR and - that is the good news - a maximal FAR. The results published in [27] show that for practically used biometric databases (fingerprint and iris), the minimal false-reject rates are way from being negligible, even when the code dimension is not that large. Luckily, using Error Correction, the False Accept Rates will be relatively small. As we will show in Section IV.1, we are able to design Secure Sketches that are quite near to this capacity limit.

Model

We suppose that the templates are binary, and we consider two separate channels with a noise model based on the differences between any two biometric templates.

- The first channel, called the **matching channel**, is generated by errors $b \oplus b'$ where b and b' come from the same user U.

- The second channel, the **non-matching channel**, is generated by errors where b and b' come from different biometric sources.

The model of a channel is justified by the practical use of biometrics. At enrolment, a template b is registered; at the identification / verification step, a new template b' is measured; b' is obtained from b through the matching channel if the same biometric trait provided the two templates, and through the non-matching channel otherwise. In a practical biometric system, the number of errors in the **matching channel** is on average lower than in the **non-matching channel**.

Moreover, the templates are not restricted to a constant length. Indeed, when a sensor captures biometric data, we want to keep the maximum quantity of information but it is rarely possible to capture the same amount of data twice – for instance an iris may be occulted by eyelids – hence the templates are of variable length. This variability can be smoothed by forming a list of erasures, i.e. the list of coordinates where they occur. More precisely, in coding theory, an erasure in the received message is an unknown symbol at a known location. We thus have an erasure-and-error decoding problem on the **matching channel**. Simultaneously, to keep the **FAR** low, we want a decoding success to be unlikely on the **non-matching channel**: to this end we impose bounds on the correction capacity.

In the sequel, we deal with binary templates with at most N bits and assume, for the theoretical analysis that follows, that the probabilities of error and erasure on each bit are independent, i.e. we work on a binary input memoryless channel.

Remark 16. The encoding of biometric templates is often based on a geometrical interpretation - this is the case for the encodings given as examples in section II.1 (the fingerprint and the iriscode). As the patterns are not only encoded on one bit, there is a strong correlation between consecutive bits, and the memoryless channel model does not hold.

However, resorting to interleaving removes in practice the correlations between consecutive values – and, if the interleaving is random enough, makes the binary memoryless error-and-erasures channel an acceptable model.

Taking into Account Errors and Erasures

As we take into account erasures into our biometric model, we also need to slightly enhance Juels and Wattenberg's scheme. Let (b, m) and (b', m') be two biometric templates, b, b' denoting the known information, and m, m' the list of erasures, in the way IrisCodes are represented. We can represent some $(b, m) \in \{0,1\}^N \times \{0,1\}^N$ by a ternary vector $\tilde{b} \in \{0, 1, \star\}^N$, where the third symbol $\star$ represents an erasure.

The updated **xor** rule on $\{0, 1, \star\}$ is very similar to the usual one: we define $x\tilde{\oplus}x'$ to be $x \oplus x'$ if x and x' are bits, and $\star$ if one of x, x' is $\star$.

In order to protect c and b, the updated sketch will simply be the sum $z = c\tilde{\oplus}\tilde{b}$. The verification step will also use the $\tilde{\oplus}$ operation to combine z with $\tilde{b}'$ into $z\tilde{\oplus}\tilde{b}'$. The decoding can then proceed to correct incorrect bits and erasures.

Theoretical Limit

Our goal is to estimate the capacity, in the Shannon sense [163], of the matching channel when we work with a code of a given dimension. Namely, we want to know the maximum number of errors and erasures between two biometric measures that we can manage with secure sketches for this code.

Starting with a representative range of matching biometric data, the theorem below gives an easy way to estimate the lowest achievable **FRR**. The idea is to check whether the best possible code with the best generic decoding algorithm, i.e. a **maximum-likelihood** (**ML**) decoding algorithm which systematically outputs the most likely codeword, would succeed in correcting the errors.

Theorem III.1. *Let $k \in \mathbb{N}^*$, C be a binary code of length N and size 2^k, and m a random received message, from a random codeword of C, of length N with w_n errors and w_e erasures. Assume that C is an optimal code with respect to N and k, equipped with an **ML** decoder.*

If $\frac{w_n}{N-w_e} > \theta$ then the probability of decoding m is lower-bounded by $1 - o(N)$, where θ is such that the Hamming sphere of radius $(N-w_e)\theta$ in $\mathbb{F}_2^{N-w_e}$, i.e. the set $\{x \in \mathbb{F}_2^{N-w_e},\ d_H(x, \mathbf{0}) = (N-w_e)\theta\}$, contains 2^{N-w_e-k} elements.

Proof In the case of errors only (i.e. no erasures) with error-rate $p := w_n/N$, the canonical second theorem of Shannon asserts that there are families of codes with (transmission) rate $R := k/N$ coming arbitrarily close to the *channel capacity* $\kappa(p)$, decodable with ML-decoding and a vanishing (in N) word error probability P_e.

In this case, $\kappa(p) = 1-h(p)$, where $h(p)$ is the (binary) entropy function:

$$h(x) = -x\log_2 x - (1-x)\log_2(1-x).$$

Furthermore, P_e displays a threshold phenomenon: for any rate arbitrarily close to, but above capacity and any family of codes, P_e tends to 1 when N grows.

Equivalently, given R, there exists an error-rate threshold of

$$p = h^{-1}(1-R),$$

h^{-1} being the inverse of the entropy function.

Back to the errors-and-erasures setting now. Our problem is to decode to the codeword nearest to the received word on the *non-erased* positions.

Thus we are now faced with a punctured code with length $N - w_e$, size 2^k, transmission rate $R' := k/(N-w_e)$ and required to sustain an error-rate $p' := \frac{w_n}{N-w_e}$.

By the previous discussion, if

$$p' > \theta_0 := h^{-1}(1-R'),$$

then no code and no decoding procedure exist with a non-vanishing probability of success.

The number of vectors of $\mathbb{F}_2^M$ of weight αM is $\binom{M}{\alpha M}$ which, through the Stirling approximation, is equivalent to $\frac{2^{Mh(\alpha)}}{\sqrt{2\pi\alpha(1-\alpha)n}}$. This shows that a Hamming sphere of radius αM in $\mathbb{F}_2^M$ contains more than $2^{h(\alpha)M}$ elements.

In other words, the normalized radius θ such that the sphere of radius $(N-w_e)\theta$ contains 2^{N-w_e-k} elements, is such that $\theta \leq \theta_0$. This concludes the proof.

□

This result allows us to estimate the correcting capacity of a biometric matching channel with noise and erasures under the binary input memoryless channel hypothesis.

Indeed applying Theorem III.1 to the **matching channel** gives a lower-bound on the **FRR** achievable (i.e. the *best* **FRR**), whereas applying it to the **non-matching channel** gives an upper-bound of the **FAR** (say the *worst* **FAR**).

Corollary III.1. *For a given biometric authentication system based on a binary secure sketch of length N and dimension k, and a given biometric database $\mathcal{B} = \{b_i\}$, let the function $f_{N,k}$ be $f_{N,k}(\tilde{y}) = \frac{w_n}{N-w_e} - h^{-1}\left(1 - \frac{k}{N-w_e}\right)$, with w_n the number of 1's occuring in $\tilde{y}$ and w_e the number of $\star$*[3]*.*

Define $p^G_{N,k}(x)$ (resp. $p^I_{N,k}(x)$) as the probability density of results of all genuine (resp. impostor) comparisons $f_{N,k}(\tilde{b}\tilde{\oplus}\tilde{b}')$ for $b, b' \in \mathcal{B}$.

Under these hypotheses, and for large enough N, the following inequalities stand:

$$FRR \geq \int_0^{+\infty} p^G_{N,k}(t)dt \text{ and } FAR \leq \int_{-\infty}^{0} p^I_{N,k}(t)dt.$$

Proof According to Theorem III.1, the decoding of a received vector $c\tilde{\oplus}\tilde{b}\tilde{\oplus}\tilde{b}'$ is possible with non-negligible probability only if the vector $\tilde{b}\tilde{\oplus}\tilde{b}'$, containing w_e erasures and w_n errors is such that $\frac{w_n}{N-w_e} \leq h^{-1}(1 - \frac{k}{N-w_e})$.

In other words, if $f_{N,k}(\tilde{b}\tilde{\oplus}\tilde{b}') > 0$ and b and b' come from the same user, then there is a False Reject - except with negligible probability. Conversely, if $f_{N,k}(\tilde{b}\tilde{\oplus}\tilde{b}') > 0$ and b and b' come from different user, then the reject will be genuine.

The corollary follows from the definition of $p^G_{N,k}$ and $p^I_{N,k}$. □

In other words, Corollary III.1 can lead to a kind of theoretical ROC curve which is not represented thanks to the classical matching score distributions but with the dimension of the underlying optimal code on the abscissa axis. Therefore, from a given database and a given features extraction scheme – dedicated to discrete representation – it is possible to induce an approximation of the error-rates one can expect from templates of the same quality. In particular, it may help to evaluate the efficiency of the extraction algorithm.

In the next section, we shall illustrate practical implications of these Theorem and Corollary.

Application to Biometric Data

We now present the estimation of these optimal performances on several public biometric databases.

[3] The notation $\tilde{y}$ is used here to emphasize the fact that $\tilde{y}$ takes values in $\{0, 1, \star\}$

Our Setting: Data Sets and Templates

We made our experiments on the ICE 2005 and CASIA v1 datasets for the iris (see Section I.1) and the FVC 2000 (Db. 2) dataset for the fingerprint (see Section I.1). The irises are encoded into IrisCodes following [55], and the fingerprints are quantized into binary vectors following Algorithm 3.

For each dataset, we will represent the boundaries on **FRR** and **FAR**. The matching-score distribution is given on Figures III.5 and III.6, where the scores of matching (intra-eyes / intra-finger) and non-matching (inter-eyes / inter-finger) comparisons are represented. We can see that there is an overlap between the two curves, and that the number of errors to handle in the matching channel is large.

On iris matching-channel an additional difficulty originates from the number of erasures which varies, for instance for ICE, from 512 to 1977.

Although we know that all bits are not independent and that they do not follow the same distribution (see e.g. [90]), following (II.1.2) the typical matching score computation does not use any internal correlations between bits of the iris codes. So in this setting it is coherent to suppose the matching channel to be a binary input memoryless channel with independent bit errors and erasures. It will thus be possible to apply Theorem III.1 in this context.

For the fingerprint, we selected 6 images per finger for the enrolment phase, one 2048-bit template per enrolled finger is obtained, possibly with some erasures, and the remaining 200 images are kept for verification. As the verification step is done on just one picture, the verification template will always contain at least $2048 - 1984 = 64$ erasures; this is well captured by the decoding algorithm. To increase the overall number of comparisons, we iterate the tests for every choices of 6 images. This gives us a genuine match count of 5600, and an impostor match count of 19800.

Any other biometrics may be used to apply Theorem III.1 as soon as we succeed in getting a discrete representation of the templates associated to a Hamming distance classifier.

Performances Estimation on these Databases

For each one of these databases we represent, in Figures III.5, III.6 and III.7 (subfigure (a)), the relative Hamming distance distribution following Eq. (II.1.2) for the matching and the non-matching channel and the corresponding FRR and FAR curves.

We also estimate the optimal performances given by Corollary III.1 and the results are drawn in Figures III.5, III.6 and III.7 (subfigure (c)). These last curves correspond to the best FRR achievable with respect to the code's dimension and the greatest possible FAR as a function of this dimension; they are obtained by computing number of errors and erasures for each $\tilde{b}\tilde{\oplus}\tilde{b}'$, computing the distribution of the corresponding $f_{N,k}$, and summing the distributions over $\mathbb{R}_+$ and $\mathbb{R}_-$.

From the Hamming Distance distributions, it is obvious that, while iris recognition performs well with the IrisCode algorithm, the chosen quantization is not as well adapted to fingerprint matching. Therefore, the different results we shall have will significantly differ.

For the three datasets, we see that the ratio of errors to handle to approach the Equal Error Rate – **EER** – is very high, which is a problem for classical correcting codes as explained in the next section.

We summed up some of the numerical limits on **FAR** and **FRR** in Tables III.2 and III.1, for dimensions likely to be chosen for practical purposes. A general consequence is that the dimension of the code can not be chosen too high in order to keep good **FR** rates.

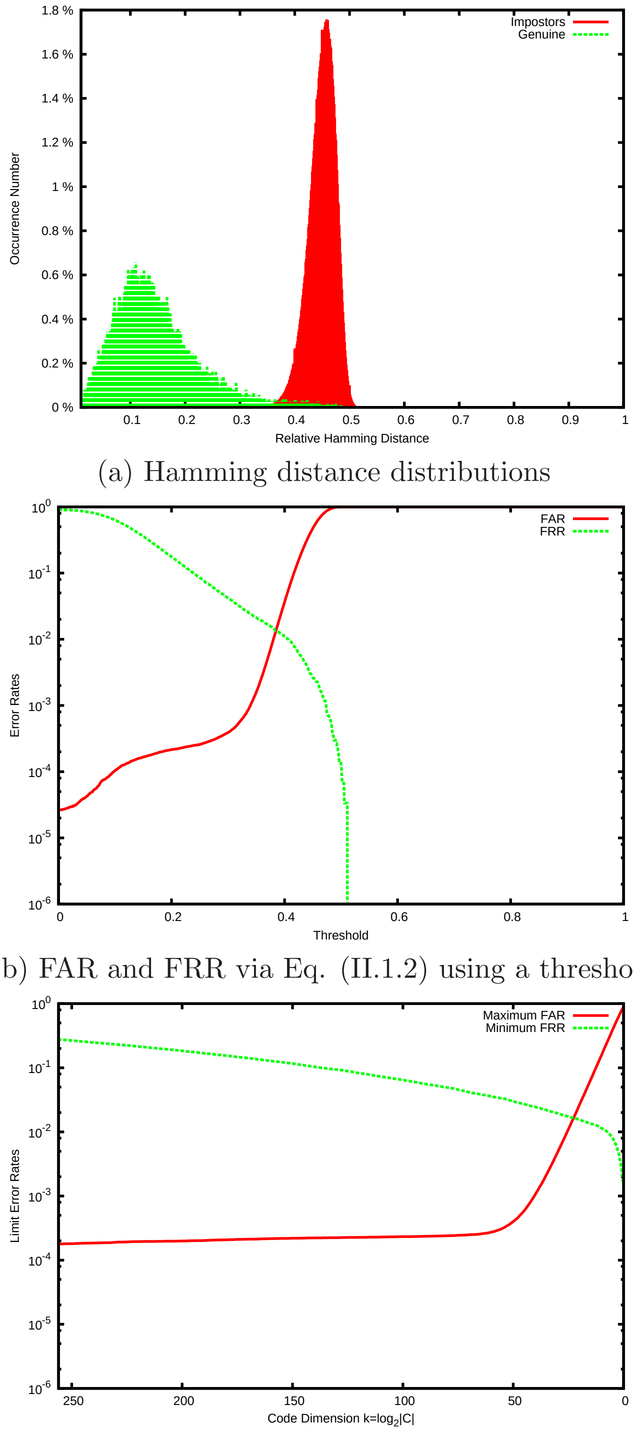
Code's dimension	Minimum **FRR**		
	ICE	CASIA	FVC
42	$2.49 \cdot 10^{-2}$	$3.15 \cdot 10^{-2}$	$0.59 \cdot 10^{-2}$
64	$3.76 \cdot 10^{-2}$	$4.47 \cdot 10^{-2}$	$1.26 \cdot 10^{-2}$
80	$4.87 \cdot 10^{-2}$	$5.77 \cdot 10^{-2}$	$1.93 \cdot 10^{-2}$
128	$9.10 \cdot 10^{-2}$	$9.18 \cdot 10^{-2}$	$5.87 \cdot 10^{-2}$

Table III.1: Theoretical Limits on Studied Databases - Minimum FRR

Code's dimension	Maximum **FAR**		
	ICE	CASIA	FVC
42	$8.14 \cdot 10^{-4}$	$1.13 \cdot 10^{-4}$	$17.88 \cdot 10^{-2}$
64	$2.74 \cdot 10^{-4}$	0	$10.32 \cdot 10^{-2}$
80	$2.57 \cdot 10^{-4}$	0	$7.07 \cdot 10^{-2}$
128	$2.41 \cdot 10^{-4}$	0	$2.67 \cdot 10^{-2}$

Table III.2: Theoretical Limits on Studied Databases - Maximum FAR

Note that Theorem III.1 gives us estimations of the theoretical limits based on asymptotic analysis under a memoryless channel hypothesis, i.e. independent bits. In principle, it could be possible to expect more efficiency without resorting to bit interleaving which in practice makes the channel memoryless.

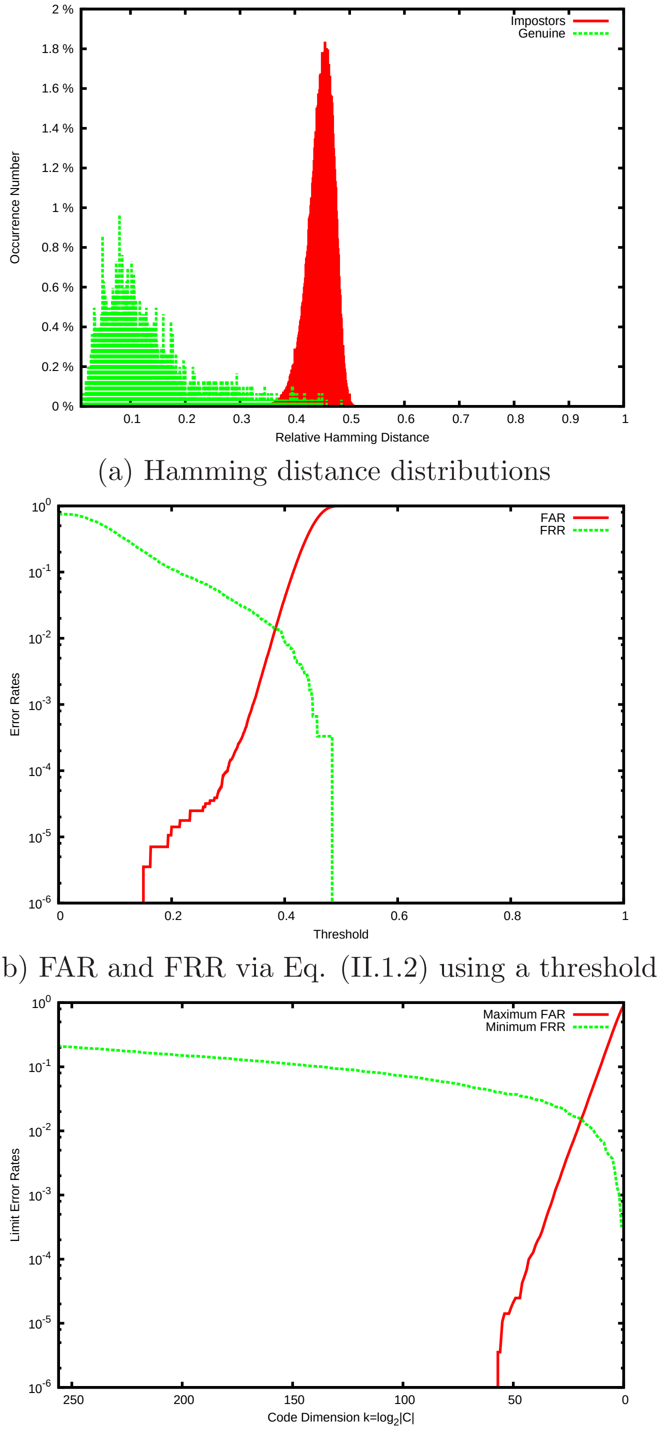


(a) Hamming distance distributions

(b) FAR and FRR via Eq. (II.1.2) using a threshold

(c) Worst FAR and best FRR w.r.t. the code dimension

Figure III.5: The ICE 2005 Dataset, IrisCodes

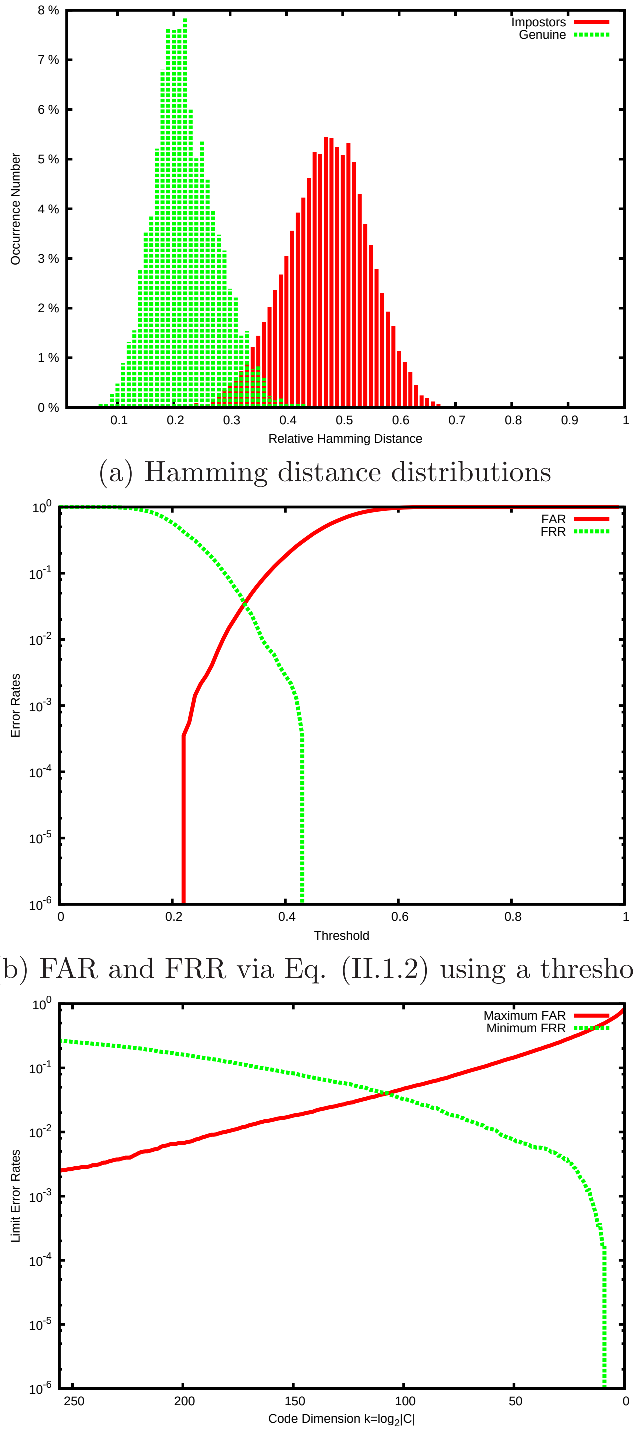


(a) Hamming distance distributions

(b) FAR and FRR via Eq. (II.1.2) using a threshold

(c) Worst FAR and best FRR w.r.t. the code dimension

Figure III.6: The CASIA v1 Dataset, IrisCodes



(a) Hamming distance distributions

(b) FAR and FRR via Eq. (II.1.2) using a threshold

(c) Worst FAR and best FRR w.r.t. the code dimension

Figure III.7: The FVC 2000 (Db. 2) Dataset, Binary Encoding

However this would require highly intricate modelling of the matching channel, and it seems unreasonable to expect that the decoding problem would be within reach of present day algorithms.

Chapter IV

Enhancing the Status Quo

> Status quo, you know, that is
> Latin for "the mess we're in."
>
> Ronald Reagan

This chapter presents three suggested constructions published respectively in [27, 31, 32]. They have in common the fact of being based on the classical template protection algorithms, presented in Chapter III. Section IV.1 presents a Secure Sketch construction that is very near to the Shannon capacity as referred to in Theorem III.1, using an iterative decoding algorithm on concatenated codes. Section IV.2 shows how it is possible to use the best out of Secure Sketches and Cancelable Biometrics. Finally, Section IV.3 presents a more exotic construction that uses Cancelable Biometrics while preventing replay attacks.

IV.1 A Near Optimal Construction

Quantization and BCH codes

In known applications of secure sketches to quantized biometrics, for instance [100, 176], the error correcting codes are seen directly to act as a Hamming distance classifier at a given threshold. Hence, the correction capacity naturally corresponds to the threshold we want to reach. To this end, the use of BCH codes [21] is proposed: the advantage is their existence for a wide class of parameters, the main drawback is that the correction capacity is a hard constraint for the dimension.

As an illustration, in [100] the quantization technique is applied to face recognition on two databases, FERET database [143] and one from Caltech [185]. A Hamming distance classifier gives Equal Error Rates (EER) of 2.5% and 0.25% respectively for a threshold greater than 0.32 with code length 511. Unfortunately to achieve this minimal distance, the BCH code has dimension

1. A BCH of dimension 40 enables a threshold of 0.185 with a **FRR** greater than 10% and 1% on the precited databases respectively.

This phenomenon holds in [176] as well as in our first experiments on the FVC2000 dataset. Following Fig. III.7, we remark that to achieve a **FRR** better than the EER, the threshold is high: for example, for a rate around 2%, the threshold is near 0.4 which is not realistic with non-trivial BCH codes. To overcome this limitation, we propose in the sequel to use more appropriate codes.

IrisCodes and Concatenated Codes

More efficient codes are proposed in [87]. The secure sketch scheme is applied with a concatenated error-correcting code combining a Hadamard code and a Reed-Solomon code. More precisely, the authors use a $[32, k_{RS}, 33 - k_{RS}]_{64}$ Reed-Solomon code and a $[64, 7, 32]_2$ Hadamard code: a codeword of 2048 bits is in fact constructed as a set of 32 blocks of 64 bits where each block is a codeword of the underlying Hadamard code. As explained in [87], the Hadamard code is introduced to deal with the background errors and the Reed-Solomon code to deal with the bursts (e.g. caused by eyelashes, reflections, ...).

Note that in this scheme, the model is not exactly the same as ours, as the masks are not taken into account. Moreover, the quality of the database used in [87] is better than the public ones we worked with. The mean intra-eye Hamming distance reported in the paper is 3.37% whereas this number becomes 13.9% in the ICE database, which means that we must have a bigger correcting capacity. The inter and intra-eyes distributions reported by the authors is drawn on Fig. IV.1.

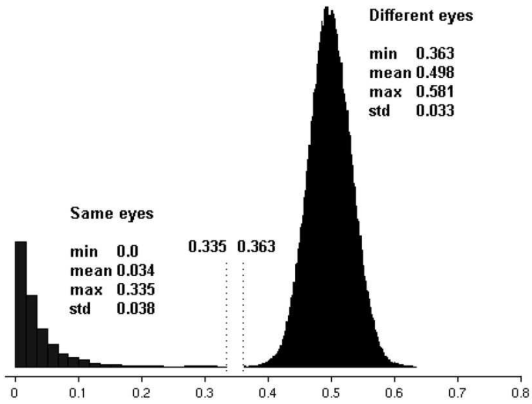


Figure IV.1: Hamming distance distributions from [87]

Even if [87] reports very good results on their experiments with a 700-

image database , the codes do not seem appropriate in our case as the same parameters on the ICE database gave us a too large rate of **FR** (e.g. 10% of **FR** with 0.80% of **FA**), even for the smallest possible dimension of the Reed-Solomon code when $t_{RS} = 15$.

To sum up, with respect to the Hamming distance distribution in Figures III.5, III.6 and III.7, we need to find correcting codes with higher correction capacity. Achieving performances closer to the theoretical estimation given in section III.5 is also a great motivation.

Description of the Two-Dimensional Iterative Min-Sum Decoding Algorithm

We now describe a very efficient algorithm which will help us to overcome the difficulties mentioned above.

For a linear code with a minimum distance d_{min}, we know that an altered codeword with w_n errors and w_e erasures can always be corrected, disregarding decoding complexity issues, provided that $2w_n + w_e < d_{min}$.

Classical algebraic decoding of BCH codes and concatenated Reed-Solomon codes achieve this bound, but hardly more. This upper bound is however a conservative estimate: it has been known since Shannon's days that it is possible in principle to correct many more errors and erasures, all the way to the channel capacity. In practice, *iterative decoding* algorithms are now known to be capable of achieving close-to-capacity performance, for such code families as LDPC or turbo codes. It is therefore natural to try and bring in iterative decoding to improve the performances of secure sketches that use algebraic decoders.

LDPC codes and turbo codes are however not usually designed for such noisy channels as the type we have to deal with: in particular, classical turbo codes are known to have a non-negligible error-floor in the high noise area, where they do not behave as well as desired.

We have therefore chosen to use product codes. Under the high noise condition particular to biometrics, we have to use codes of small dimension to apply maximum-likelihood decoding (exhaustive search) to the constituent codes; we can therefore alternate between both decoders with an iterative process. This yields a particularly efficient blend of iterative decoding and exhaustive search.

We now describe product codes together with the specific iterative decoding algorithm we will use. A product code $C = C_1 \otimes C_2$ is constructed from two codes: $C_1[N_1, k_1, d_1]_2$ and $C_2[N_2, k_2, d_2]_2$. The codewords of C can be viewed as matrices of size $N_2 \times N_1$ whose rows are codewords of C_1 and columns are codewords of C_2, see Fig. IV.2.

This yields a $[N_1 \times N_2, k_1 \times k_2, d_1 \times d_2]_2$ code. When k_1 and k_2 are small enough for C_1 and C_2 to be decoded exhaustively, a very efficient iterative decoding algorithm is available, namely the *min-sum* decoding algorithm. Min-sum decoding of LDPC codes was developed by Wiberg [187] as a particular instance of message passing algorithms. In a slightly different setting it was also proposed by Tanner [174] for decoding generalized LDPC (Tanner) codes. The variant we will be using is close to Tanner's algorithm and is adapted to product codes. Min-sum is usually considered to perform slightly worse than the more classical sum-product message passing algorithm on the Gaussian, or binary-symmetric channels, but it is specially adapted to our case where knowledge of the channel is poor, and the emphasis is simply to use the Hamming distance as the appropriate basic cost function.

Let (x_{ij}) be a vector of $\{0,1\}^{N_1\times N_2}$. The min-sum algorithm associates to every component x_{ij} a cost function κ_{ij} for every iteration of the algorithm. The cost functions are defined on the set $\{0,1\}$. The initial cost function κ_{ij}^0 is defined as

$$\kappa_{ij}^0(x) = 1 - \delta_{x,x_{ij}}$$

where $\delta_{a,b}$ is Kroenecker's symbol ($\delta_{a,b} = 1$ if $a = b$, and is 0 otherwise).

In other words, switching a component x_{ij} costs 1, while keeping the component is costless.

A *row* iteration of the algorithm takes an *input* cost function κ_{ij}^{in} and produces an *output* cost function κ_{ij}^{out}. The algorithm first computes, for every row i and for every codeword $c = (c^{(1)} \dots c^{(N_1)})$ of C_1, the *sum*

$$\kappa_i(c) = \sum_{j=1}^{N_1} \kappa_{ij}^{in}(c^{(j)})$$

which should be understood as the cost of putting codeword c on row i. The algorithm then computes, for every i, j, κ_{ij}^{out} defined as the following *min*, over

$$c = \begin{pmatrix} c_{1,1} & \dots & c_{1,j} & \dots & c_{1,n_1} \\ & & \vdots & & \\ c_{i,1} & \dots & c_{i,j} & \dots & c_{i,n_1} \\ & & \vdots & & \\ c_{n_2,1} & \dots & c_{n_2,j} & \dots & c_{n_2,n_1} \end{pmatrix}$$

$$\forall i \in [\![1, n_2]\!], (c_{i,1}, c_{i,2}, \dots, c_{i,n_1}) \in C_1$$
$$\forall j \in [\![1, n_1]\!], (c_{1,j}, c_{2,j}, \dots, c_{n_2,j}) \in C_2$$

Figure IV.2: A codeword of the product code $C_1 \otimes C_2$ is a matrix where each line is a codeword of C_1 and each column a codeword of C_2

the set of codewords of C_1,

$$\kappa_{ij}^{out}(x) = \min_{c \in C_1, c_j = x} \kappa_i(c).$$

This last quantity should be thought of as the minimum cost of putting the symbol x on coordinate (ij) while satisfying the row constraint.

A *column* iteration of the algorithm is analogous to a row iteration, with simply the roles of the row and column indexes reversed, and code C_2 replacing code C_1. Precisely we have

$$\kappa_j(c) = \sum_{i=1}^{N_2} \kappa_{ij}^{in}(c^{(i)}) \tag{IV.1.1}$$

and

$$\kappa_{ij}^{out}(x) = \min_{c \in C_2, c_i = x} \kappa_j(c).$$

The algorithm alternates row and column iterations as illustrated by Fig. IV.3. After a given number of iterations (or before, if we find a codeword) it stops, and the value of every symbol x_{ij} is put at $x_{ij} = x$ if $\kappa_{ij}^{out}(x) < \kappa_{ij}^{out}(1-x)$. If $\kappa_{ij}^{out}(x) = \kappa_{ij}^{out}(1-x)$ then the value of x_{ij} stays undecided (or erased).

The following theorem, proved by Zémor, is fairly straightforward and illustrates the power of min-sum decoding.

Theorem IV.1. *If the number of errors is less than $d_1 d_2/2$, then two iterations of min-sum decoding of the product code $C_1 \otimes C_2$ recover the correct codeword.* □

Proof

Without loss of generality, the correct codeword is the all-zero vector.

Suppose that after the second iteration the algorithm prefers 1 to 0 in some position (i, j). This means that the cost $\kappa_j(c)$ (IV.1.1) of some non-zero codeword c of C_2 is smaller than the cost $\kappa_j(0)$ of the zero column vector, $\kappa_j(c) < \kappa_j(0)$.

Now the cost $\kappa_j(c)$ of putting codeword c in column j is equal to the Hamming distance between the received vector (x_{ij}) and a vector $\mathbf{x}_c$ that has c in column j and only rows belonging to C_1. The cost $\kappa_j(0)$ of putting the zero vector in column j is equal to the Hamming distance between the received vector (x_{ij}) and a vector $\mathbf{x}_0$ that has only zeros in column j and only rows belonging to C_1. In other words, $d(\mathbf{x}, \mathbf{x}_c) < d(\mathbf{x}, \mathbf{x}_0)$.

Since c belongs to C_2 and is non-zero, it has weight at least d_2, and $\mathbf{x}_c$ has at least d_2 rows of weight at least d_1 and at distance at least d_1 from

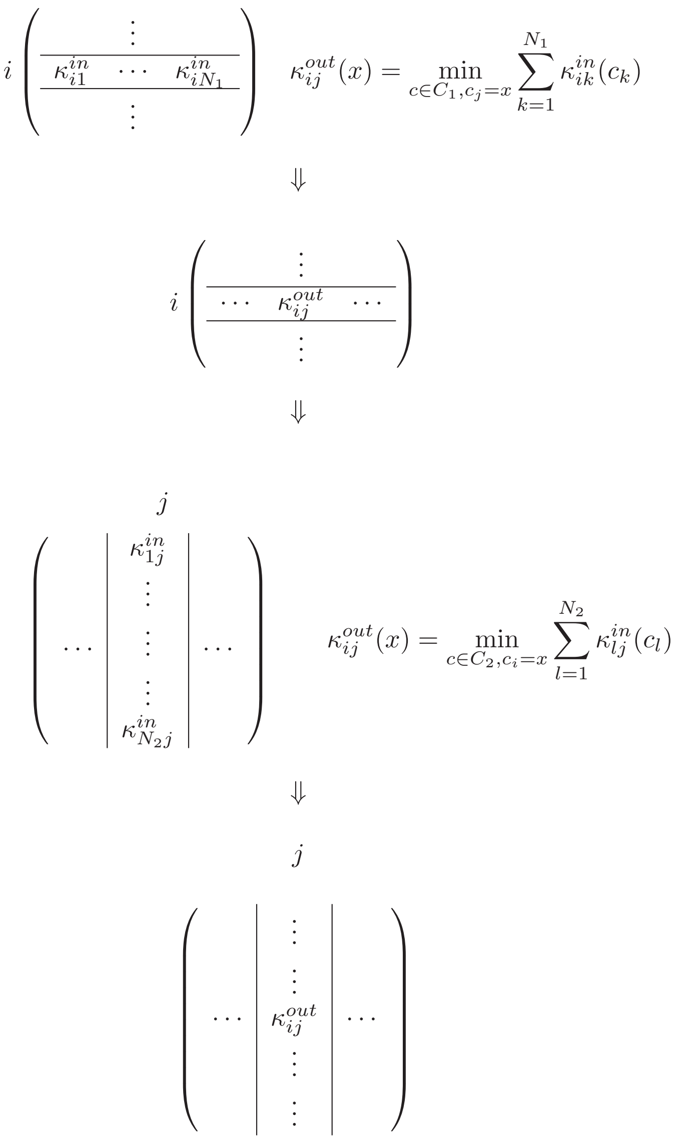

i
$\kappa_{i1}^{in} \cdots \kappa_{iN_1}^{in}$
$\kappa_{ij}^{out}(x) = \min_{c \in C_1, c_j = x} \sum_{k=1}^{N_1} \kappa_{ik}^{in}(c_k)$
$\Downarrow$
i
$\cdots \kappa_{ij}^{out} \cdots$
$\Downarrow$
j
κ_{1j}^{in}
$\kappa_{N_2 j}^{in}$
$\kappa_{ij}^{out}(x) = \min_{c \in C_2, c_i = x} \sum_{l=1}^{N_2} \kappa_{lj}^{in}(c_l)$
$\Downarrow$
j
$\cdots \kappa_{ij}^{out} \cdots$

Figure IV.3: A row iteration followed by a column one

the corresponding rows of $\mathbf{x}_0$. Therefore, the Hamming distance between $\mathbf{x}_c$ and $\mathbf{x}_0$ is at least $d(\mathbf{x}_c, \mathbf{x}_0) \geq d_2 d_1$.

From the triangle inequality, if the received vector (x_{ij}) is closer to $\mathbf{x}_c$ than to $\mathbf{x}_0$, it must have weight at least $d_1 d_2 / 2$. □

Remark 17. In the binary symmetric channel of transition probability p, the probability of receiving a given n-word translated by a vector e is

$$\Pr[x+e|x] = p^{w(e)}(1-p)^{n-w(e)} = (1-p)^n(\frac{p}{1-p})^{w(e)}.$$

The initial cost function is in fact affine in $\log \Pr[x+e|x]$ if x was a sent codeword; this decoding algorithm is well adapted to the BSC channel.

IV.2 Combining Cancelable Biometrics with Secure Sketches

Cancelable biometrics

Although cancelable biometrics [145] have been introduced with similar objectives to biometric secure sketches, *i.e.* to limit the privacy threats raised by biometric authentication, the methods are somewhat opposed. As stated in Section III.3, the idea is to transform biometric data with an irreversible transformation and to perform the matching directly on the transformed data. The advantage pointed out by [145, 147, 146] is the capability to use existing feature extraction and matching algorithms. However, the main drawback is that, with classical matching algorithms, the performances quickly decrease when the transformation breaks the structure of biometrics. For instance for fingerprints, if the matching uses minutiae then a random permutation of image's blocks leads to bad FR rates (cf. [147, Fig. 7 (a) Cartesian case] and Fig. III.4). There is thus a compromise between irreversibility and performances.

Note that the security does not concern the same layer as secure sketches does. Indeed, with cancelable biometrics the matching is performed on transformed data and so the original data is never clearly revealed after the enrolment. Thus, it protects the representation of biometrics whereas secure sketch is a way to protect the storage of your biometric data until you present a close template.

Cancelable and secure biometrics

We now apply secure sketches to cancelable biometrics. In doing so, our goal is to add the security of both schemes together and to switch from the matching step of cancelable biometrics to an error-correction problem.

Assume that the biometric templates are in the metric space $\mathcal{B}$, let f be a transformation on $\mathcal{B}$, we propose to use an $(\mathcal{B}, m, m', t)$-secure sketch with functions (Sk, Rec) as follows. We define the enrolment function Enrol by

$$\mathsf{Enrol}(b;\, f) = \mathrm{Sk}(f(b)). \tag{IV.2.1}$$

The verification function Verif takes an enrolled data P, a vector $b' \in \mathcal{B}$ and the function f as inputs and outputs $\mathrm{Rec}(P,\, f(b'))$, *i.e.* $f(b)$ whenever $d(f(b'), f(b)) \leq t$.

It keeps the correction's principle of secure sketches and the recovery of b from $\mathsf{Enrol}(b;\, f)$ is at least as hard as the recovery of $f(b)$ from the sketch $\mathrm{Sk}(f(b))$. The more f hides w, the greater the security. This construction also enables to enhance the diversity of the enrolled data, as the function f can depend on the user, or on the application, or on both[1].

This combination holds the advantage that it makes for an attacker a distinguishing attack more difficult. As the cancelable template $f(b)$ is not made available, the attacker can only distinguish b from two templates b_1 and b_2 if at least one of the verification of $\mathsf{Enrol}(b;\, f)$ with a template b' does not fail, which is not the case when using matching functions.

Moreover, it is worth noting that in some applications, f could be stored in a token directly by the user – especially when f is invertible – so that the transformation is unknown to the server and from the outside. Indeed, the cancelable transformations can be computed by the user before sending data for enrolment or verification. In this setting, the function f acts as a secret key – and this weakens the model as well as the acceptability by users.

Anonymous protocol. To avoid any tracking of authentications, we can also change the transformation used for a user after each succeeded verification. The transmitted data $f(w')$ will then be unrelated to the next ones and thus it allows to achieve an anonymous authentication protocol. This can be done by applying a new transformation g on the recovered data $f(w)$ and thereafter to transmit $g \circ f$ for the next verification; the cost for this is a greater loss of performances. We will further develop the idea of anonymity through cancelable biometrics in Section IV.3.

Security analysis

We consider the functions Enrol and Verif which are defined in section IV.2 via an $(\mathcal{B}, m, m', t)$-secure sketch with functions (Sk, Rec) and a transformation f on $\mathcal{B}$. Two situations are possible: f can be public or secret.

In both cases, the following lemma is straightforward. We underline that it implies that the protection of b is at least as strong as the protection of $f(b)$ achieved by the secure sketch, under the condition that the entropy of $f(b)$ is sufficiently high.

Lemma IV.1. *For all random variables B on $\mathcal{B}$,*

$$\overline{\mathbf{H}}_\infty(B \mid \mathsf{Enrol}(B;\, f)) \geq \overline{\mathbf{H}}_\infty\left(f(B) \mid \mathrm{Sk}(f(B))\right).$$

If f is invertible, it is an equality.

[1] The number of records is still limited by the dimension of the underlying code in order to avoid trivial False-Accept Attacks.

Proof The average min-entropy of a random variable X knowing another random variable Y can only be decreased by applying a function to X (see the appendix proposition A.1 for an explicit statement).

As $\mathsf{Enrol}(B;\, f) = \mathrm{Sk}(f(B))$, application of f to the random variable B only decreases the information known on B; thus the Lemma. □

Via definition 3, we deduce that for all random variables B on $\mathcal{B}$ with $\mathbf{H}_\infty(f(B)) \geq m$, then

$$\overline{\mathbf{H}}_\infty(B \mid \mathsf{Enrol}(B;\, f)) \geq m'.$$

We also see that the more f is irreversible, the more it would be difficult to recover B in general. For instance, if f is such that

$$\Pr(B = b) \leq \frac{\Pr(f(B) = f(b))}{\lambda}$$

with $\lambda \geq 1$, we obtain

$$\overline{\mathbf{H}}_\infty(B \mid \mathsf{Enrol}(B;\, f)) \geq \overline{\mathbf{H}}_\infty(f(B) \mid \mathrm{Sk}(f(B))) + \log_2 \lambda. \tag{IV.2.2}$$

If the entropy of $f(B)$ is sufficiently large, it means that the security of both schemes are added together.

However, as we stated before, the entropy of biometric data is difficult to estimate, and the more f will be irreversible, the more the entropy of $f(B)$ will decrease. In term of entropy, there is thus a kind of compensation between security of secure sketches and security of cancelable transformation. In this way, for the code-offset construction where the maximal loss of entropy is independent on the input's entropy. In other words, the amount of information released to an adversary does not increase by adding a cancelable transformation:

Proposition IV.1. *Given $f : \mathbb{F}_q^n \to \mathbb{F}_q^n$, let $\alpha \geq 0$ such that for all random variables B on $\mathbb{F}_q^n$,*

$$\mathbf{H}_\infty(f(B)) \geq \mathbf{H}_\infty(B) - \alpha.$$

For code-offset $(\mathbb{F}_q^n, m, m-(n-k)\log_2 q, t)$-secure sketch $(\mathrm{Sk}, \mathrm{Rec})$, the average min-entropy of B knowing the enrolled data $\mathsf{Enrol}(B;\, f)$ does not depend on α, and is bounded by:

$$\overline{\mathbf{H}}_\infty(B \mid \mathsf{Enrol}(B;\, f)) \geq \mathbf{H}_\infty(B) - (n-k)\log_2 q.$$

Proof This proposition states that the average min-entropy of B knowing enrolled data that reveals, in the worst case, less information than $\mathrm{Sk}(B)$, is at least as large as the minimal entropy retention of Secure Sketches. It is an intuitive property that is corroborated by the following lower bounds:

- The secure sketch provides that

$$\overline{\mathbf{H}}_\infty(f(B) \mid \mathrm{Sk}(f(B))) \geq \mathbf{H}_\infty(f(B)) - (n-k)\log_2 q; \qquad \text{(IV.2.3)}$$

- For such a f, Eq. IV.2.2 states that:

$$\overline{\mathbf{H}}_\infty(B \mid \mathsf{Enrol}(B;\, f)) \geq \overline{\mathbf{H}}_\infty\left(f(B) \mid \mathrm{Sk}(f(B))\right) + \alpha;$$

- The proof is completed by adding these equations with the constraint

$$\mathbf{H}_\infty(f(B)) \geq \mathbf{H}_\infty(B) - \alpha.$$

□

For the specific case where f is invertible and secret, the security of secure sketches and cancelable biometrics also add up together: an attacker would try to recover $f(w)$ from the sketch and thereafter to construct w from $f(w)$.

Moreover, the construction brings to secure sketches the advantages of cancelable biometrics, and among them the protection against cross-matching attacks. Indeed, starting from 2 sketches $\mathrm{Sk}(f_1(b))$ and $\mathrm{Sk}(f_2(b))$, it seems difficult to establish a link between them as $f_1(b)$ should not match with $f_2(b)$. Finally, contrary to secure sketches where a successful attack of a sketch compromises forever the underlying biometric data, here cancelable biometrics act as a second layer of protection.

An example for fingerprints

To underline the feasibility and the interest of this construction, we experiment it on the fingerprint FVC2000 second database [116]. In fact, we merge three techniques: 1. a cancelable biometrics transformation, 2. an enrolment algorithm adapted from the reliable component scheme [176], slightly modified with techniques from [100], to extract binary features and 3. the coding/decoding algorithm presented in Section IV.1 for the secure sketch.

Algorithm for Enrolment

Feature Extraction. We use the method described in Algorithm 3 (Section II.1), which consists in pre-alignment[2], computation of real components using the directional field and the Gabor responses, and quantization.

The reliable bit selection (Algorithm 2) outputs a subset $W_i \subset [\![1, n]\!]$ of the components to be selected for a user $\mathcal{U}_i$. We keep this subset, re-noted $P_{1,i}$ as part of the Sketch.

[2] Note that here this pre-alignment was done manually for all the database to simplify the experiment.

Cancelable Transformation. The cancelable transformation we choose (on binary templates) is actually pretty simple. For $i \in [\![1, N]\!]$, a random permutation $\sigma_i \in \Sigma_n$ of $[\![1, n]\!]$ is chosen and we apply them on the database to obtain the transformed database containing new vectors $(Y_i)_{i=1..N}$ where for all i we set

$$\forall k \in \{1, \ldots, n\},\ (Y_i)^{(k)} = (X_i)^{(\sigma_i(k))},$$

which means that we apply the transformations f_i on all templates of user i to construct cancelable templates Y_i. These transformations are stored either by a server or by the relating users for future verifications. Here we will consider them as secrets.

Sketching The code-offset construction is applied with a binary product code C of length n. Let c_i be a random codeword of C and compute $P_{2,i} = c_i \oplus Y_i$. The data $(i, P_{1,i}, P_{2,i}, H(c_i))$ are stored in a database, where H is a given cryptographic hash.

Algorithm for Verification

When a user $\mathcal{U}_i$ wants to authenticate himself, a new fingerprint image is captured and a real vector Z_i of length n is extracted, once again with some bits of information and some erasures, using the reliable positions in $P_{1,i}$. As it is possible for the fresh fingerprint and the enrolled data not to be aligned, it is likely that more erasures be present in Z_i than in the stored data $P_{2,i}$.

Cancelable transformation. In order to compute the cancelable representation of Z_i, the transformation σ_i is recovered from its storage location – e.g. a server or the user's token – then we construct T_i as $(T_i)^{(k)} = (Z_i)^{(\sigma_i(k))}$ for all $k \in [\![1, n]\!]$.

Recovery and verification. $P_{2,i} \oplus T_i = c_i \oplus (T_i \oplus Y_i)$ is computed and the min-sum decoding algorithm of Section IV.1 is run to recover a message c'_i. One nice feature is that it enables efficient decoding of errors and erasures at the same time. Finally, we compare the value $H(c'_i)$ with the stored value $H(c_i)$.

Discussion

Note that here the use of secret permutations of $[\![1, n]\!]$ to transform the extracted features fulfils the condition of cancelability. It is clear that, with a high probability, it allows to match neither a data x with a transformed version $f(x)$ nor two transformed versions $f_1(x)$ and $f_2(x)$ together. Even if they are not irreversible functions by construction, they are computationally irreversible thanks to their secrecy and randomness.

Following an observation of [147], for each individual, we assume that a new random permutation is assigned at each enrolment. Hence, due to the large number of possibilities, a given permutation will only, with an overwhelming probability for n large (e.g. $n > 500$), be used once during the system's life (for all users together). So that, given a transformed template $\sigma(x)$, there is no other available information on σ which would have permit to interpolate σ and to recover x. Moreover, it implies that an adversary could distinguish $(x_1, \sigma(x_1))$ from $(x_2, \sigma(x_1))$ only with a negligible probability when x_1 and x_2 have the same binary weight.

Of course, a truly irreversible function would be preferable than a secret one for some applications but we think that the results achieved below worth to consider this slight constraint.

Results

To follow the cancelable biometrics configuration, all the results are always computed by assuming that the right transformation is used in verification; *i.e.* that when the verification involves the reference data of a user i, then the new template is always[3] transformed via σ_i, even if it concerns a non-legitimate user $j \neq i$.

We choose randomly $M = 6$ images per user for enrolment and the 2 remaining for the verification. We construct binary templates of length $n = 2048$ and we consider the $[2048, 42, 512]_2$ product code $C = RM(1,6) \otimes RM(1,5)$. It yields a FR rate of 3% and a FA rate of 5.53%. With respect to the performances announced in [176, Fig. 5] (for comparable FR rate or FA rate), it compares favourably to the results obtained with the $[511, 67, 175]_2$ BCH code, 5.2% of FR for 5.5% of FA, and it is even sligthly better than the 3.4% of FR and 6.1% of FA given by the restriction to the 87 most reliable users.

We also check the Hamming distance distribution to evaluate the performances of a Hamming distance classifier, which has the same effect of a BCH decoder, by adding the number of errors to half the number of erasures. For similar rates, we need a very large threshold: with a threshold of 0.4×2048 it gives a FR rate of 3% for a FA rate of 6.40%. First, it means that we can not achieve these performances with a BCH code with the same dimension: for the length 2048 and a capacity of correction of 0.4×2048, the dimension must be smaller than 2 thanks to the Plotkin bound – cf. [115]. It also underlines that the min-sum algorithm helps to improve the performances.

Note that, even if here the dimension of the code seems quite small, it has the merit to prove that to include cancelable biometrics into secure sketches still permits to have a good discrimination between matching fingerprints and non-matching fingerprints.

At last, we underline that these error rates improvement are reported when biometric templates are binarized and the similarity measure uses Hamming

[3] Otherwise, with the wrong transformations σ_j, the FA rate would be almost 0%.

distance whereas, without any quantization, biometric templates such as fingerprints can be compared with more efficient matching mechanisms. For instance, as explained in [176], a likelihood ratio-based algorithm would yield here an EER of 1.4%.

Remark 18. To conclude this Section, we wish to emphasize once more that the security of this construction is not based on the use of a Secure Sketch. It is based on the use of a permutation $\sigma \in \Sigma_n$, which is supposed to be secret. The Secure Sketch enables to use a coding algorithm to match two biometric templates, but leaks a huge deal of information on the transformed template Y_i; we refer the reader to [166] for details on the leak.

This first example underlines the interest and the feasibility of the technique. In our opinion, it will help to improve the security of biometric data.

It is also thinkable to add a physical layer of protection by embedding an enrolled template and the matching algorithm in a smart card, as in general computations rely on a decoding algorithm.

IV.3 Time-Dependant Cancelable Biometrics

In this section, we go one step further from classical biometric identification: we want to identify people, yet preventing the system from tracking them. We are thus looking at the paradoxical functionality of *anonymous identification*, as published in [32].

Cancelable biometrics alone do not suffice to thwart an adversary who tries to locate an user by determining if the same distorted biometric data are involved many times (in this case, the adversary can not find who is trying to be identified, but can say whether it is the same person). Our proposal borrows to One-Time Passwords [86] the supplementary idea of time-dependent distortions. We introduce here Time-Dependent Cancelable Biometrics.

To the best of our knowledge, the term of One-Time Biometric Authentication first appears in [178]. [110] describes a scheme with One-Time Templates for Face Authentication. Before, in [77, 94, 190], a token is used to randomize cancelable biometric data. Our work differs as we want to keep the ability of biometric data to identify people directly with their biometric characteristics alone (rather than authenticate them with the help of an extra token). Furthermore, we place ourselves at a system builder level who exploits generic properties coming with cancelable biometrics. We benefit from the security of randomizing biometrics and from the resistance to replay attacks obtained as a side effect of making data and functions vary across time.

System Entities

Formally, the system components are:

- Human user U, who uses his biometrics to identify himself to a server.

- Sensor clients $\mathcal{C}$, which extract human user's biometric template using a biometric sensor and which communicate with the server and the user.
- Server $\mathcal{S}$, which deals with human user's identification. It has access to a database $\mathcal{DB}$ to run the identification process. The Database $\mathcal{DB}$ stores templates which are computed from the biometric information obtained during the enrolment of users. We do not want the server to have to deal with biometric data which can help to determine who is identifying himself.

Assumptions on the Sensor client $\mathcal{C}$. Our sensors implement liveness detection to check that they are dealing with a living person during their identification requests. Examples of such anti-spoofing functions can be found in [156]. These functions are protected by physical means inside $\mathcal{C}$. More precisely, on intrusion, all the erasable memories of $\mathcal{C}$ are deleted. Furthermore, to each sensor is associated an unique serial number ID and a symmetric key K_{ID}. We put in $\mathcal{C}$ a tamper resistant element to perform all the computations made with the key K_{ID}. In case of compromise, the Sensor client $\mathcal{C}$ and its security element do not continue to work.

Assumptions on the Server $\mathcal{S}$. We also consider that the server possesses an Hardware Security Module (HSM) [5] to make all its cryptographic computations. In particular, this HSM can compute from a master key and a sensor ID, the corresponding key K_{ID}. This shared key is then used to perform a mutual authentication, and determine a session key between $\mathcal{C}$ and $\mathcal{S}$. At the end of this authentication, a secured link is established [13].

Assumptions on the link between $\mathcal{C}$ and $\mathcal{S}$.

1. Our secured link will be used (see below) to send new distortion functions from $\mathcal{S}$ to $\mathcal{C}$ and get back the distorted template just captured by the sensor. These communications are encrypted and authenticated by the session key of $\mathcal{C}$ and $\mathcal{S}$.
2. We assume that the server $\mathcal{S}$ and the sensor $\mathcal{C}$ are linked together in a way that enables them to stay time-synchronized.

Remark 19. An adversary is able to steal a distortion function by sacrificing a sensor (after compromise, the mutual authentication step fails).

Assumptions on the server $\mathcal{S}$ and its database $\mathcal{DB}$.

1. The interactions between $\mathcal{S}$ and $\mathcal{DB}$ are as follows. $\mathcal{S}$ sends a distortion function g_t and a distorted biometric template α. The database returns $\sup_{\delta \in \mathcal{DB}} m(\alpha, g_t(\delta))$, *i.e.* the database replies 1 if α matches an element of $g_t(\mathcal{DB})$.

2. The server and the database are "honest-but-curious", which means that they will follow the protocol. Our aim is to limit the amount of private information that $\mathcal{S}$ and $\mathcal{DB}$ can gather.

Properties of our Families of Distortions

We do not want users to need an additional element to identify themselves. This implies that the distortions we apply to obtain cancelable biometric templates are implemented into sensors.

A first distortion f is applied to biometric data at the enrolment phase. The Database $\mathcal{DB}$ stores $f(b)$ for all users U. This ensures that an adversary with an access to the Database $\mathcal{DB}$ cannot determine who is identifying himself.

During an identification, we maintain the untraceability of the data sent to the server thanks to a time-dependent family of cancelable transformations g_t.

To mitigate the risk of compromise of a sensor, we store inside $\mathcal{C}$ a distortion which only depends on the sensor. This means that if an adversary has access to this specific distortion, he cannot apply it elsewhere. To this end, we compute $g_t = g_{t,out} \circ g_{t,in}$ as the composition of two cancelable transformations. Let $h_{t,ID}$ be a family of time-dependent bijective functions of the template domain, then

$$g_t = (g_{t,out} \circ h_{t,ID}) \circ (h_{t,ID}^{-1} \circ g_{t,in})$$

splits in two parts the computation of g_t in a way which depends on the sensor. Only one part, composed with f, $h_{t,ID}^{-1} \circ g_{t,in} \circ f$ is stored in a sensor avoiding the compromise of g_t if the memory of a sensor is read out by an attacker.

Remark 20. At a given time, the distortion function g_t is independent of the sensor. In fact, all the sensor functions $h_{t,ID}^{-1} \circ g_{t,in} \circ f$ are different and their counterpart $g_{t,out} \circ h_{t,ID}$ are implemented in the server. That means, that when $\mathcal{S}$ is dealing with a sensor which sends it $\beta = h_{t,ID}^{-1} \circ g_{t,in} \circ f(b')$ where b' is a freshly captured template, $\mathcal{S}$ terminates the computation with $g_{t,out} \circ h_{t,ID}(\beta) = g_t \circ f(b')$. Further details are given below.

Our Proposal

At the enrolment phase of user U, his distorted biometric template $f(b)$ is stored into $\mathcal{DB}$.

When U wants to identify himself on sensor $\mathcal{C}$, $\mathcal{C}$ has to retrieve the distortion of the present time. As stated before, this happens after $\mathcal{C}$ and $\mathcal{S}$ have authenticated themselves. One part of the distortion $g_t \circ f = \phi_{out} \circ \phi_{in}$, ϕ_{in} is sent to $\mathcal{C}$ and the other part ϕ_{out} is kept on the server side (see Remark 20 for the exact definitions of ϕ_{out}, ϕ_{in}). b' from U is captured by $\mathcal{C}$ which sends $\beta = \phi_{in}(b')$ to $\mathcal{S}$. $\mathcal{S}$ then tries to identify U by matching $\phi_{out}(\beta)$ against $g_t(\mathcal{DB})$.

Security Analysis

Additional Security Properties on Distortion Functions

We have to extend the security properties on the distortion functions to cover their use in our protocol.

The most important new property we need is that the composition of several cancelable distortions must still verify conditions of Section III.3. This is easy to achieve. The distortion functions detailed in Section III.3 complies with our new requirement on composition. For instance, the composition of two cartesian transformations is still a cartesian transformation. The same holds for polar and functional transformations.

Regarding time-dependent distortion functions g_t, the diversity property (see Condition 4: **Transformation function design** of cancelable transformations) ensures us that if we choose the time-dependent g_t's as functions of the same family independent among themselves, we can not match together distorted templates $g_t(b_t), g_{t'}(b_{t'})$ computed at different times t, t', even if they come from the same user.

Finally, the knowledge of different functions ϕ_{in} implemented in sensors which correspond to the composition of a cancelable distortion with a bijective mapping must not reveal too much information on the underlying cancelable distortion.

Untraceability

As stated in Section IV.3, communications between sensors and the server $\mathcal{S}$ are encrypted. Consequently, an adversary is necessarily an insider. The classical definition of untraceability is recalled now:

Definition 5. Let $I = \{g_1(b_1), \ldots, g_n(b_n)\}$ be a set of time-differed identification requests. Let $\mathcal{A}$ be a probabilistic polynomial time algorithm that take such an I as input and outputs two different $i, j \in [\![1, n]\!]$; $\mathcal{A}$ is successful if b_i and b_j come from the same user. The identification procedure is *untraceable* if, for all such $\mathcal{A}$, the probability of success of $\mathcal{A}$ is negligibly close to that of the random sampling without replacement $i, j \in [\![1, n]\!]$.

The interface between $\mathcal{S}$ and its database $\mathcal{DB}$ is restricted and permits $\mathcal{S}$ to only know whether some biometric template matches some element present in $\mathcal{DB}$. We rely on the previous section and on the new property we impose on time-dependent functions g_t to achieve untraceability. In fact, in our case, definition 5 leads to "from different $g_{t_0}(b_{t_0})$, $g_{t_1}(b_{t_1})$, determine $m(b_{t_0}, b_{t_1})$". This is precisely what we protect with our time-dependent functions.

Remark 21 (Concluding Remark). This new identification scheme, which is based on cancelable biometrics, relies on realistic assumptions concerning sensors, server and database.

As opposed to association of cancelable biometrics with secure sketches, this construction does not impose to users an extra token, and allows identification instead of authentication.

An additional advantage of this construction is the untraceability of the users from the point of view of an eavesdropper that would not have access to the server. This notion of *Time Dependant Cancelable Biometrics* raises new challenges, in particular, the iterative application of different cancelable functions. This is a topic that is not studied in the literature, and the error rates that are to be expected in such a setting are unknown.

Conclusion

This chapter provides three contributions we made to the topic of "classical" template protection. A very interesting coding structure for biometrics, along with the associated decoding algorithm is presented in Section IV.1. An association of secure sketches with cancelable biometrics, which combines both advantages, is described in Section IV.2; finally, using only cancelable biometrics, and with pretty strong hypotheses on the functions underneath, we get an "anonymous identification" protocol in Section IV.3.

The next chapters will have a significantly different approach, as we shall try and use modern cryptographic primitives as building blocs for the protection of the biometric templates, as well as the users' privacy.

Part 3

Cryptography Applied to Biometric Data

Chapter V

The Asymmetric Case

> The difference between the almost right word and the right word is really a large matter – it's the difference between the lightning bug and the lightning.
>
> Mark Twain

In this chapter we focus on designing a construction that uses cryptographic functions as the effective way to protect the data. Etymologically speaking, cryptography *is* the science of hiding data from an adversary. This discipline is now a widely studied subject by a large community of researchers. The cryptographic primitives that are regularly proposed are therefore crypt-analysed *i.e.* attacked, in order to validate their robustness.

The use of cryptographic functions is thus a first guarantee that a scheme does not reveal too much information, under computational assumptions, such as the hypothesis that it is difficult to decrypt the output of an AES encryption with no knowledge of the key. The following part will therefore contain security "proofs". These state that, under the hypothesis that a given problem is difficult, a certain security level can be achieved.

The security concepts that we wish to gain depends on the application and its context, and can therefore not be defined once and for all. However, standard properties will be used, such as the *indistinguishability under chosen-plaintext attack* (IND-CPA). Elements on these cryptographic notions are given in Appendix C.

V.1 Secure Querying and Secure Authentication

We recall here our goal: to have a biometric authentication or identification system, in which the stored data are encrypted, and not decrypted.

The ideal situation consists in having a database, where all the elements are encrypted, and where the elements are never decrypted - actually, the matching are made in the encrypted domain, and even the results are encrypted. An attacker who would gain access to the system without knowing the key would gain no information on the stored elements.

As we will see in more details, this would indeed provide interesting properties, but would not answer all privacy issues. In particular, we will take interest in the cases where the database administrator is *honest-but-curious*. This means that the privacy issues will not be entirely solved by the cryptographic elements.

This section will then present the state of the art on secure biometric authentication. This is inspiring material to build a secure biometric identification.

Privacy Requirements for a Secure Biometric Authentication

The secure biometric authentication problem has been the subject of increasing interest this past decade, and has proven to be difficult. Using a coherent security model to this primitive is nevertheless recent, and no standard privacy property is used in the whole community. We therefore propose the following model and assumptions, widely inspired by [173]. The different threats to a biometric systems have been investigated in the literature, and formalized by, *inter alia*, [18, 85].

We focus on these two possible threats:

- An adversary uses the system to be authenticated in place of the genuine user (identity theft).
- The information exchanged with the system is clear enough for an "honest-but-curious" database to get more information than required (privacy leakage).

Note that this second point can be declined in two versions: the owner of the database can learn information on the user that gets authenticated, or it can learn information on his biometric template. Both cases are undesirable.

To prevent these threats, the following measures need to be taken:

- Design the scheme in such a way that two distant templates are also recognised as distant templates. In this case, the False Accept Rate should not be too high, and an impostor could not get easily authenticated.
- Apply encryption to the templates at the sensor level. This means that at the database level, it should not be possible to get information on the templates.

- Use classical methods to anonymize the queries made to the database.

The first point is the subject of an entire research subject, which is to find a satisfactory encoding of the templates. In order to achieve the second point, we need to be able to make some computations in the encrypted domain. This means that not every cipher can be used to encrypt the templates, as it is necessary to use *homomorphic* encryption. The overall goal is to represent the templates as binary vectors, on which the matching operation can be done by simple bits operation, so that these operations can be made in the encrypted domain.

The third point is necessary for the privacy of the users. If it is not implemented, the service provider is able to trace a user in each of his requests. The situation imagined in [173] is in fact the case where the matching was made in the database, in the encrypted domain, and then the result is privately retrieved, in the way that we shall see here.

Private Information Retrieval Protocols

In 1998, Chor *et al.* [51] proposed a primitive to retrieve a bit from an array in such a way that the owner of the array does not know which bit was retrieved. This is what is called a *Private Information Retrieval* (**PIR**) protocol.

The direct extension of a PIR is a Private Bloc Retrieval protocol. It is a protocol between two parties Alice and Bob. Bob owns an array of n entries, m bits each. Alice wants to know the content of the i^{th}entry of the array; the security property that is required is that Bob is oblivious to i.

One obvious way to achieve such a protocol is for Bob to send his whole array to Alice, but this is hardly efficient in terms of communication (overall communication cost: $m \cdot n$ bits). Therefore Bob must do operations on his array depending on the query Alice made, and in order for him to be completely unaware of i, it is necessary to do these operations on each entry of the database. In other terms, one of the concern of PIR is the communication cost, which should be sublinear in n, and the other concern is the computing cost, which is at least linear in n for non-trivial protocols. Both aspects make it a costly primitive.

Existing Constructions

There have been several propositions for PIR (or, by extension, Private Bloc Retrieval) protocols, as it was surveyed in [135]. Such constructions often make one of the following two assumptions:

- There are multiple databases, and their owners do not collude;

- There is only one database, and the security is based on a computational assumption.

The first case is of no practical implication for us; for the second scenario, the results in the litterature are promising, with a protocol [113] of communication cost of only $O(m \log n + \ell \log^2 n)$ where ℓ is the security parameter (in bits). This protocol uses the homomorphic cryptosystem of Damgård and Jurik to process the database, producing a response of growing size to be sent to Alice.

1-out-of-n Oblivious Transfer

There is a close link between PIR protocols and another primitive called *Oblivious Transfer* (OT), in which Alice receives an element without Bob knowing which one was queried *and* Alice does not get more information than the value of the queried element. This is why this primitive is also called a *Symmetric Private Information Retrieval* protocol (SPIR).

Such a primitive prevents the trivial PIR protocol in which Bob sends the whole content of his array. It will be used when we need a scheme in which a memory cell belongs to one person only.

Secure Authentication

In the last five years, several biometrics *authentication* protocols, *e.g.* [36, 37, 39, 157, 173, 181], have proposed to embed the matching directly. They use the property of homomorphic encryption schemes to compute the Hamming distance between two encrypted templates. Some other interesting solutions, based on adaptation of known cryptographic protocols, are also investigated in [25, 38].

In a few lines, [25] proposes to make the computation of the sketching and recovery operations in the encrypted domain using a homomorphic cryptosystem $(\mathsf{Enc}, \mathsf{Dec})$ (see Appendix C for definitions and examples of homomorphic encryption). As the key elements for the protection of the templates are the "exclusive or" (xor, $\oplus$), this is well captured by the homomorphic encryption. A smart combination of the Goldwasser-Micali [79] and Pailler [138] enable to make all the sensitive computations in the encrypted domain, and the decoding is made in a physically secure element.

The drawback with all these techniques is that they do not fit well with *identification* in large databases as the way to run an identification among L data would be to run as many authentication algorithms. As far as we know, no non-trivial protocol for biometric *identification* involving privacy and confidentiality features was proposed before [33, 34].

V.2 Secure Identification in the Asymmetric Setting

This section describes results published in [33, 34]. [33] describes a theoretical framework for what we call Error-Tolerant Searchable Encryption. [34] provides a direct application of the scheme to biometric data, and shows in more details how the theoretical framework and the application are intertwined. [34] also provides a construction for an even more private scheme than the others.

Identification and Nearest Neighbour

Identification consists in finding the best match in a set; when the match can be expressed in terms of a distance, then the problem can be rephrased into a more classical one, namely *the search for the nearest neighbour.*

Several algorithms have been proposed for the so-called *Nearest Neighbour* and *Approximate Nearest Neighbour* (**ANN**) problems. The definitions of each of these problems follow, and we refer to Indyk's review [92] on these topics for more details.

Problem 1 (Nearest Neighbour)**.** Given a set P of points in the metric space (B, d), pre-process P to efficiently answer queries. The answer of a query x is a point $p_x \in P$ such that $d(x, p_x) = \min_{p \in P} d(x, p)$.

Solving this problem in the general case is unfortunately impossible in reasonable time and memory: that is called *the curse of dimensionality.* If the dimension of the space B is high, then no solution with a "reasonable" preprocessing complexity and a low query cost has been discovered.

For that reason, a relaxation of the previous problem was proposed[1], namely:

Problem 2 (ϵ-Approximate Nearest Neighbour)**.** Given a set P of points in the metric space (B, d), pre-process P to efficiently answer queries. The answer of a query x is a point $p_x \in P$ such that $d(x, p_x) \leq (1+\epsilon) \min_{p \in P} d(x, p)$.

Note that our problematic – biometric identification over encrypted data – can use the solutions for the ANN problem, but that these are not enough, as we need to add a security layer to such protocols. For example, Hao *et al.* [88] demonstrated the efficiency of the **ANN** approach for iris biometrics where projected values of iris templates are used to speed up identification requests over a large database; indeed [88] derived a specific **ANN** algorithm from the iris structure and statistical properties. However, in their construction the iris biometric data are never encrypted, and the way they boost the search for the nearest match reveals a large amount of information about sensitive data. We here add the required cryptographic protection that provides an answer to the privacy issues of the neighbour search.

As a direct consequence of this model, our works are also influenced by the problem of finding a match on encrypted data. Boneh *et al.* defined the notion of *Public-key encryption with Keyword Search* (**PEKS**) [19], in which specific

[1] In the same way that the γ-shortest vector problem was proposed for euclidean lattices.

trapdoors are created for the lookup of keywords over public-key encrypted messages. Several publications, among which [16, 43, 78, 101, 153], have also elaborated solutions in this field.

However the main difference between the search for a keyword as understood by Boneh *et al.* [19, 20] and biometric matching is that an exact match for a given bit string in the plaintext suffices for the former, but not for the problem stated. For this purpose, we introduce a new model for error-tolerant search in section V.2 and specific functions to take into account fuzziness in section V.2.

Remark 22. The most significant difference here from the primitives introduced previously in [19] is that messages are no longer associated to keywords. Moreover, our primitives enable some imprecision on the message that is looked up. For example, one can imagine a mailing application, where all the mails are encrypted, and where it is possible to make queries on the mail subject. If there is a typo in the query, then looking for the correct word should also give the mail among the results – at least, we would like that to happen. Note that wildcards[2] are not well-adapted to this kind of application, as a wildcard permits to catch errors providing that we know where it is located; here, we wish to be able to find a match even if we do not know where the errors are likely to happen.

After recalling the notions of locality-sensitive hashing and Bloom filters, we introduce a new structure that enables approximate searching by combining both notions.

Locality-Sensitive Hashing

Most algorithms proposed to solve the ANN problem consider real spaces over the l_p distance, which is not relevant in our case. A way to search the approximate nearest neighbour in a Hamming space is to use a generic construction called locality-sensitive hashing. It looks for hash functions[3] that give the same result for near points, as defined in [93]:

Definition 6 (Locality-Sensitive Hashing [93])**.** Let (B, d_B) be a metric space, U a set of smaller dimensionality. Let $r_1, r_2 \in \mathbb{R}$, $p_1, p_2 \in [0, 1]$ such that $p_1 > p_2$.

A family $H = \{h_1, \ldots, h_\mu\}$, $h_i : B \to U$ is (r_1, r_2, p_1, p_2)-**LSH**, if

$$\forall h \in H,\ x, x' \in B \left\{ \begin{array}{l} Pr[h(x) = h(x') \,|\, d_B(x, x') < r_1] > p_1 \\ Pr[h(x) = h(x') \,|\, d_B(x, x') > r_2] < p_2 \end{array} \right.$$

Such functions reduce the differences occurring between similar data with high probability, whereas distant data should remain significantly remote.

[2] A wildcard $\star$ enables to catch erasures as defined in III.5.

[3] These hashes have no cryptographic property.

A noticeable example of a LSH family was proposed by Kushilevitz *et al.* in [109]; we briefly describe these functions for the sake of completeness. For more LSH families, see also [106, 93, 6].

These functions are based on a parameter $\beta \in [0,1]$. Let $r \in B$ be a binary vector; the associated projection is noted $\phi_r : x \mapsto \phi_r(x) = \sum_{i=1}^{n} x_i r_i$; all computations are made modulo 2. Any set of t chosen vectors $r^1, \ldots, r^t$ gives a hash function $h = (\phi_{r^1}, \ldots, \phi_{r^t}) : B \longrightarrow \{0,1\}^t$.

To design a LSH function, we pick $r^1, \ldots, r^t \in \{0,1\}^n$ such that for all $i \in [\![1,t]\!]$, for all $j \in [\![1,n]\!]$, $\Pr\left[r_i^j = 1\right] = \beta$ where $r_i = (r_i^1, \ldots, r_i^n)$.

The following lemma, proved in [109], explains the LSH property:

Lemma V.1. *Let $x \in B$, $r^1, \ldots, r^t \in B$ random vectors such that each bit have been picked randomly with probability $\frac{1}{2l}$.*

There exists $\delta_1 > 0$ such that for all $\epsilon > 0$, $a, b \in B$ two points such that $d_B(x,a) \leq l$ and $d_B(x,b) > (1+\epsilon)l$, there exists a constant $\delta_2 = \delta_1 + \delta$ (with $\delta > 0$) depending only on ϵ for which:

$$\begin{aligned} \Pr\left[d(h(x),h(a)) > (2\delta_1 + \delta_2)t/3\right] \leq & \quad e^{-\frac{2}{9}\delta^2 t} \\ \Pr\left[d(h(x),h(b)) < (2\delta_2 + \delta_1)t/3\right] \leq & \quad e^{-\frac{2}{9}\delta^2 t} \end{aligned}$$

where d is the Hamming distance over $\{0,1\}^t$.

More than this specific construction, keep in mind that LSH families *exist*; this construction will be most useful in the following chapters.

Bloom Filters

As introduced by Bloom in [17], a set of Bloom filters is a data structure used for answering set membership queries.

Definition 7. Let D be a finite subset of Y. For a collection of ν (independent) hash functions $H' = \{h'_1, \ldots, h'_\nu\}$, with each $h'_i : Y \to [\![1,M]\!]$, the induced (ν, M)-*Bloom filter* is H', together with an array $(t_1, \ldots, t_M) \in \{0,1\}^M$, defined as:

$$t_\alpha = \begin{cases} 1 \text{ if } \exists i \in [\![1,\nu]\!], y \in D \text{ s.t. } h'_i(y) = \alpha \\ 0 \text{ otherwise} \end{cases}$$

With this setting, testing if y is in D is the same as checking if for all $i \in [\![1,\nu]\!], t_{h'_i(y)} = 1$. The best setting for the filter happen when the involved hash function is as randomized as possible, in order to uniformly fill all the buckets t_α.

In this setting, some false positive may happen, *i.e.* it is possible for all $t_{h'_i(y)}$ to be set to 1 and $y \notin D$. This event is well known, and if the functions are balanced, the probability for a query to be a false positive is:

$$\left(1 - \left(1 - \frac{\nu}{M}\right)^{|D|}\right)^{\nu}. \qquad \text{(V.2.1)}$$

This probability can be made as small as needed by tuning ν and m for a fixed D. On the other hand, no false negative is enabled.

We work here with the *Bloom filters with storage* (**BFS**) defined in [20] as an extension of Bloom filters. Their aim is to give not only the result of the set membership test, but also an index associated to the element. The iterative definition below introduces these objects and the notion of *tags* and *buckets* which are used in the construction.

Definition 8 (Bloom Filter with Storage, [20])**.** Let D be a finite subset of a set Y. For a collection of ν hash functions $H' = \{h'_1, \ldots, h'_\nu\}$, with each $h'_j : Y \to [\![1, M]\!]$, a set V of *tags* associated to D with a *tagging* function $\psi : D \to \mathcal{P}(V)$, a (ν, M)-*Bloom Filter with Storage* is H', together with an array of subsets $(T_1, \ldots, T_M)$ of V, called *buckets*, iteratively defined as:

1. $\forall i \in [\![1, M]\!], T_i \leftarrow \emptyset$,
2. $\forall y \in D, \forall j \in [\![1, \nu]\!]$, update the bucket T_α with $T_\alpha \leftarrow T_\alpha \cup \psi(y)$ where $\alpha = h'_j(y)$.

In other words, the bucket structure is empty at first, and for each element $y \in D$ to be indexed, we add to the bucket T_α all the tags associated to y. Construction of such a structure is illustrated in Fig. V.1.

Remark 23. Another definition of the Bloom Filter with Storage can be obtained with the following equation, which sums up the iterative construction:

$$T_\alpha = \bigcup_{j=1}^{\nu} \bigcup_{y \in Y_\alpha^j} \psi(y),$$

with $Y_\alpha^j = \left\{ y \in D \text{ s.t. } h'_j(y) = \alpha \right\}$. Another formulation is:

$$T_i = \bigcup_{y \in D : \exists j \in [\![1,\nu]\!] : h'_j(y) = i} \psi(y). \tag{V.2.2}$$

Example 1. In Figure V.1, assume that $D = \{y_1, y_2, y_3\}$ and $\nu = 3$, the tags associated to y_1 (resp. y_2) have already been incorporated into the buckets T_2, T_3 and T_α (resp. T_1, T_2 and T_3) so that $T_1 = \{\psi(y_2)\}$, $T_2 = T_3 = \{\psi(y_1), \psi(y_2)\}$, $T_\alpha = \{\psi(y_1)\}$ and $T_i = \emptyset$ otherwise. We are now treating the case of y_3:

- $h'_1(y_3) = \alpha$ so $T_\alpha \leftarrow T_\alpha \cup \{\psi(y_3)\}$, *i.e.* $T_\alpha = \{\psi(y_1), \psi(y_3)\}$;
- $h'_2(y_3) = 2$ so $T_2 \leftarrow T_2 \cup \{\psi(y_3)\}$,*i.e.* $T_2 = \{\psi(y_1), \psi(y_2), \psi(y_3)\}$;
- $h'_3(y_3) = M$ so $T_M \leftarrow T_M \cup \{\psi(y_3)\}$, *i.e.* $T_M = \{\psi(y_3)\}$.

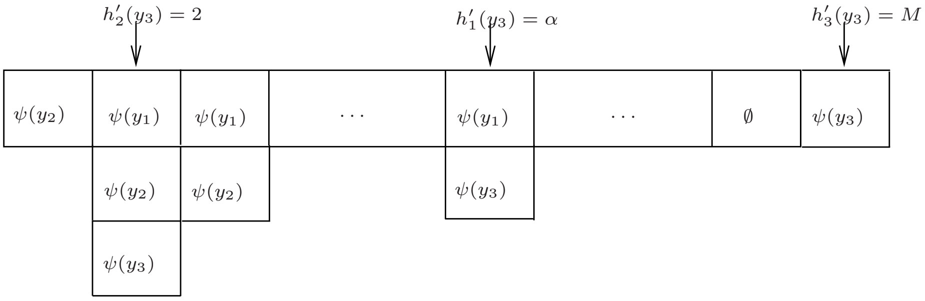


Figure V.1: Construction of Bloom Filters with Storage

This construction is designed to obtain $\psi(y)$, the set of tags associated to y, by computing $\bigcap_{j=1}^{\nu} T_{h'_j(y)}$. For instance, in the previous example,

$$\bigcap_{j=1}^{\nu} T_{h'_j(y_3)} = T_2 \cap T_\alpha \cap T_M = \{\psi(y_3)\}.$$

This intersection may capture inappropriate tags, but the choice of relevant hash functions and increasing their number allow to reduce the probability of that event. These properties are summed up in the following lemma.

Lemma V.2. *Let $(H', T_1, \ldots, T_M)$ be a (ν, M)-Bloom filter with storage indexing a set D with tags from a tag set V. Then, for $y \in D$, the following properties hold:*

- $\psi(y) \subset T(y) := \bigcap_{j=1}^{\nu} T_{h'_j(y)}$, *i.e. each of y's tag is retrieved,*
- *the probability for a false positive $t \in V$ is* $\Pr[t \in T(y)$ *and* $t \notin \psi(y)] = \left(1 - \left(1 - \frac{\nu}{M}\right)^{|D|}\right)^{\nu}$.

Proof The first part of the lemma is straightforward.

The second part is deduced from Equation V.2.1, by the simple transformation that maps a Bloom filter with storage into a classical Bloom filter. Let $t \in V$, define the application $\Phi_t : \mathcal{P}(V) \to \{0, 1\}$ by $\Phi_t(A) = 1$ if and only if $t \in A$. It then appears that $\Pr[t \in T(y)$ and $t \notin \psi(y)]$ is the precise probability of a false accept of t in the classical Bloom filter obtained by projecting $(T_1, \ldots, T_M)$ into $\{0, 1\}^M$. □

Combining BFS and LSH

We want to apply Bloom filters to data that are very likely to vary. The following section shows how to apply LSH-families as input to Bloom filters. This approach differs from [106] in which the Bloom filters use as binning functions some distance-sensitive hash functions, resulting in asymptotically bad error probabilities.

We choose μ hash functions from an adequate LSH family $h_1, \dots, h_\mu : B \to \{0,1\}^t$, and ν hash functions dedicated to a Bloom filter with Storage $h'_1, \dots, h'_\nu : \{0,1\}^t \times \llbracket 1, \mu \rrbracket \to \llbracket 1, M \rrbracket$. The LSH family is denoted H, and H' is the BFS one. To obtain a BFS with locality-sensitive functionality, we use composite $\mu \times \nu$ hash functions induced by both families.

We define $h^c_{(i,j)} : B \to \llbracket 1, M \rrbracket$ the corresponding composite functions (c stands for composite) with $h^c_{(i,j)}(y) = h'_j(h_i(y), i)$. Let $H^c = \{h^c_{(i,j)}, (i,j) \in \llbracket 1, \mu \rrbracket \times \llbracket 1, \nu \rrbracket\}$ the set of all these functions.

Algorithm 4 Combination of Bloom filter with storage with locality-sensitive hash functions

```
On input:
  • $H = \{h_1, \dots, h_\mu\}$ a $(\lambda_{min}, \lambda_{max}, \epsilon_1, \epsilon_2)-$LSH family from $B$ to $\{0,1\}^t$
  • $H' = \{h'_1, \dots, h'_\nu)$ a set of hash functions dedicated to a BFS, from $\{0,1\}^t \times \llbracket 1, \mu \rrbracket$ to $\llbracket 1, M \rrbracket$
On output: $H^c$ a family of $\nu \cdot \mu$ functions from $B$ to $\llbracket 1, M \rrbracket$
  • For $i \in \llbracket 1, \mu \rrbracket$,
  • For $j \in \llbracket 1, \nu \rrbracket$,
      – Define $h^c_{(i,j)}$ by $h^c_{(i,j)}(y) = h'_j(h_i(y), i)$.
```

To sum up, we modify the update of the buckets in Def. 8 by $\alpha = h'_j(h_i(y), i)$. Later on, to recover tags related to an approximate query $x' \in B$, all we have to consider is $\bigcap_{j=1}^{\nu} \bigcap_{i=1}^{\mu} T_{h'_j(h_i(x'),i)}$. Indeed, if x and x' are close enough, then the LSH functions give the same results on x and x', effectively providing a Bloom filter with storage that has the LSH property. This property is numerically estimated in the following lemma:

Lemma V.3. *Let H, H', H^c be families constructed following Algorithm 4. Let $x, x' \in B$ be two binary vectors. Assume that H is $(\lambda_{min}, \lambda_{max}, \epsilon_1, \epsilon_2)$-LSH from B to $\{0,1\}^t$; assume that H' is a family of ν pseudo-random hash functions. If the tagging function ψ associates only one tag per element, then the following properties stand:*

1. *If x and x' are far enough apart, then $\psi(x')$ intersects all the buckets that index x with probability*

$$\Pr_{x'}\left[\psi(x') \subset \bigcap_{h^c \in H^c} T_{h^c(x)}, d(x,x') \geq \lambda_{max}\right] \leq \left(\epsilon_2 + (1-\epsilon_2)\frac{1}{M}\right)^{|H^c|},$$

2. *If x and x' are close enough, then $\psi(x')$ is in all the buckets that index x' except with probability*

$$\Pr_{x'}\left[\psi(x') \not\subset \bigcap_{h^c \in H^c} T_{h^c(x)} \text{ and } d(x,x') \leq \lambda_{min}\right] \leq 1-(1-\epsilon_1)^{|H^c|} .$$

Note that this lemma used the simplified hypothesis that $\forall x, |\psi(x)| = 1$: there is only one identifying tag per x. This has a direct application in section V.2. In practice, $\psi(x)$ can be a unique handle for x.

Proof The first part of the lemma expresses the fact that if $d(x,x') \geq \lambda_{max}$, due to the composition of a LSH function with a pseudorandom function, the collision probability is $\frac{1}{M}$. Indeed, if $h'_1(y_1) = h'_2(y_2)$, either $y_1 = y_2$ and $h'_1 = h'_2$, or two independent pseudo-random hash functions collide.

The probability for collision of two such hash functions is $\frac{1}{2^t}$ which is negligible in t.

In the other case, if $y_1 = y_2$, then

$$(h_{i_1}(x), i_1) = y_1 = y_2 = \left(h_{i_2}(x'), i_2\right) .$$

To these vectors to be the same, $i_1 = i_2$ and $h_{i_1}(x) = h_{i_2}(x')$, which happens with probability ϵ_2.

The second part of the lemma says that for each $h^c \in H^c$, $h^c(x)$ and $h^c(x')$ are the same with probability $1-\epsilon_1$. Combining the incremental construction of the T_i with this property gives the lemma. □

Construction Outline

We propose to use recent advances done in the fields of similarity searching and public-key cryptography. Our technique narrows our identification to a few candidates. In a further step, we complete it by fine-tuning the results in checking the remaining identities so that the identification request gets a definite answer, *i.e.* we apply a Secure Authentication scheme.

The first step is accomplished by combining Bloom filters with locality-sensitive hashing functions. Bloom filters enable to speed up the search for a specified keyword using a time-space trade-off. We use locality-sensitive hashing functions to speed the search for the (approximate-)nearest neighbour of an element in a reference set. Combining these primitives enables to efficiently use cryptographic methods on biometric templates, and to achieve error-tolerant searchable encryption.

Architecture and Model for Biometric Identification

In the following, we restrict ourselves to $\mathcal{B} = \{0,1\}^N$ equipped with the Hamming distance d. Two different templates b, b' from the same user $\mathcal{U}$ are with high probability at a Hamming distance $d(b,b') \leq \lambda_{min}$; measurements b_1, b_2 of different users $\mathcal{U}_1, \mathcal{U}_2$ are at a Hamming distance $d(b_1, b_2) > \lambda_{max}$. In this case, the matching score $m(b, b')$ is affine in the Hamming distance between b and b'.

Remark 24. As an example, IrisCodes (as described in section II.1) do fit this model, with $N = 2048$. The best matching algorithm is more precise than just an affine classifier, see Eq. (II.1.2), but the Hamming comparison performs fine enough for a first approximation.

A system is given by a reference data set $D \subset B$ and a identification function $\mathsf{id} : \mathcal{B} \to \mathcal{P}(D)$. On input b_{new}, the system outputs a subset C of D containing biometric templates $b_{ref} \in D$ such that the matching score between b_{new} and b_{ref} is small. This means that b_{new} and b_{ref} possibly corresponds to the same person. C is the emptyset $\emptyset$ if no such template can be found; the size of C depends on the accuracy of the system. With pseudo-identities (either real identities of people or pseudonyms) registered together with the reference templates in D, the set C gives a list of candidates for the pseudo-identity of the person associated to b_{new}.

The general idea for identification is to search for candidates among a database. As the database could be very large (often more than hundred of thousands templates), this search has to be very fast. That is why its first goal is to obtain a smaller set of candidates on which a final comparison with b_{new} is possible – via the matching algorithm – to strengthen the result.

Architecture

Our general model for biometric identification relies on the following entities:

- Human users $\mathcal{U}_i$: a set of L users are registered using a sample of their biometrics β_i and pseudo-identities ID_i, more human users $\mathcal{U}_j$ $(j > L)$ represent possible impostors with biometrics β_j.
- Sensor client $\mathcal{SC}$: a device that extracts the biometric template from β_i.
- Identity Provider $\mathcal{IP}$: replies to queries sent by $\mathcal{SC}$ by providing an identity,
- Database $\mathcal{DB}$: stores the biometric data.

Remark 25. Here the sensor client is a client that captures the raw image of a biometric data and extracts its characteristics to output a so-called biometric template. Consequently, we assume that the sensor client is always honest and trusted by all other components. Indeed, as biometrics are public information, additional credentials are always required to establish security links in order to prevent some well-known attacks (*e.g.* replay attacks) and to ensure that, with a high probability, the biometric template captured by the sensor and used in the system is from a living human user. In other words, we assume that it is difficult to produce a fake biometric template that can be accepted by the sensor. This implies that countermeasures are deployed against biometric forgery. This assumption is strong, but every biometric system can be subject to "spoof attacks"; these must be put aside formally in this model, and in practice using liveness detection devices [156].

In an identification system, we have two main services:

1. Enrolment registers users using their physiological characteristics (for a user $\mathcal{U}_i$, it requires a biometric sample $b_i \leftarrow \beta_i$ and its identity ID_i)

2. Identification answers to a request by returning a subset of the data that was registered

The enrolment service is run each time a new user has to be registered. Depending on the application, the identification service can output either the candidates' identity or their reference templates.

As protection against outsiders, such as eavesdroppers, can be achieved with classical cryptographic techniques, we consider that the only threat to such a system resides in the server. For this reason, we mean to essentially protect the data against insiders, *i.e.* people that have full possession of the server. In particular, we assume that no attacker is able to interfere with communications between the server and the sensor client.

Informal Objectives

We here formulate the properties we would like to achieve in order to meet good privacy standards.

Condition 5. When the biometric identification system is dealing with the identification of a template b coming from the registered user $\mathcal{U}_i$ with identity ID_i, it should return a subset containing a reference to (ID_i, b_i) except for a negligible probability.

Condition 6. When the system is dealing with the identification of a template b coming from an unregistered user, it should return the empty set $\emptyset$ except for a negligible probability.

We do not want a malicious database to be able to link an identity to a biometric template, nor to be able to make relations between different identities.

Condition 7. The database $\mathcal{DB}$ should not be able to distinguish two enrolled biometric data.

Another desired property is the fact that the database knows nothing of the identity of the user who goes through the identification process. More precisely, we do not want the database to be able to link together different identifications of the same user at different times.

Condition 8. The database $\mathcal{DB}$ should not be able to guess which identification request is executed.

Security Model for Error-Tolerant Searchable Encryption

We now give the formal model for Error-Tolerant Searchable Encryption, as published in [33]. This scheme enables to approximately search and retrieve a message stored in a database, i.e. with some error-tolerance on the request. A specific construction fitting this model follows.

Remark 26. The questions of search in a database with error-tolerance is a problem quite close to biometric identification and the corresponding cryptographic primitives are thus used in our system, *cf.* section V.3. We however emphasize that this section provides a cryptographic tool that will be applied to our concerns later on, but does not require a biometric background to be defined.

Entities for the Protocol

Our primitive models the interactions between users that store and retrieve information, and a remote server. We distinguish the user who stores the data from the one who wants to get it. This leads to three entities:

- The server S: a remote storage system. The content and the communications with this server are considered public.
- The sender $\mathcal{X}$ incrementally creates the database, by sending data to S,
- The receiver $\mathcal{Y}$ makes queries to the server S.

Remark 27. $\mathcal{X}$ and $\mathcal{Y}$ are not necessarily the same user, as $\mathcal{X}$ has full knowledge of the database he created whereas $\mathcal{Y}$ knows only what he receives from S. However, when we integrate this cryptographic primitive into in a biometric identification system (see section V.3), we merge the entities by the correspondence described in table V.1.

Definition of the Primitives

In the sequel, messages are binary strings of a fixed length N, and $d(x_1, x_2)$ is the Hamming Distance between $x_1, x_2 \in \{0,1\}^N$.

Here comes a formal definition of the primitives that enable to perform an error-tolerant searchable encryption; this definition cannot be separated from the definitions of Completeness(λ_{min}) and ϵ-Soundness(λ_{max}), which follow.

Architecture for Biometric Ident. (Sec. V.3)	Entities for E.-T. Searchable Encryption (Sec. V.2)
Users $\mathcal{U}_i$, with biometric samples b_i and pseudo-identities ID_i	Messages x
Identity Provider $\mathcal{IP}$	Sender $\mathcal{X}$ at enrolment, Receiver $\mathcal{Y}$ at identification
Database $\mathcal{DB}$	Server $\mathcal{S}$

Table V.1: Correspondence between entities.

Definition 9. A $(\epsilon, \lambda_{min}, \lambda_{max})$*-Public Key Error-Tolerant Searchable Encryption* [33] is obtained with the following probabilistic polynomial-time methods:

- $\mathsf{KeyGen}(1^k)$ initializes the system, and outputs public and private keys (pk, sk); k is the security parameter. The public key pk is used to store data on a server, and the secret key sk is used to retrieve information from that server.

- $\mathsf{Send}_{\mathcal{X},\mathsf{S}}(x, pk)$ is a protocol in which $\mathcal{X}$ sends to S the data $x \in \{0,1\}^N$ to be stored on the storage system. At the end of the protocol, S associated an identifier to x, denoted $\varphi(x)$.

- $\mathsf{Retrieve}_{\mathcal{Y},\mathsf{S}}(x', sk)$ is a protocol in which, given a fresh data $x' \in \{0,1\}^N$, $\mathcal{Y}$ asks for the identifiers of all data that are stored on S and are close to x', with Completeness(λ_{min}) and ϵ-Soundness(λ_{max}). This outputs a set of identifiers, denoted $\Phi(x')$.

These definitions are completed by condition 9 that defines Completeness and ϵ-Soundness. In a few words, Completeness implies that a registered message x is indeed found if the query word x' is at a distance less than λ_{min} from x, while ϵ-Soundness means that with probability greater than $1 - \epsilon$, no message at a distance greater than λ_{max} from x' will be returned.

The Send protocol produces an output $\varphi(x)$ that identifies the data x. This output $\varphi(x)$ is meant to be a unique identifier, which is a binary string of undetermined length – in other words, elements of $\{0,1\}^\star$ – that enables to retrieve x. It can be a timestamp, a name or nickname, *etc.* depending on the application.

Security Requirements

First of all, it is important that the scheme actually works, *i.e.* that the retrieval of a message near a registered one gives the correct result. This can be formalized into the following condition:

Condition 9 (Completeness, ϵ-Soundness)**.** Let $x_1, \ldots, x_p \in B = \{0,1\}^N$ be p different binary vectors, and let $x' \in B$ be another binary vector. Suppose that the system was initialized, that all the messages x_i have been sent by user $\mathcal{X}$ to the system S with identifiers $\varphi(x_i)$, and that user $\mathcal{Y}$ retrieved the set of identifiers $\Phi(x')$ associated to x'.

1. The scheme is said to be **complete**(λ_{min}) if the identifiers of all the x_i that are near x' are almost all in the resulting set $\Phi(x')$, *i.e.* if

   $$\eta_c = \Pr_{x'}\left[\exists i \text{ s.t. } d(x', x_i) \leq \lambda_{min}, \varphi(x_i) \notin \Phi(x')\right]$$

   is negligible.

2. The scheme is said to be ϵ-**sound**(λ_{max}) if the probability of finding an unwanted result in $\Phi(x')$, i.e.

   $$\eta_s = \Pr_{x'}\left[\exists i \in [\![1, p]\!] \text{ s.t. } \ d(x', x_i) > \lambda_{max}\ , \varphi(x_i) \in \Phi(x')\right],$$

   is bounded by ϵ.

The first condition simply means that registered data is effectively retrieved if the input is close. η_c expresses the probability of failure of this Retrieve operation.

The second condition means that only the close messages are retrieved, thus limiting false alarms. η_s measures the reliability of the Retrieve query, *i.e.* if all the results are identifiers of messages near to x'.

These two properties (*Completeness* and ϵ-*Soundness*) are sufficient to have a working set of primitives which allows to make approximate queries on a remote storage server. The following conditions, namely *Sender Privacy* and *Receiver Privacy*, ensure that the data stored and retrieved in the server is secure, and that communications can be done on an untrusted network. In these, Ω is an integer polynomial in the security parameter k.

Condition 10 (**Sender Privacy**)**.** The scheme is said to respect *Sender Privacy* if the advantage of any malicious server is negligible in the $\mathsf{Exp}_{\mathcal{A}}^{\text{Sender Privacy}}$ experiment, described below. Here, $\mathcal{A}$ is an "honest-but-curious" opponent taking the place of S, and $\mathcal{C}$ is a challenger at the user side.

```
Exp_A^{Sender Privacy}
  1. $(pk, sk)$                  ←  $\mathsf{KeyGen}(1^k)$                        $(\mathcal{C})$
  2. $\{x_2, \ldots, x_\Omega\}$  ←  $\mathcal{A}$                                 $(\mathcal{A})$
  3. $\varphi(x_i)$              ←  $\mathsf{Send}_{\mathcal{X},\mathcal{A}}(x_i, pk)$     $(\mathcal{C})$
  4. $\{x_0, x_1\}$              ←  $\mathcal{A}$                                 $(\mathcal{A})$
  5. $\varphi(x_e)$              ←  $\mathsf{Send}_{\mathcal{X},\mathcal{A}}(x_e, pk)$,    $(\mathcal{C})$
                                    $e \in_R \{0,1\}$
  6. Repeat steps $(2, 3)$
  7. $e' \in \{0,1\}$            ←  $\mathcal{A}$                                 $(\mathcal{A})$
```

The advantage of the adversary is $|\Pr[e' = e] - \frac{1}{2}|$.

In this experiment, $\mathcal{A}$ first chooses $\Omega - 1$ messages to send (Step 2) and observes the corresponding Send requests executed by $\mathcal{C}$ (Step 3). Based on this, he chooses 2 messages x_0, x_1 (Step 4) over which he believes he has an advantage. One of these messages is randomly chosen by $\mathcal{C}$, and sent to $\mathcal{S}$ (Step 5). The adversary can then try to obtain more information, repeating Steps 2 and 3 a polynomial number of times; after that, he returns his estimation e' on the message $x_e \in \{x_0, x_1\}$ that was sent by $\mathcal{C}$ (Step 7). $\mathcal{A}$ wins the game if $e' = e$.

This condition is that of the privacy of the content stored in the server. The content that the sender transmits is protected, justifying the title "Sender Privacy".

Another important privacy aspect is the secrecy of the receiver's data. We do not want the server to have information on the fresh data x' that is queried; this is expressed by the following condition.

Condition 11 (**Receiver Privacy**). The scheme is said to respect *Receiver Privacy* if the advantage of any malicious server is negligible in the $\mathsf{Exp}_{\mathcal{A}}^{\text{Receiver Privacy}}$ experiment described below. As in the previous condition, $\mathcal{A}$ denotes the "honest-but-curious" opponent taking the place of S, and $\mathcal{C}$ the challenger at the user side.

```
$\mathsf{Exp}_{\mathcal{A}}^{\text{Receiver Privacy}}$
    1. $(pk, sk)$                              $\leftarrow$  $\mathsf{KeyGen}(1^k)$                          $(\mathcal{C})$
    2. $\{x_1, \ldots, x_\Omega\}$                $\leftarrow$  $\mathcal{A}$                                     $(\mathcal{A})$
       $d(x_i, x_j) > \lambda_{max}, \forall i, j \in [\![1, \Omega]\!]$
    3. $\varphi(x_i), (i \in [\![1, \Omega]\!])$   $\leftarrow$  $\mathsf{Send}_{\mathcal{X},\mathcal{A}}(x_i, pk)$       $(\mathcal{C})$
    4. $\{x'_2, \ldots, x'_p\}$                 $\leftarrow$  $\mathcal{A}$                                     $(\mathcal{A})$
    5. $\Phi(x'_j), (j \in [\![2, p]\!])$       $\leftarrow$  $\mathsf{Retrieve}_{\mathcal{Y},\mathcal{A}}(x'_j, sk)$   $(\mathcal{C})$
    6. $(x'_0, x'_1)$                          $\leftarrow$  $\mathcal{A}$                                     $(\mathcal{A})$
    7. $\Phi(x'_e)$                            $\leftarrow$  $\mathsf{Retrieve}_{\mathcal{Y},\mathcal{A}}(x'_e, sk),$  $(\mathcal{C})$
                                                     $e \in_R \{0, 1\}$
    8. Repeat Steps $(4, 5)$
    9. $e' \in \{0, 1\}$                        $\leftarrow$  $\mathcal{A}$                                     $(\mathcal{A})$
```

The advantage of the adversary is $|\Pr[e' = e] - \frac{1}{2}|$.

This condition is the mirror image of the previous one. It transposes the idea that the receiver $\mathcal{Y}$ can make his queries to S without leaking information on their content. For this, $\mathcal{A}$ chooses a set of messages $\{x_1, \ldots, x_\Omega\}$ with a minimal distance between each pair of messages greater than λ_{max} (Step 2), and these messages are sent by $\mathcal{C}$ (Step 3). $\mathcal{A}$ then chooses messages to be queried (Step 4); they are queried by $\mathcal{C}$ (Step 5). Based on the information gathered during these five steps, $\mathcal{A}$ issues the challenge, namely two more messages to be queried (Step 6). One of them is retrieved by $\mathcal{C}$ (Step 7). $\mathcal{A}$ has the right to issue some more $\mathsf{Retrieve}$ queries before issuing his estimate $e' \in \{0, 1\}$ (Steps 8 and 9). $\mathcal{A}$ wins the game if $x'_{e'}$ was the message that was retrieved at Step 7.

Remark 28. Conditions 10 and 11 are the transposition of their homonym statement in [20]. They aim for the same goal, *i.e.* privacy – against the server – of the data that is registered first, then looked for.

Remark 29. In this model, the adversary is always the server. However, in realistic implementations of Biometric Identification systems, a reasonable goal would be for the users that get identified or for a specific officer who manages the identification **not** to learn information on the rest of the database. This condition, called *Symmetric Receiver Privacy*, is described hereafter.

Symmetric Receiver Privacy aims at limiting the amount of information that $\mathcal{Y}$ gets through the protocol. Indeed, if previous constructions of Searchable Encryption such as [19, 78] seem to consider that the sender and the receiver are the same person, thus owning the database in the same way, there are applications where the receiver must not dispose of the entire database. If for example different users $\mathcal{Y}_i$ have access to the application, we do not want user $\mathcal{Y}_i$ to obtain information on another user $\mathcal{Y}_j$'s data.

For this purpose, we define a database simulator $\mathcal{S}_1$.

$\mathcal{S}_1(x')$ is a simulator which only knows the tags of the registered elements that are in $\Phi(x')$, while the other elements are random. In other words, after sending words $x_1, \ldots, x_\Omega$ to $\mathcal{S}_1(x')$, only the x_i at distance less than λ_1 from x' are taken into account when replying to a $\mathsf{Retrieve}$ query. Here, x' stands for the message to be retrieved.

On the other hand, $\mathcal{S}_0$ is the regular server, which genuinely runs the protocol.

Condition 12 (Symmetric Receiver Privacy)**.** The scheme is said to respect *Symmetric Receiver Privacy* if there exists a simulator $\mathcal{S}_1$ such that the advantage of any malicious receiver is negligible in the $\mathsf{Exp}_{\mathcal{A}}^{\text{Sym-Rec-Privacy}}$ experiment described below. Here, $\mathcal{A}$ is the "honest-but-curious" opponent taking the place of $\mathcal{Y}$, and $\mathcal{C}$ the challenger at the server side.

$$
\begin{array}{l}
\mathsf{Exp}_{\mathcal{A}}^{\text{Sym-Rec-Privacy}} \\
\quad \left|
\begin{array}{lllll}
1. & (pk, sk) & \leftarrow & \mathsf{KeyGen}(1^k) & (\mathcal{A}) \\
2. & \{x_1, \ldots, x_\Omega\}, & \leftarrow & \mathcal{A} & (\mathcal{A}) \\
 & \multicolumn{3}{l}{d(x_i, x_j) > \lambda_{max}, \forall i, j \in [\![1, \Omega]\!]} & \\
3. & \varphi(x_i) & \leftarrow & \mathsf{Send}_{\mathcal{A},\mathcal{S}}(x_i, pk) & (\mathcal{A}) \\
4. & e \in_R \{0, 1\} & \leftarrow & \mathcal{C} & (\mathcal{C}) \\
5. & \{x'_1, \ldots, x'_p\}, & \leftarrow & \mathcal{A} & (\mathcal{A}) \\
 & \multicolumn{3}{l}{d(x'_i, x'_j) > \lambda_{max}, \forall i, j \in [\![1, p]\!]} & \\
6. & \Phi(x'_i) & \leftarrow & \mathsf{Retrieve}_{\mathcal{A},\mathcal{S}_e}(x'_i, sk) & (\mathcal{A}) \\
7. & e' \in \{0, 1\} & \leftarrow & \mathcal{A} & (\mathcal{A})
\end{array}
\right.
\end{array}
$$

The advantage of the adversary is $|\Pr[e' = e] - \frac{1}{2}|$.

This new condition does not fit into previous models for Searchable Encryption, and is not satisfied by constructions such as [20, 78]. It is inspired by the Data Privacy property of SPIR protocols, which states that it is not possible to tell whether or not $\mathcal{S}$ possesses more data than the received messages. Indeed, if the receiver is able to tell the difference between a server $\mathcal{S}_0$ that possess more data than what $\mathcal{Y}$ received, and a server $\mathcal{S}_1$ that just has in memory the information that $\mathcal{Y}$ needs, then $\mathcal{Y}$ detains more information than what he ought to; that is why this indistinguishability game fits the informal description of *Symmetric Receiver Privacy.*

A Construction for Error-Tolerant Searchable Encryption

Technical Description

Our searching scheme uses all the tools we described, along with cryptographic primitives. More information on the cryptographic primitives can be found in Appendix C. As we will see, this enables to meet the privacy requirements defined earlier. More precisely:

- We choose a family H of functions: $h : \{0,1\}^N \rightarrow \{0,1\}^t$ that have the LSH property,
- We pick a family H' of functions: $h' : \{0,1\}^t \times [\![1, |H|]\!] \rightarrow [\![1, M]\!]$, adapted to a Bloom filter structure,
- From these two families, and following Algorithm 4, we deduce a family H^c of functions $h^c : \{0,1\}^N \rightarrow [\![1, M]\!]$,
- We use a semantically secure public key cryptosystem $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$ [79],
- We use a PIR protocol with query function $\mathsf{Query}^{PIR}_{\mathcal{Y},\mathsf{S}}$.
- We use a PIS function $\mathsf{update}_{\mathsf{BF}}(val, i)$ that adds val to the i-th bucket of the Bloom filter.

Here come the details of the implementation. In a few words, storage and indexing of the data are separated, so that it becomes feasible to search over the encrypted documents. Indexing is made with Bloom Filters, with an extra precaution of encrypting the content of all the buckets. Finally, using our locality-sensitive hashing functions permits error-tolerance.

System setup

The method $\mathsf{KeyGen}(1^k)$ initializes M different buckets to $\emptyset$. The public and secret keys of the cryptosystem (pk, sk) are generated by $\mathsf{Setup}(1^k)$, and sk is given to $\mathcal{Y}$.

Sending a message

The protocol $\mathsf{Send}_{\mathcal{X},\mathcal{S}}(x, pk)$ goes through the following steps (cf. Fig. V.2):

1. **Identifier establishment** $\mathcal{S}$ attributes to x a unique identifier $\varphi(x)$, and sends it to $\mathcal{X}$.
2. **Data storage** $\mathcal{X}$ sends $\mathsf{Enc}(x)$ to $\mathcal{S}$, who stores it in a memory cell that depends on $\varphi(x)$.
3. **Data indexing**
    - $\mathcal{X}$ computes $h^c(x)$ for all $h^c \in H^c$,
    - and executes $\mathsf{update_{BF}}(\mathsf{Enc}(\varphi(x)), h^c(x))$ to send $\mathsf{Enc}(\varphi(x))$ to be added to the filter's bucket of index $h^c(x)$ on the server side.

Note that for privacy concerns, we complete the buckets with random data in order to get the same bucket size l for the whole data structure.

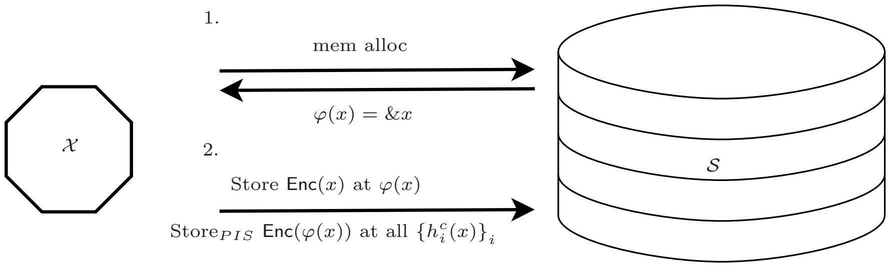

1.
mem alloc
$\varphi(x) = \&x$
$\mathcal{X}$
2.
Store $\mathsf{Enc}(x)$ at $\varphi(x)$
Store_{PIS} $\mathsf{Enc}(\varphi(x))$ at all $\{h_i^c(x)\}_i$
$\mathcal{S}$

Figure V.2: Sending a message in a nutshell

The first phase (identifier establishment) is done to create an identifier that can be used to register and then retrieve x from the database. For example, $\varphi(x)$ can be the time at which $\mathcal{S}$ received x, or the first memory address that is free for the storage of $\mathsf{Enc}(x)$. In this case, the address will be used in the second phase.

The third phase applies the combination of BFS and LSH functions to x so that it is possible to retrieve x with some approximate data. This is done with the procedure described hereafter.

Retrieving data

The protocol $\mathsf{Retrieve}_{\mathcal{Y},\mathcal{S}}(x', sk)$ goes through the following steps (cf. Fig. V.3):

1. $\mathcal{Y}$ computes each $\alpha_i = h_i^c(x')$ for each $h_i^c \in H^c$, then executes $\mathsf{Query}^{PIR}_{\mathcal{Y},\mathcal{S}}(\alpha_i)$ to receive the filter bucket T_{α_i},

2. $\mathcal{Y}$ decrypts the content of each bucket T_{α_i} and computes the intersection of all the $\mathsf{Dec}(T_{\alpha_i})$,

3. This intersection is a set of identifiers $\{\varphi(x_{i_1}), \dots, \varphi(x_{i_\gamma})\}$, which is the result of the execution of $\mathsf{Retrieve}$.

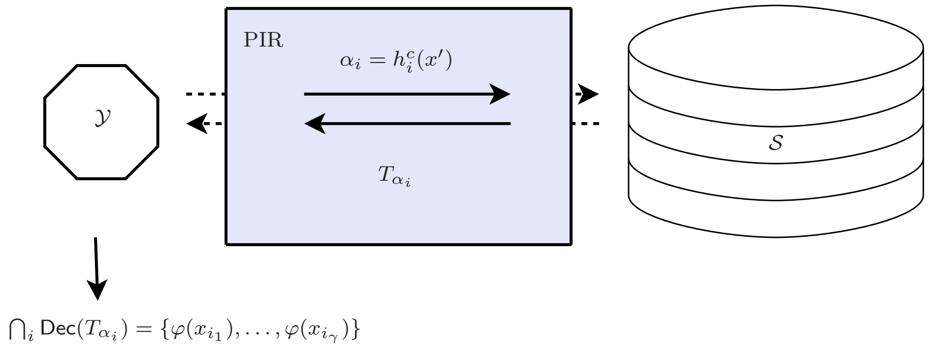

PIR
$\alpha_i = h_i^c(x')$
T_{α_i}
$\mathcal{Y}$
$\mathcal{S}$
$\bigcap_i \mathsf{Dec}(T_{\alpha_i}) = \{\varphi(x_{i_1}), \dots, \varphi(x_{i_\gamma})\}$

Figure V.3: Retrieving data in a nutshell

As we can see, the retrieving process follows that of a BFS, with the noticeable differences that 1. the identifiers are always encrypted in the database, and 2. the query is made following a PIR protocol. This allows us to benefit from both the Bloom filter structure, the locality-sensitive hashing, and the privacy-preserving protocols.

The secure protocols involved do not leak information on the requests made, and the next section discusses more precisely the security properties achieved.

Security Properties

We now demonstrate that this construction faithfully achieves the security requirements we defined in section V.2.

Proposition V.1 (**Completeness**). *Provided that H is a $(\lambda_{min}, \lambda_{max}, \epsilon_1, \epsilon_2)$-LSH family, for a negligible ϵ_1, this scheme is complete.*

Proposition V.2 (**ϵ-Soundness**). *Provided that H is a $(\lambda_{min}, \lambda_{max}, \epsilon_1, \epsilon_2)$-LSH family from $\{0,1\}^N$ to $\{0,1\}^t$, and provided that the Bloom filter functions H' behave like pseudo-random functions from $\{0,1\}^t \times [\![1, |H|]\!]$ to $[\![1, M]\!]$, then the scheme is ϵ-sound, with:*

$$\epsilon = \left(\epsilon_2 + (1-\epsilon_2)\frac{1}{M}\right)^{|H^c|}$$

Propositions V.1 and V.2 are direct consequence of Lemma V.3.

Remark 30. Proposition V.2 assumes that the Bloom filter hash functions are pseudo-random; this hypothesis is pretty standard for Bloom filter analysis. It can be achieved by using cryptographic hash functions with a random oracle-like behaviour.

Proposition V.3 (**Sender Privacy**). *Assume that the underlying cryptosystem is semantically secure and that the PIS function* update_{BF} *achieves User Privacy, then the scheme ensures Sender Privacy.*

Proof If the scheme does not ensure Sender Privacy, then there exists an attacker who can distinguish between the output of $\mathsf{Send}(x_0, pk)$ and $\mathsf{Send}(x_1, pk)$, after the execution of $\mathsf{Send}(x_i, pk)$, $i \in [\![2, \Omega]\!]$.

Note that the content of the Bloom filter buckets does not reveal information that can permit to distinguish between x_0 and x_1. Indeed, the only information $\mathcal{A}$ has with the filter structure is a set of $\mathsf{Enc}(\varphi(x_i))$ placed at different indexes $h^c(x_i)$, $i = e, 2, \dots, \Omega$. Due to the semantic security of Enc, this does not permit to distinguish between $\varphi(x_0)$ and $\varphi(x_1)$.

This implies that, with inputs

$$\{\mathsf{Enc}(x_i), \mathsf{update_{BF}}(\mathsf{Enc}(\varphi(x_i)), h^c(x_i))\}_{i \geq 2},$$

the attacker can distinguish between $\mathsf{Enc}(x_0)$, $\mathsf{update_{BF}}(\mathsf{Enc}(\varphi(x_0)), h^c(x_0))$ and $\mathsf{Enc}(x_1)$, $\mathsf{update_{BF}}(\mathsf{Enc}(\varphi(x_1)), h^c(x_1))$.

As $\mathsf{update_{BF}}$ does not leak information on its inputs, that means that the attacker can distinguish between $\mathsf{Enc}(x_0)$ and $\mathsf{Enc}(x_1)$ by choosing some other inputs to Enc. That contradicts the semantic security assumption. □

Proposition V.4 (**Receiver Privacy**). *Assume that the PIR ensures User Privacy, then the scheme ensures Receiver Privacy.*

Proof This property is a direct deduction of the PIR's User Privacy, as the only information $\mathcal{S}$ gets from the execution of a $\mathsf{Retrieve}$ is a set of Query^{PIR}. □

These properties show that this protocol for Error-Tolerant Searchable Encryption has all the security properties that we looked for, except Symmetric Receiver Privacy, which will be achieved in the next section. LSH functions are used in such a way that they do not degrade the security properties of the system.

Achieving Symmetric Receiver Privacy

Specific Tools

For this purpose, we specify a second cryptosystem $(\mathcal{S}et, \mathcal{E}nc, \mathcal{D}ec)$ to be that of El Gamal (see Appendix C). This system has the following homomorphic property:

$$\mathcal{D}ec(\mathcal{E}nc(x)\mathcal{E}nc(x')) = xx'.$$

Secret splitting Let s be a *small* secret; we wish to split s into n re-randomizable parts. There is a general technique for this, called Proactive Secret Sharing [89, 161], but for clarity reasons, we propose a simple technique for this. We construct n shares $A_1, \dots, A_n$ such that $A_i = g^{r_i}$ where r_i is a random integer, for $i \in [\![1, n-1]\!]$ and $A_n = g^{-\sum r_i + s}$, where g is the generator of a group of *large* prime order q. Recovering s can be done by multiplying all the A_i, and then proceeding to an exhaustive search to compute the discrete logarithm of g^s in basis g. Re-randomization of the parts A_i can easily be done by choosing a random integer t, and replacing each A_i by A_i^t. The generator for the discrete logarithm must then be replaced by g^t.

Extending our Scheme

The scheme proposed does not achieve Symmetric Receiver Privacy. For example, the user $\mathcal{Y}$ has access to all the $\varphi(x_i)$ such that there exists $h^c, h_0^c \in H^c, h^c(x_i) = h_0^c(x')$. Without further caution, a malicious user could get more information than what he ought to. We here describe an example of a protocol variant that leads to the desired properties.

We will apply secret splitting to the tags $\varphi(x)$ returned by Send. That implies that we consider the range of $\varphi(x)$ to be relatively small, for example of 32-bit long integers[4]. Primitives are adapted this way:

- $\mathsf{KeyGen}(1^k)$ is unchanged, but here both Setup and $\mathcal{S}et$ are used to generate (pk, sk),
- $\mathsf{Send}_{\mathcal{X},\mathcal{S}}(x, pk)$ is slightly modified, namely:

  1. **Identifier establishment (*unchanged*)** $\mathcal{S}$ attributes to x a unique identifier $\varphi(x)$, and sends it to $\mathcal{X}$.
  2. **Data storage (*unchanged*)** $\mathcal{X}$ sends $\mathsf{Enc}(x)$ to $\mathcal{S}$, who stores it in a memory cell that depends on $\varphi(x)$.
  3. **Data indexing** First $\mathcal{X}$ splits the tag $\varphi(x)$ into $|H^c|$ shares noted $A_{x,1}, \dots, A_{x,|H^c|}$ by applying the method described above, and picks a random integer r_x,

[4] 32-bit long integers are a good example for a practical construction, as it is the size of a standard memory address in many computer architectures.

then $\mathcal{X}$ computes all $h_i^c(x)$, and executes the queries

$$\mathsf{update}_{\mathsf{BF}}((\mathcal{E}nc(f^{r_x}), \mathcal{E}nc(A_{x,i})), h_i^c(x))$$

to send $(\mathcal{E}nc(f^{r_x}), \mathcal{E}nc(A_{x,i}))$ to be added to the filter's bucket of index $h_i^c(x)$, where h_i^c is the i-th function of H^c, for $i \in [\![1, |H^c|]\!]$.

At the end of this update, the bucket T_α of the filter is filled with $l = |T_\alpha|$ couples

$$(\mathcal{E}nc(f^{z_{\alpha,j}}), \mathcal{E}nc(B_{\alpha,j})), j \in [\![1, l]\!].$$

$B_{\alpha,j}$ is a share of some tag, or a random element of the group.

- $\mathsf{Retrieve}_{\mathcal{Y},\mathcal{S}}(x', sk)$ is adapted consequently:

  1. ***(unchanged)*** $\mathcal{Y}$ computes each $\alpha_i = h_i^c(x')$ for $h_i \in H^c$, then executes $\mathsf{Query}_{\mathcal{Y},\mathcal{S}}^{PIR}(\alpha_i)$,
  2. $\mathcal{S}$ first re-randomizes with the values $(c_1, c_2) \in \mathbb{N}^*$ the content of each bucket of the Bloom filter database by the same random value. The filter bucket

     $$T_{\alpha_i} = \{(\mathcal{E}nc(f^{z_{\alpha_i,j}}), \mathcal{E}nc(B_{\alpha_i,j})), j \in [\![1, l]\!]\}$$

     becomes

     $$T_{\alpha_i}^{c_1,c_2} = \{(\mathcal{E}nc(f^{z_{\alpha_i,j}})^{c_1}, \mathcal{E}nc(B_{\alpha_i,j})^{c_2}), j \in [\![1, l]\!]\}$$

     $\mathcal{S}$ then answers to the PIR Query, and sends along g^{c_2} to $\mathcal{Y}$,
  3. $\mathcal{Y}$ decrypts the content of each bucket $T_{\alpha_i}^{c_1,c_2}$ to get a set of couples $(f^{z_{\alpha_i,j}c_1}, B_{\alpha_i,j}^{c_2})$,
  4. If the same element f^{zc_1} is present in the intersection of all the different sets $T_{\alpha_i}^{c_1,c_2}$, then $\mathcal{Y}$ possesses all shares of a tag $\varphi(x)$, and can compute $\prod_{i=1}^{|H^c|} A_{x,i}^{c_2} = (g^{c_2})^{\varphi(x)}$,
  5. $\mathcal{Y}$ finally runs a discrete logarithm of $(g^{c_2})^{\varphi(x)}$ in basis g^{c_2}, and adds $\varphi(x)$ to the set of results $\Phi(x')$.

Note that this scheme can also be generalized for other Proactive Secret Sharing schemes.

Remark 31. This adaptation of Proactive Secret Sharing Schemes to our problem actually requires that all shares of the secret be present in order to reconstruct the identifier $\varphi(x)$; this is likely *not* to happen with biometrics and LSH functions. However, using threshold-based proactive secret sharing schemes enables to reconstruct secrets without all their shares.

Security Properties

This new scheme is an extension of the previous one, and the same security properties are achieved. Moreover, Condition 12 also holds. Indeed, the modification to the Send procedure is not significant enough to alter the Sender Privacy property: the only modification on $\mathcal{S}$'s side is the content of the $\mathsf{update}_{\mathsf{BF}}$ procedure, which does not leak. Moreover, the Receiver Privacy property is also preserved, as communications from $\mathcal{Y}$ to $\mathcal{S}$ in $\mathsf{Retrieve}$ only involves a PIR query.

Proposition V.5 (Symmetric Receiver Privacy)**.** *Assume the PIR ensures Data Privacy i.e. it is a SPIR, and that H is a $(\lambda_{min}, \lambda_{max}, \epsilon_1, \epsilon_2)$-LSH family with a negligible ϵ_2, then the scheme ensures Symmetric Receiver Privacy, over the Decisional Diffie-Hellman hypothesis.*

To demonstrate this proposition, let us begin with a preliminary Lemma.

Lemma V.4. *Let $s_1, \dots, s_t \in S$ be t different secrets, with $|S|$ small. Let $A_{i,1}, \dots, A_{i,n}$ be the n parts of the secret s_i split with the aforementioned method. Let $\pi_0 \subset \{A^c_{i,j}, i \in [\![1,t]\!], j \in [\![1,n]\!], c \in [\![1,q]\!]\}$ be collection of k such parts, and $\pi_1 = \{g^{r_1}, \dots, g^{r_k}\}$ a set of k random elements of the cyclic group $\mathcal{G}$.*

Under the DDH assumption, if an adversary $\mathcal{A}$ can distinguish between π_0 and π_1, then there exists $c_0 \in [\![1,q]\!], i \in [\![1,t]\!]$ such that $\{A^{c_0}_{i,1}, \dots, A^{c_0}_{i,n}\} \subset \pi_0$.

Proof

Let (g, g^a, g^b, g^c) be an instance of the DDH problem. An adversary can solve this instance if he can tell, with non-negligible probability, whether $g^c = g^{ab}$ or not.

We take $t = 1$, because all secrets are independent, and $n = 2$ (if $n > 2$, we multiply the parts and return to the case $n = 2$). Suppose the lemma is false, that means there exists a polynomial algorithm $\mathcal{A}$ which takes as inputs couples $(g^{c_u}, g^{c_u r})$ and $(g^{c_v}, g^{c_v(s-r)})$, with $c_u \neq c_v$, and that returns the secret s with non-negligible probability.

We then give as input to $\mathcal{A}$ the couples (g, g^a) and (g^b, g^{bs-c}) for $s \in S$. If $\mathcal{A}$ returns s, then $g^{c-bs} = g^{b(a-s)} = g^{ba}g^{-bs}$. We finally have an advantage on the DDH problem; that proves the lemma. □

Proof *(Proposition V.5)*

We now build a simulator $\mathcal{S}_1$ for the server in order to prove the proposition. Let x' be the request and $\Phi(x') = \{\varphi(x_1), \dots, \varphi(x_k)\}$ be the genuine answer to $\mathsf{Retrieve}(x', sk)$. First, the simulator generates Ω random elements $\{z_1, \dots, z_\Omega\} \subset [\![1,q]\!]$; he associates the first k elements to the elements of $\Phi(x')$. The simulator splits each of the $\varphi(x_j)$ into the $n = |H^c|$ parts $A_{x_j,1}, \dots, A_{x_j,n}$. Finally, he picks random integers c_1, c_2.

Since the PIR is symmetrical, we can impose the response to each $\mathsf{Query}(\alpha_i)$ to be a set containing the k elements that must be present in the intersection, namely $\left(\mathcal{E}nc(f^{z_j})^{c_1}, \mathcal{E}nc(A_{x_j,i})^{c_2}\right)$, and the remaining random values

$$(\mathcal{E}nc(f^z)^{c_1}, \mathcal{E}nc(g^r))\,,$$

with z a random element of $\{z_{k+1}, \ldots, z_\Omega\}$ and r a random integer. We give to the simulator enough memory to remember which z was returned for which α, so that multiple queries to the same α are consistent. The simulator also returns g^{c_2}.

Let $\mathcal{A}$ be a malicious receiver in the $\mathsf{Exp}_{\mathcal{A}}^{\text{Sym-Rec-Privacy}}$ experiment. Following Cond. 12, $\mathcal{A}$ makes p $\mathsf{Retrieve}$ queries to $\mathcal{S}$; each of these requests lead to $|H^c|$ calls to Query. As the requests x'_i are λ_{max}-separated, and as the hashes are $\lambda_{min}, \lambda_{max}, \epsilon_1, \epsilon_2$ with a negligible ϵ_2, we can consider these $\mathsf{Retrieve}$ queries to be independent.

Note that the first parts of the Bloom filters are always indistinguishable, as they are generated in the same way. Therefore, if $\mathcal{A}$ distinguishes between $\mathcal{S}_0$ and $\mathcal{S}_1$, that means that he distinguished between a given π_0 and π_1, constructed by taking the set of all answers to the Query request he made. By application of the Lemma, we deduce the proposition.

□

V.3 Application to Identification with Encrypted Biometric Data

Our Biometric Identification System

We now apply our construction for error-tolerant searchable encryption to our biometric identification purpose. Due to the security properties of the above construction, this enables us to design a biometric identification system which achieves the security objectives stated in section V.2.

While applying the primitives of error-tolerant searchable encryption, the database $\mathcal{DB}$ takes the place of the server $\mathcal{S}$; the role of the Identity Provider $\mathcal{IP}$ varies with the step we are involved in. During the Enrolment step, $\mathcal{IP}$ behaves as $\mathcal{X}$, and as $\mathcal{Y}$ during the Identification step. In this step, $\mathcal{IP}$ is in possession of the private key sk used for the $\mathsf{Retrieve}$ query.

Enrolment

- To enrol a user $\mathcal{U}_i$, the sensor $\mathcal{SC}$ acquires a sample b_i from his biometrics and sends it to $\mathcal{IP}$,
- The Identity Provider $\mathcal{IP}$ then executes $\mathsf{Send}_{\mathcal{X},\mathcal{S}}(b_i, pk)$.

Identification

- $\mathcal{SC}$ captures a fresh biometric template b' from a user $\mathcal{U}$ and sends it to $\mathcal{IP}$,
- The Identity Provider $\mathcal{IP}$ then executes $\mathsf{Retrieve}_{\mathcal{Y},\mathcal{S}}(b', sk)$.

At the end of the identification, $\mathcal{IP}$ has the fresh biometric template b' along with the address of the candidate reference templates in the database. To reduce the list of identities, we can use a secure matching scheme [37, 157] to run a final secure comparison between b' and the candidates.

Practical Considerations

Choosing the LSH family: an Example

Let's place ourself in the practical setting of human identification through iris recognition, using Daugman's IrisCode presented in Chapter II.1. The biometric templates are in $\mathcal{B} = \{0, 1\}^{2048}$.

There are several paths to design LSH functions adapted to this kind of data. Random projections such as those defined in [109], is a convenient way to create LSH functions for binary vectors. However, for the sake of simplicity, we propose to use the functions used in [88], in which they are referred as 'beacon indexes'. These functions are based on the fact that all IrisCode bits do not have the same distribution probability.

In a few words, these functions first reorder the bits of the IrisCode by rows, so that in each row, the bits that are the most likely to induce an error are the least significant ones. The column are then reordered to avoid correlations between following bits. The most significant bits of rows are then taken as 10-bit hashes. The efficiency of this approach is demonstrated in [88] where the authors apply these LSH functions to identify a person with his IrisCode. They report experiments done on the UAE database which contains $L = 632500$ records; trivial identification would then require L classical matching computation, or just a fraction of L if the first matching element is selected, which is way too much for a large database. Instead, they apply $\mu = 128$ of those hashes to the biometric data, and look for IrisCodes that get the same LSH results for at least 3 functions. In doing this, they limit the number of necessary matching to 41 instead of L.

To determine the LSH capacity of these hash functions is not easy to do with real data; however, if we model b and b' as binary vectors such that each bit of b is flipped with a fixed probability (*i.e.* if b' is obtained out of b through a binary symmetric channel), then the family induced is $(r_1, r_2, (1 - \frac{r_1}{2048})^{10}, (1 - \frac{r_2}{2048})^{10})$-LSH. This estimation is conservative as IrisCodes are not random noisy data, and the selected bits are more reliable than the average.

Combining these functions with a Bloom filter with storage in the way described in section V.2 enables to have a secure identification scheme.

Overall complexity and efficiency

We here evaluate the computational complexity of an identification request on the client's side. We denote by $\kappa(op)$ the cost of operation op, and $|S|$ the size of the set S. Recalling section V.2, the overall cost of a request is:

$$\begin{aligned}\kappa(\text{request}) &= |H^c|(\kappa(\text{hash}) + \kappa(PIR) + |T|\kappa(\mathsf{Dec})) + \kappa(\text{intersection}) \\ &\leq |H^c|\left(\kappa\left(\text{h}_{BF}\right) + \kappa\left(\text{h}_{LSH}\right) + \kappa\left(PIR\right) + |T|\kappa\left(\mathsf{Dec}\right)\right) + O(|T||H^c|)\end{aligned}$$

We here used data structures in which intersection of sets is linear in the set length, hence the term $O(|T||H^c|)$; $|T|$ is the maximum size of a Bloom filter with storage bucket.

To conclude this complexity estimation, let us recall that the cost of a hash function can be neglected in front of the cost of a decryption step. The PIR query complexity at the sensor level depends on the scheme used (remember that the PIR query is made only over the m buckets and not over the whole database); in the case of Lipmaa's PIR [113], this cost $\kappa(PIR)$ is dominated by the cost of a Damgård-Jurik encryption. The overall sensor complexity of an identification request is $O(\mu\nu(|T|\kappa(\mathsf{Dec}) + \kappa(PIR)))$.

On Biometric Error Rates

The variability of biometrics induce False Acceptances and False Rejects. On the original biometric system, *i.e.* in the authentication setting, False Acceptances happen if the distance $d(b, b'') < \lambda_{min}$ where b, b'' are templates computed from different users. Conversely, False Rejects happen if $d(b, b') > \lambda_{max}$ where b, b' are template computed from the same user.

In our identification scheme, due to the Completeness, False Reject Rates are not to be increased (by more than a negligible quantity). The scheme is also ϵ-Sound, but here ϵ is not negligible. Therefore, the False Accept Rate will be increased by $FAR' = (1-\epsilon)FAR + \epsilon > FAR$. A good approximation for the small ϵ is $FAR' \approx FAR + \epsilon$. If ϵ is not small then the scheme induces too much errors.

Note that Biometric Error Rates are hard to avoid. The quality of the sensors and the matching algorithm can separate for the better the matching and non-matching curves, *i.e.* render a better $\lambda_{max} - \lambda_{min}$. Moreover, the larger this difference is, the smaller p_2 and $1 - p_1$ can be computed, thus the smaller ϵ can be.

Conclusion

This chapter details the first non-trivial construction for biometric identification over encrypted binary templates. This construction meets the privacy model one can expect from an identification scheme. While the aim was deliberately to design a scheme for secure biometric identification, the design of the construction was general enough for the underlying cryptographic primitive, Error-Tolerant Searchable Encryption, to be used in other contexts than biometrics.

An essential tool for the establishment of such a protocol is the existence and implementation of LSH functions. On one hand, working with real biometric data would lead to a careful choice of the LSH-functions underlying our construction. These LSH-functions would determine the maximum error tolerance of our scheme. On the other hand, it would be interesting to conduct these experimental studies with various biometric modalities. We referred to the IrisCode, which can be expressed in the Hamming space and whose distortion is quite well controlled. Face biometrics which are compared via Euclidean distance seem to be adapted to LSH as well. However, some biometrics like the fingerprint are more sensible to distortions and do not provide an easy metric to be exploited by the LSH strategy. Whether the proposed construction can still handle such biometric features is an open topic.

The next chapter focuses on providing a construction of a radically different type, using symmetric cryptography. Indeed, in order to lighten the computation weight over the server, a possible solution is to use Searchable Symmetric Encryption rather than PIR-based protocols.

Chapter VI

Secure Identification in a Symmetric Setting

> Symmetry is a
> complexity-reducing concept;
> seek it everywhere.
>
> Alan Perlis

The theoretical and practical constructions provided in the previous chapter have very interesting privacy properties; unfortunately, it happens that their scalability is a real issue. Indeed, we avoided doing $\mathcal{O}(n)$ secure matching operations by applying a Bloom filter, thus reducing the number of secure matching. However, the model still requires to process the results through a PIR, and this operation is linear in the size M of the database. Recalling Proposition V.2, we see that it is necessary to have a large M in order to have a scheme ϵ-sound with ϵ small indeed. To avoid this pitfall, we focus here, as it was done in [2], on symmetric cryptography.

Symmetric Searchable Encryption (**SSE**) is a primitive not unlike PEKS; however, the fact that there is no public key changes the privacy model. We show in this chapter that when the privacy requirements can be relaxed to tolerate symmetric cryptography, then we can build even faster schemes, with different tools.

Recent works on Symmetric Searchable Encryption [12, 19, 50, 53, 78, 160] provide schemes with constant-time access to servers. The price to pay is a leakage of the **search pattern**: the server can tell whether a word was queried twice and can even recover links between documents and words. This enables to infer relations between requests and for instance to determine, after a statistical survey, the queried word. We formalize this advantage in the adversarial model stated in Section VI.2. In particular, Condition 15 is a barrier to statistical attacks. To cope with this classical weakness, we introduce a way to protect the access pattern on the server's side.

Here, we solve again the issue of preserving privacy in a biometric identification system in a way such that the computational cost for this purpose that is quite low. We make use of recent advances in the fields of similarity searching and secure querying on a remote server; in the end, we perform biometric identification over a wholly encrypted database, in such a way that the server does not have an advantage over the users' privacy.

Here again, we restrict ourselves to the case where biometric templates are in the Hamming space $B = \{0,1\}^n$ with the Hamming distance d. Two templates b, b' of a same user $\mathcal{U}$ are with a high probability at distance $d(b,b') < \lambda_{min}$. Similarly, when b and b' comes from different users, they are with a high probability at distance $d(b,b') > \lambda_{max}$. That means that once again, the matching algorithm consists in evaluating a Hamming distance.

VI.1 Symmetric Searchable Encryption - SSE

Searchable Encryption is described as follows.

- A client $\mathcal{U}$ has a collection of documents consisting of sequences of words.
- He encrypts the whole collection along with some indexing data.
- He stores the result on a (remote) server.

The server should be able to return all documents which contain a particular keyword, without learning anything about the aforementioned keyword.

Let $\Delta = \{\omega_1, \cdots, \omega_d\}$ be the set of d distinct words (typically a dictionary). A *document* $D \in \Delta^*$ is a sequence of words of Δ. The *identifier* `id`*(D)* is a bitstring that uniquely identifies the document D (e.g. its memory address). A *collection* $\mathcal{D} = (D_1, \cdots, D_n)$ is a set of n documents. $\mathcal{D}(\omega)$ denotes the lexicographically ordered list of identifiers of documents which contains the word ω.

Remark 32. We stated in chapter V that the way we do searchable encryption differs from previous work as we do not use keywords, but the document itself as referring data. This idea is used here again, as the previous definition is about documents that contains only keywords $\omega \in \Delta$. It requires only a trivial adjustment to allow documents to be any kind of data, indexed by these keywords.

Here is defined the **symmetric** searchable encryption paradigm.

Definition 10 (Symmetric Searchable Encryption Scheme [53]). A Symmetric Searchable Encryption scheme is a collection of four polynomial-time algorithms `Keygen`, `BuildIndex`, `Trapdoor`, `Search` such that:

`Keygen` (1^ℓ) is a probabilistic key generation algorithm, run by the client to setup the scheme. It takes a security parameter ℓ and returns a secret key K.

`BuildIndex` $(K,\mathcal{D})$ is a (possibly probabilistic) algorithm run by the client to compute the index $\mathcal{I}_\mathcal{D}$ of the collection $\mathcal{D}$. It takes as entry a secret key K and a collection of documents $\mathcal{D}$. The index returned allows the server to search for any keyword appearing in $\mathcal{D}$.

`Trapdoor` (K, ω) is a deterministic algorithm which generates a trapdoor T_ω for a given word ω under the secret key K. It is perfomed by the client whenever he wants to search securely for all the documents where ω occurs.

`Search` $(\mathcal{I}_\mathcal{D}, T_w)$ is run by the server to search in the entire collection $\mathcal{D}$ for all the documents identifiers where the queried word ω appears. It returns $\mathcal{D}(\omega)$.

These primitives give a functional aspect of what Symmetric Searchable Encryption provides. The associated security model is described in [53], and briefly depicted in section VI.1. The goal is to achieve *Adaptive Indistinguishability*, a security property stating that an adversary does not get information on the content of the registered documents. More precisely, if two different collections are registered, with constraints on the number of words per document, an adversary cannot distinguish between two sequences of search requests.

Remark 33. A noteworthy construction of a scheme adaptively indistinguishable was also provided in [53] (*cf.* section VI.1), and inspired the following identification data structure. Although this scheme is proved secure in their model, this does not cover statistical attacks where an adversary tries to break the confidentiality of the documents or the words based on statistics about the queried words and the index (*cf.* Remark 36).

Security Model Associated to Symmetric Searchable Encryption

The following model for Symmetric Searchable Encryption was proposed in [53]. We briefly state the requirements and provide the construction given by the authors to comply with the model.

Security model for Symmetric Searchable Encryption

A **history** H_q is an interaction between a client and a server over q queries, consisting of a collection of documents $\mathcal{D}$ and q keywords $\omega_1, \cdots, \omega_q$. Let $\mathcal{D}$ be a collection of n documents $(D_1, \cdots, D_n)$, and let `Enc` be an encryption function. If the documents of $\mathcal{D}$ are stored encrypted by `Enc`, and $H_q =$

$(\mathcal{D}, \omega_1, \cdots, \omega_q)$ is a history over q queries, an adversary's **view** of H_q under the secret key K is defined as

$$V_K(H_q) = (\boldsymbol{id}(D_1), \ldots, \boldsymbol{id}(D_n), \mathtt{Enc}_K(D_1), \ldots, \mathtt{Enc}_K(D_n), \mathcal{I}_\mathcal{D}, T_{\omega_1}, \ldots, T_{\omega_q})$$

The History and the View of an interaction determine what did an adversary obtain after a client executed the protocol; an estimation of the information leaked is given by the Trace.

Let $H_q = (\mathcal{D}, \omega_1, \ldots, \omega_q)$ be a history over q queries. The **trace** of H_q is the sequence

$$Tr(H_q) = (\boldsymbol{id}(D_1), \ldots, \boldsymbol{id}(D_n), |D_1|, \ldots, |D_n|, \mathcal{D}(\omega_1), \ldots, \mathcal{D}(\omega_q), \Pi_q)$$

where Π_q is a symmetric matrix representing the access pattern, *i.e.*

$$\Pi_q[i, j] = \delta_{\omega_i, \omega_j}.$$

For such a scheme, the security definition is the following.

Definition 11 (Adaptive Indistinguishability Security for SSE [53])**.** A SSE scheme is said to be *adaptively indistinguishable* if for all $q \in \mathbb{N}$, for all probabilistic polynomial-time adversaries $\mathcal{A}$, for all traces Tr_q of length q, and for all polynomially sampleable distributions

$$\mathcal{H}_q = \{H_q \ : \ Tr(H_q) = Tr_q\}$$

(the set of all histories of trace Tr_q), the advantage $\mathtt{Adv}_\mathcal{A} = \left|\Pr[b' = b] - \frac{1}{2}\right|$ of the adversary is negligible.

```
$\mathtt{Exp}_\mathcal{A}^{\mathtt{IND}}$

1. $K$                                 $\leftarrow$  $\mathtt{Keygen}(1^k)$                                          $(\mathcal{C})$
2. $(\mathcal{D}_0, \mathcal{D}_1)$    $\leftarrow$  $\mathcal{A}$                                                    $(\mathcal{A})$
3. $b$                                 $\overset{R}{\leftarrow}$  $\{0,1\}$                                            $(\mathcal{C})$
4. $(\omega_{1,0}, \omega_{1,1})$      $\leftarrow$  $\mathcal{A}(\mathcal{I}_b)$                                     $(\mathcal{A})$
5. $T_{\omega_{1,b}}$                  $\leftarrow$  $\mathtt{Trapdoor}(K, \omega_{1,b})$                            $(\mathcal{C})$
6. $(\omega_{i+1,0}, \omega_{i+1,1})$  $\leftarrow$  $\mathcal{A}(\mathcal{I}_b, T_{\omega_{1,b}}, \ldots, T_{\omega_{i,b}})$  for $i = 1, \ldots, q-1$  $(\mathcal{A})$
7. $T_{\omega_{i+1,b}}$                $\leftarrow$  $\mathtt{Trapdoor}(K, \omega_{i+1,b})$                          $(\mathcal{C})$
8. $b'$                                $\leftarrow$  $\mathcal{A}(V_K(H_b))$                                         $(\mathcal{A})$
```

In this experiment, the attacker begins by choosing two collections of documents (2.), which each contains the same number of keywords; then the challenger follows by flipping a coin b (3.), and the adversary receives the index of one of the collections $\mathcal{D}_b$; he then submits two words $(\omega_{1,0}, \omega_{1,1})$ (4.) and receives the trapdoor for $\omega_{1,b}$ (5.). The process goes on until the adversary has submitted q queries (6. and 7.) and he is challenged to output b (8.).

Algorithm 5 Adaptively secure SSE construction [53]

```
Keygen(1^k): Generate a random key $K \overset{R}{\leftarrow} \{0,1\}^k$

BuildIndex ($K$, $\mathcal{D}$):

  • Initialization:

      - scan $\mathcal{D}$ and build $\Delta$, the set of distinct words in $\mathcal{D}$.
      - for each $\omega \in \Delta$, build $\mathcal{D}(\omega) = \{D_\omega^j\}_j$
      - compute $\mathtt{max} = \max_{\omega\in\Delta}(|\mathcal{D}(\omega)|)$ and $m = \mathtt{max} \cdot |\Delta|$

  • Build look-up table T :

      - for each $\omega \in \Delta$
            for $1 \leq j \leq |\mathcal{D}(\omega)|$
                set T $[\pi_K(\omega \parallel j)] = \mathtt{id}(D_\omega^j)$
      - if $m' = \sum_{\omega\in\Delta} |\mathcal{D}(\omega)| < m$, then set the remaining $(m - m')$
        entries of T to identifiers of documents $\mathtt{id}(D_r)$, $r \in [\![1, n]\!]$ such
        that the same identifier holds for the same number of entries.

  • Output $\mathcal{I}_\mathcal{D} = \mathtt{T}$

Trapdoor ($K, \omega$): Output $T_\omega = (\pi_K(\omega \parallel 1), \ldots, \pi_K(\omega \parallel \mathtt{max}))$

Search ($\mathcal{I}_\mathcal{D}, T_\omega$): For $1 \leq i \leq \mathtt{max}$: retrieve $\mathtt{id} = \mathcal{I}_\mathcal{D}\,[T_\omega[i]]$
```

SSE Construction

The algorithms that implement the Symmetric Searchable Encryption in [53] are depicted in Algorithm 5. The scheme is proven indistinguishable against adaptive adversaries.

For this construction, a pseudo-random permutation noted π_K is used, where K is the secret key of the system. The security of this scheme rests on the indistinguishability of this pseudo-random permutation which ensures the indistinguishability of the sent data.

VI.2 Fast and Secure Biometric Identification

This chapter's construction does not simply mix a SSE scheme with a LSH family. Indeed, we ensure the security of this biometric identification protocol against statistical attacks, which is an improvement with respect to a direct combination of SSE with LSH.

The Idea in a Nutshell

This Biometric Identification process has two phases: a **search phase** which carries out every request on the database $\mathcal{DB}$ and sends back to the sensor client $\mathcal{SC}$ the search result, an **identification phase** which treats data extracted from search results to proceed to the identification. The search phase is constructed following the principle of the SSE scheme from [53]. The following entities interact:

- Human users $\mathcal{U}_i$: a set of N users who register their biometrics.
- Sensor client $\mathcal{SC}$: a device that captures the biometric data and extracts its characteristics to output the biometric template. It also sends queries to the server to identify a user.
- The server: replies to queries sent by $\mathcal{SC}$ by providing a set of search results and owns a database $\mathcal{DB}$ to store the data related to the registered users.

Remark 34. We consider that $\mathcal{SC}$ is honest and trusted by all other components. In particular, $\mathcal{SC}$ is the only entity which is in possession of the cryptographic key material used in the protocol. To justify this assumption, we emphasize that the object of this chapter is to provide a solution to the secure storage of reference templates, but not to provide an end-to-end architecture. See Remark 37 for details on key management.

We provide the three following methods:

1. $\texttt{Initialize}(1^\ell)$: It produces the parameters $\mathcal{K}$ of the system, according to a security parameter ℓ. $\mathcal{K}$ must contain secret keys sk used to encrypt the identities, and K used in the SSE scheme.
2. $\texttt{Enrolment}(b_1, \ldots, b_N, ID_1, \ldots, ID_N, \mathcal{K})$: It registers a set of users with their biometric characteristics. For a user $\mathcal{U}_i$, it needs a biometric sample b_i and his identity ID_i. This returns an index $\mathcal{I}$.
3. $\texttt{Identification}(\mathcal{K}, b)$: It takes as input a newly captured template b and it returns a set of identities for which the associated templates are close to b. See Conditions 13 and 14, Section VI.2.

Definition 12. In our proposal, *keywords* are evaluations of LSH functions on templates, concatenated with the index of the considered function, *i.e.* $(h_i(b), i)$, for $i \in [\![1, \mu]\!]$ where b is the captured template of a user.

Identifiers are the encryptions of the *identities* of the registered users. We have, $\texttt{id}(\mathcal{U}_i) = \mathcal{E}_{sk}(ID_i) \quad$ for $i \in [\![1, N]\!]$ where $\mathcal{E}_{sk}$ is an encryption function with the secret key sk, and ID_i is the identity of the user $\mathcal{U}_i$.

The interaction between the server and $\mathcal{SC}$ defines the identification view, required for the security experiments. It consists of the encrypted identities of the registered users, and informations sent by $\mathcal{SC}$ when a user $\mathcal{U}$ is being identified.

Definition 13 (Identification View)**.** The *identification view* under the secret keys K and sk is defined as

$$IdV_{K,sk}(b') = (\mathcal{I}, T_{(h_1(b'),1)}, \ldots, T_{(h_\mu(b'),\mu)}, \mathcal{E}_{sk}(ID_1), \ldots, \mathcal{E}_{sk}(ID_N))$$

where b' is a freshly captured template from $\mathcal{U}$.

Security Requirements

We assume that the Hamming space $B = \{0,1\}^n$ is such that $n \geq \ell$, where ℓ is the security parameter.

As we did in Chapter V.2 Condition 9, we define the completeness and soundness of the system.

Condition 13 (Completeness)**.** The system is *complete* if for all $b' \in B$, the result of $\texttt{Identification}(\mathcal{K}, b')$ contains the set of identities for which the associated templates b_i are close to b' (*i.e.* $d(b', b_i) < \lambda_{min}$), except for a negligible probability.

Condition 14 (Soundness)**.** The system is sound if, for each template b' such that $d(b', b_i) > \lambda_{max}$, $\texttt{Identification}(\mathcal{K}, b')$ is the empty set $\emptyset$, except with negligible probability.

To avoid statistical attacks, we do not want the database to infer relations between different identities. This is formalized by the following condition.

Condition 15 (Adaptive Confidentiality)**.** An identification system achieves adaptive confidentiality if the advantage $\texttt{Adv}_{\mathcal{A}} = |\Pr(b_0 = b'_0) - \frac{1}{|B|}|$ of any polynomial-time adaptive adversary is negligible in the next experiment, where $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ is an opponent taking the place of the server, and $\mathcal{C}$ is a challenger at $\mathcal{SC}$'s side.

$$\begin{array}{|llcll}
1. & \mathcal{K} & \overset{R}{\longleftarrow} & \texttt{Initialize}(1^\ell) & (\mathcal{C}) \\
2. & b_1, \ldots, b_N & \longleftarrow & B & (\mathcal{A}) \\
3. & \mathcal{I}_1 & \longleftarrow & \texttt{Enrolment}(b_1, \ldots, b_N) & (\mathcal{C}) \\
4. & b, IdV_{K,sk}(b) & \longleftarrow & \mathcal{A}_1^{\texttt{Identification}}(\mathcal{I}_1) & (\mathcal{A}) \\
5. & b_0 & \overset{R}{\longleftarrow} & B & (\mathcal{C}) \\
 & \multicolumn{3}{l}{\text{such that } \forall i \in [\![1, N]\!], d(b_0, b_i) > \lambda_{max}} & \\
6. & \mathcal{I}_2 & \longleftarrow & \texttt{Enrolment}(b_0, b_1, \ldots, b_N) & \\
6'. & b, IdV_{K,sk}(b) & \longleftarrow & \mathcal{A}_1^{\texttt{Identification}}(\mathcal{I}_1, \mathcal{I}_2) & (\mathcal{A}) \\
7. & b'_0 & \longleftarrow & \mathcal{A}_2(\mathcal{I}_1, \mathcal{I}_2, b, IdV_{K,sk}(b), IdV_{K,sk}(b_0)) &
\end{array}$$

$\texttt{Enrolment}(b_1, \ldots, b_N)$ stands for $\texttt{Enrolment}(b_1, \ldots, b_N, ID_1, \ldots, ID_N, K, sk)$.

In this game, the attacker is allowed to set a templates database $b_1, \ldots, b_N$ of its choice (2.). Then the challenger creates the database by enrolling the whole collection (3.), and the adversary can make a polynomial number of identifications (using the method `Identification`) of the templates of his choice (4.). The challenger then picks a random template b_0 (5.) and it recreates the database $\mathcal{I}_2$ (6.). The attacker is allowed once again to make a polynomial number of identifications from the templates of its choice (6′.) and he is challenged to retrieve the initial template b_0 (7.), given the knowledge of $\mathcal{I}_1, \mathcal{I}_2$, and the views of the identifications.

The next condition expresses the confidentiality of the enrolled templates, even if the adversary has access to the index and to identification views, which may give him the possibility to construct a statistical model on it.

Condition 16 (Non-adaptive Indistinguishability)**.** We say that a biometric identification system achieves indistinguishability if the advantage $\mathtt{Adv}_{\mathcal{A}} = |\Pr(e = e') - \frac{1}{2}|$ of any polynomial-time adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ is negligible in the following game:

$$\left|\begin{array}{llll} 1.\ b_1, \ldots, b_N & \xleftarrow{R} & B & (\mathcal{C}) \\ 2.\ b^{(0)}, b^{(1)} & \longleftarrow & \mathcal{A}_1(\mathcal{C}(IdV_{K,sk})) & (\mathcal{A}) \\ 3.\ e & \xleftarrow{R} & \{0,1\} & (\mathcal{C}) \\ 4.\ e' & \longleftarrow & \mathcal{A}_2(IdV_{K,sk}(b^{(e)})) & (\mathcal{A}) \end{array}\right.$$

$\mathcal{A}_1(\mathcal{C}(IdV_{K,sk}))$ stands for the fact that the adversary $\mathcal{A}_1$ has access to the identification view produced when $\mathcal{C}$ executes a polynomial number of identification requests, without knowing the input randomly chosen by the challenger.

This experiment is executed as follows: The challenger first creates a set of templates $b_1, \ldots, b_N$ (1.), and executes a polynomial number of identification requests. The adversary has access to all the identification views (2.). The attacker then chooses two templates for which he believes he has an advantage (2.), and the challenger picks at random one of them and executes its identification (3.). The attacker is finally challenged to determine which template the challenger chose (4.).

Our Identification Protocol

<u>`Initialize`</u>(1^ℓ):

- We choose an IND-CPA symmetric encryption scheme $(\mathcal{G}, \mathcal{E}, \mathcal{D})$.
- We use the Symmetric Searchable Encryption scheme from [53] (see Appendix VI.1 for the construction detail) out of which we pick the functions (`Keygen`, `Trapdoor`, `Search`) and adapt them to our needs.
- We fix a threshold $0 < \lambda \leq \frac{1}{2}$.

- Let $H = (h_1, \dots, h_\mu)$ be a $(\lambda_{min}, \lambda_{max}, p_1, p_2)$- LSH family, $\mu \geq \ell$.
- Let $K = \texttt{KeyGen}(1^\ell)$, and $sk = \mathcal{G}(1^\ell)$.
- Let π_K be the pseudo-random permutation indexed by the key K used in the SSE scheme.

Output $\mathcal{K} = (h_1, \dots, h_\mu, K, sk, \lambda)$.
<u>`Enrolment`</u>$(b_1, \dots, b_N, ID_1, \dots, ID_N, \mathcal{K})$: Consider N users $\mathcal{U}_1, \dots, \mathcal{U}_N$ to be enrolled. Their template are denoted by b_i, and their identity ID_i, $i \in \llbracket 1, N \rrbracket$. We recall that in our construction, the words we consider are the $(h_i(b), i)$, $i \in \llbracket 1, \mu \rrbracket$, $b \in B$, where h_i is one of the chosen LSH functions, and b is a reference template from a registered user.

We alter the `BuildIndex` algorithm of the SSE scheme into `Enrolment` to take into account the need for identification. The result is Algorithm 6.

Algorithm 6 Enrolment Procedure

```
Enrolment($b_1, \dots, b_N, ID_1, \dots, ID_N, \mathcal{K}$):

  • Initialization:
      - build $\Delta = \{(h_i(b_k), i); \quad i \in \llbracket 1, \mu \rrbracket,\ k \in \llbracket 1, N \rrbracket\}$
      - for each $\omega \in \Delta$, build $\mathcal{D}(\omega) = \{ID_\omega^j\}_j$ the set of identifiers of users
        $\mathcal{U}_k$ such that $(h_i(b_k), i) = \omega$
      - compute $\texttt{max} = \max_{\omega \in \Delta}(|\mathcal{D}(\omega)|)$ and $m = \texttt{max} \cdot |\Delta|$

  • Build look-up table T:
      - for each $\omega \in \Delta$
            for $1 \leq j \leq |\mathcal{D}(\omega)|$
                set T $[\pi_K(\omega \parallel j)] = \mathcal{E}_{sk}(ID_\omega^j)$
      - if $m' = \sum_{\omega \in \Delta} |\mathcal{D}(\omega)| < m$, then set the remaining $(m - m')$ entries
        of T to random values.

  • Output $\mathcal{I} = \texttt{T}$
```

Remark 35. Our scheme stores identifiers encrypted by an IND-CPA scheme so that no relation between the entries could be found by observing the index $\mathcal{I}$. This prevents inferring statistics from the $\mathcal{DB}$ content. Proposition VI.3 formalizes this intuition.

<u>`Identification`</u>$(\mathcal{K}, b')$:
Search phase: When a user $\mathcal{U}$ wants to be identified, $\mathcal{SC}$ captures his biometric trait in a template b'. $\mathcal{SC}$ evaluates each LSH function on b' to compute $\omega_i = (h_i(b'), i)$, $i \in \llbracket 1, \mu \rrbracket$ and sends to the server the trapdoors:

$$T_{\omega_i} = \texttt{Trapdoor}(K, \omega_i) = (\pi_K(\omega_i, 1), \dots, \pi_K(\omega_i, \max))$$

The server executes the `Search` algorithm on the different trapdoors T_{ω_i} – each call to $\texttt{Search}(t_1, \dots, t_{\max})$ returns $\texttt{T}[t_1], \dots \texttt{T}[t_{\max}]$ – and sends to $\mathcal{SC}$ the array $\mathcal{ID}(b')$ which corresponds to all the search results:

$$\mathcal{ID}(b') = \begin{bmatrix} \mathcal{E}_{sk}(ID_{k_1,1}) & \cdots & \mathcal{E}_{sk}(ID_{k_1,\mathtt{max}}) \\ \vdots & \ddots & \vdots \\ \mathcal{E}_{sk}(ID_{k_\mu,1}) & \cdots & \mathcal{E}_{sk}(ID_{k_\mu,\mathtt{max}}) \end{bmatrix}$$

where each row is made of the output of $\texttt{Search}(T_{\omega_i})$. It may happen that a virtual address $\pi_K\big(h_i(b'), i, j\big)$ is invalid, in this case the server sends $\perp$ instead of an identifier.

Identification phase: $\mathcal{SC}$ decrypts all received identifiers and determines the number of occurrences of each identity to output the list of the ones that appear more than $\lambda\mu$ times, *i.e.* the list of identities $\{ID(\mathcal{U}_l)\}$ that verify this inequality: $\sum_{i=1}^{\mu}\sum_{j=1}^{\mathtt{max}} \delta_{ID(\mathcal{U}_l),ID_{k_i,j}} > \lambda\mu$. If the result is still ambiguous after that the identity that appeared the most was selected, an empirical rule is applied.

Security Properties

We use Shannon's binary entropy[1], $\mathcal{H}_2(\lambda) = \lambda \cdot log\frac{1}{\lambda} + (1-\lambda)\cdot log\frac{1}{1-\lambda}$.

Proposition VI.1 (Completeness)**.** *Provided that H is a $(\lambda_{min}, \lambda_{max}, p_1, p_2)$-LSH family, for $1 - p_1 \leq \frac{1}{4^{\mathcal{H}_2(\lambda)+c}}$, with $c \geq 1$, our scheme is complete.*

> **Proof** Let $\mathcal{U}$ be a registered user to be identified, with reference template b and identity $ID(\mathcal{U})$. Let b' be a freshly captured template such that $d(b,b') < \lambda_{min}$. The scheme is complete if the probability for $ID(\mathcal{U})$ not to be returned is negligible, *i.e.* if $ID(\mathcal{U})$ appears less than $\lambda\mu$ times in $\mathcal{ID}(b')$.
>
> Let us consider the event E_i : "$\mathcal{E}_{sk}(ID(\mathcal{U}))$ does not appear in row i of $\mathcal{ID}(b')$". E_i happens if and only if $h_i(b'), i, j \neq h_i(b), i, j$, *i.e.* with probability $1 - p_1$. Then, the probability for the scheme not to be complete is given by: $\Pr[ID(\mathcal{U})$ appears in less than $\lfloor\lambda\mu\rfloor$ positions $] = \sum_{i=0}^{\lfloor\lambda\mu\rfloor}\binom{\mu}{i}p_1^i(1-p_1)^{\mu-i}$. But, considering $1 - p_1 \leq \frac{1}{4^{\mathcal{H}_2(\lambda)+c}}$, we have:
>
> $$(1-p_1)^{\mu-i} \leq \frac{1}{4^{(\mathcal{H}_2(\lambda)+c)(\mu-i)}} \leq \frac{1}{4^{(\mathcal{H}_2(\lambda)+c)(\frac{\mu}{2})}} = \frac{1}{2^{\mu(\mathcal{H}_2(\lambda)+c)}}$$
>
> .
>
> Thus, $\sum_{i=0}^{\lfloor\lambda\mu\rfloor}\binom{\mu}{i}p_1^i(1-p_1)^{\mu-i} \leq \sum_{i=0}^{\lfloor\lambda\mu\rfloor}\binom{\mu}{i}(1-p_1)^{\mu-i} \leq (\lfloor\lambda\mu\rfloor + 1)\,.\frac{2^{\mu\mathcal{H}_2(\lambda)}}{2^{\mu(\mathcal{H}_2(\lambda)+c)}} \leq (\lfloor\lambda\mu\rfloor + 1)\,.\frac{1}{2^{c\mu}}$ which is negligible. This proves the result. □

[1] The notation $\mathcal{H}_2$ is used here in order not to think of $\mathcal{H}_2$ as a hash function.

Proposition VI.2 (Soundness)**.** *Provided that H is a $(\lambda_{min}, \lambda_{max}, p_1, p_2)$-LSH family, for $p_2 \leq \frac{1}{2^{\frac{1}{\lambda}+c}}$, with $c \geq 1$, our scheme is sound.*

Proof
Let b' be a freshly captured template such that $d(b,b') > \lambda_{max}$ for any registered template b. The system returns an identity if and only if one identity appears in at least $\lceil \lambda\mu \rceil$ entries. This implies that for at least $\lceil \lambda\mu \rceil$ LSH functions h, we have, $h(b) = h(b')$. Given a hash function, and regarding the definition of a LSH family, this occurs with a probability p_2. So, $\Pr[\texttt{Identification}(\mathcal{K}, b') \neq \emptyset] = \sum_{i=\lceil \lambda\mu \rceil}^{\mu} \binom{\mu}{i} p_2^i (1-p_2)^{\mu-i} \leq 2^\mu \cdot p_2^{\lambda\mu}$. If $p_2 \leq \frac{1}{2^{\frac{1}{\lambda}+c}}$, this probability is negligible too. This gives the result. □

The underlying idea of these two proofs is that computing the μ LSH functions separates the close and the distant template pairs.

Proposition VI.3 (Adaptive Confidentiality)**.** *Provided that the underlying encryption scheme $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ is a IND-CPA secure scheme, our construction ensures the templates confidentiality.*

Proof The adversary $\mathcal{A}$ is allowed to execute some identification requests. If $\mathcal{A}$ is able to reconstruct the template b_0, then he can infer links on the enrolled b_i and the identification result $\mathcal{ID}(b_0)$.

Due to the IND-CPA security of $(\mathcal{G}, \mathcal{E}, \mathcal{D})$, a simulator can simulate the array $\mathcal{ID}(b)$ during the second enrolment phase in the following way: when it receives for the first time a set of trapdoors $\{T_{h_1(b,1)}, \ldots, T_{h_\mu(b,\mu)}\}$, for a template b, it picks up a random array of size $\mu \cdot \texttt{max}$ and stores the correspondence between the trapdoors and this array. When the adversary sends the same trapdoors, the same result is sent back by the simulator. This way, an adversary who can link information contained in the array $\mathcal{ID}(b)$, can also infer links on random identifiers, which is impossible. Thus the property. □

Proposition VI.4 (Non-Adaptive Indistinguishability)**.** *Provided that π_K is a pseudo-random permutation depending on a secret key K, and that $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ is semantically secure, our construction ensures the non-adaptive indistinguishability.*

Proof This property is mainly a consequence of the semantic security of the SSE scheme we consider. Indeed, for π_K a pseudo-random permutation, a simulator can simulate the trapdoors sent by the sensor client during an identification, and it can also simulate the server's response because of the semantic security of the symmetric encryption scheme used.

□

Remark 36. We emphasize that the aforementioned properties define an adequate description of what resistance against statistical attacks would be.

A scheme, that would be no more than a combination of the SSE scheme described in [53] with the use of LSH functions, would not be resistant against these methods. An adversary is, in that setting, able to retrieve – and compare – the identifiers of the users enrolled, and thus infer knowledge on the identity of a user that did not proceed to identification.

Similarly, if the identifiers are not encrypted, an attacker who observes the views of identification can gather statistics on the identification of the different users. This enables him to link the identity of users - as some are more likely to be identified than others - with the response of the server. Moreover, he can manage a very general statistical attack in that case: by learning the relation between identities and keywords (i.e. LSH values of biometric data), he can even reconstruct unknown templates.

Note that our technique to thwart statistical attacks is quite general and can be reused in other contexts.

VI.3 Practical Considerations

Choosing a LSH Family

To explain that our scheme meets the usual needs for a practical deployment of a biometric identification system, let us consider again the case of biometric iris recognition as a practical example, with the IrisCode algorithm (chapter II). The LSH functions of Hao *et al.* [88] presented in the previous chapter are here considered to estimate the soundness and completeness.

The family is made by $\mu = 128$ hash functions, which each extracts a 10-bit vector. Our parameter λ can be set to $\lambda = \frac{3}{128}$. According to traditional matching algorithms, we can choose $\lambda_{min} = 0.25 \cdot 2048 = 512$ and $\lambda_{max} = 0.35 \cdot 2048 = 716.8$, which gives the probabilities $p_1 \simeq 0.056$ and $p_2 \simeq 0.013$ (with the notations of Definition 6). The probability of $\texttt{Identification}(b')$ *not* returning a template close to b' is given by $\sum_{i=0}^{\lfloor \lambda\mu \rfloor} \binom{\mu}{i} p_1^i \, (1-p_1)^{\mu-i} \simeq 0.066$ and the other probability to consider is, for b' far from all the b_i, $\Pr[\texttt{Identification}(b') \neq \emptyset] = \sum_{i=\lceil \lambda\mu \rceil}^{\mu} \binom{\mu}{i} p_2^i (1-p_2)^{\mu-i} \simeq 0.095$. Note that those probabilities are small, and not negligible, but they can be considered attractive for practical uses (as asserted by the results from [88]) .

Implementation

To check further the feasibility of our scheme, we implemented our scheme and conduced a first empirical evaluation on the ICE 2005 database (described in I.1) which contains 2953 images from 244 different eyes. The results are similar to those deduced in the previous section from the results of [88]. For instance, the probability that the genuine identity is not in the output list of candidates is below 10%.

Remark 37. In addition to this performance consideration, it is important to notice that the deployment of the scheme is quite simple as only the client needs to know the secret keys. So management of the keys is reduced to a distribution to the clients that are allowed to run identification requests onto the remote server.

Complexity

We here evaluate the computational complexity of an identification request on the server's side as well as on $\mathcal{SC}$. We note $\kappa(op)$ the cost of operation *op*.

- On the server's side: assuming that we organize the look-up table in a FKS dictionnary [67], a search is made in constant time and the server has μ searches to achieve.
- On $\mathcal{SC}$'s side:

$$\begin{aligned} \kappa(identification) &= \kappa(trapdoors) + \kappa(count) \\ &= \mu.\texttt{max}.\left[\kappa(hash) + \kappa(encryption) + \kappa(decryption)\right] \end{aligned}$$

$\kappa(hash)$ is the computational complexity to evaluate a LSH function, and $\kappa(encryption)$ is the one to apply the pseudo-random permutation π_K.

The final count needs to compute the number of occurences of each identity, it can be made in computation time linear in the size of the final array, hence the term $\mu.\texttt{max}.\kappa(decryption)$ (remember that before counting, $\mathcal{SC}$ has to decrypt the search results).

If the chosen hash functions map $\{0,1\}^*$ to $\{0,1\}^m$ (for $m \in \mathbb{N}^*$) and assuming that images of these functions are equally distributed, the max value can be bounded by $\Omega(\frac{N}{2^m})$, where N is the number of registered users. So the overall complexity is $\Omega\left(\mu\frac{N}{2^m}\right)\cdot[\kappa(hash) + \kappa(encryption) + \kappa(decryption)]$. A traditional identification algorithm would cost $\mathcal{O}(N)$ matching operations; with the parameters given in section VI.3, our solution is 8 times more efficient, with the additional benefit of the encryption of the data.

Remark 38. The complexity of the construction proposed in chapter V was globally the same at the client level (modulo the use of asymmetric cryptography rather than symmetric schemes in our case). It consists in computing the LSH images of the freshly acquired template, and in preparing μ PIR queries associated to the hashes. While this computation is costly, it is still doable in reasonable time. However on the server side, $\mathcal{S}$ must compute the PIR replies, and cannot do it in less than a linear time in the database's size (2^m). Indeed, no matter what PIR scheme is used, $\mathcal{S}$ always needs to process the whole database before sending its reply; here we enable secure biometric identification with only μ constant-time operations at $\mathcal{S}$'s side.

Conclusion

Switching from asymmetric to symmetric cryptography imposes to change security models and paradigms. The whole range of tools that we used in the asymmetric setting is no longer available (among them, PIR and PIS, recall that the goal of this exploration was not to use them).

It is still possible to design interesting protocols with only symmetric primitives. As in chapter V, the LSH functions are at the heart of the protocols designed, so once again, extending these schemes to minutiae-based fingerprint identification raises issues.

In the next chapter, we go further, and use traditional matching algorithm in a secure way. The tool for that is to use a dedicated hardware in order to make the sensitive computations in a protected location.

Chapter VII

Bio-Cryptography: Concluding Elements

> One should not confuse that which appears to us to be improbable and unnatural with that which is absolutely impossible.
>
> Carl Friedrich Gauss

To conclude this part, we here sum up the different cryptographic constructions that were proposed in chapter V and VI. We show the common divisors of all the techniques used to finally get the identity of a user. This will be the opportunity to present an alternative construction, more flexible than the previous ones, and whose security relies on physical hardware, in section VII.2.

The starting point of our reflection was the finding out that a secure biometric identification protocol that involves cryptography requires a representation of the biometrics that is adapted to cryptographic methods. That is why we looked in section II.1 for a binary representation of the fingerprints and the iris that would also be of fixed length[1].

The downside of this representation is that quantization decreases the overall biometric performances in terms of error rates (and computation time, even if that was not our major concern). Ounce we found out in section II.3 that the identification mode's accuracy rates cannot be interesting if the authentication error rates are high, we focused, in all of our constructions, in proposing two-step identification protocols:

1. Selecting a small number of candidates for identification, in such a way that the reference identity is among them;

[1] A representation whose weight is balanced is of course a plus.

2. Comparing each candidate's template with the fresh one for a precise result.

Remark 39. A biometric *authorization* system only needs to execute the first step, as we do not require to know the identity of a user, but only his belonging to a group.

VII.1 Identification Candidates: Operations on Quantized Data

Locality-Sensitive Hashing Functions The tool that was used in the previous chapters is LSH families (Defined in V.2). They are handful as they enable to reduce the dimension of templates - to speed up the search for a template. The advantages of using them are:

- they were designed to solve a class of problems, one of them being precisely ours. Indeed, once the biometric templates are in a stable form (*e.g.* a binary vector of fixed length, the proximity of two templates being captured by the Hamming distance), to find the candidates *is* to find the nearest neighbours of the binary template.

- as the probability of error of LSH functions is upper-bounded, it is possible to know the probability of missing a template, and the probability of falsely identifying one as a candidate (see Propositions V.1, V.2, VI.1 and VI.2). These probabilities enable to fine-tune the candidate list's size.

Using LSH functions is however interesting only if we do not wish to find the closest match by computing all the Hamming distances.

Template-by-template Comparison In order to find the element $b_{i_0} \in \{b_1, \ldots, b_n\}$ closest to b, it is easier to compute all the distances $d(b, b_i)$, and to select the template that minimises it.

We did not use such a method in the previous chapters because it is obviously not scalable. It also requires to compute the matching score and to find the largest of all these scores, all these operations in the encrypted domain. If we know how to compute a Hamming distance of encrypted templates (see chapter V.1), finding the largest one is a more difficult problem. Secure comparison is indeed possible, but requires multi-party computation, as shown by Rivest *et al.* [150].

However, if we alter the model and reduce the size n of the template list, then it becomes possible to select the candidates using this method. Such a construction is described in section VII.2.

VII.2 Methods for Secure Comparison

Physical or Functional Secure Comparison

Homomorphic encryption and secure comparison The cryptographic approach to secure comparison of two templates is to take two vectors in the encrypted domain, $< b >$ and $< b' >$, and return the encrypted score $< \mathsf{m}(b, b') >$. This is clearly a homomorphic operation, and it cannot be achieved with all kinds of encryption. Limiting ourself to binary strings, and Hamming-like distance, the arch-example of such an encryption is the Goldwasser-Micali cryptosystem.

We refer to section that deals with secure authentication (section V.1) for bibliography and description of methods inherent to this characteristic.

Hardware-based secure comparison If we do not wish to use mathematical functions to do a secure encryption, then it is necessary to compare templates in a secure hardware. We follow the example of payment terminals which rely on a Secure Access Module (**SAM**, think of a dedicated smartcard) to provide the secure storage functionality for secret elements.

This leads to an architectural issue: how is it possible to have a system in which the biometric data remains secure from the capture, to the storage, until it is used by a trusted matching device? We choose to execute the sensitive operation of matching a fresh biometric data against the biometric references of the registered users inside a **SAM** equipped with Match-On-Card (MOC) technology. To optimize the performances of our process, we reduce the number of these MOC comparisons by using a faster pre-comparison of biometric data based on their quantization.

As it was specified in section I.2, Match-On-Card is usually used for biometric authentication. In such a setting, a person is authenticated by first inserting a smartcard into a terminal, and then by presenting his biometrics. The biometric terminal sends the resulting template to the smartcard, which computes a matching score between the fresh template, and a previously stored one, and decides if the two templates come from the same user. Typically, a MOC fingerprint template is stored on about 512 bytes.

As the computing power is limited, the matching algorithms for Match-On-Card suffer from more restrictions than usual matching functions. However, the performances are still good enough for real-life applications. As an example, the NIST MINEX test [127] reports a False Reject Rate of 4.7 10^{-3} for a False Accept Rate of 10^{-2}, and a False Reject Rate of 8.6 10^{-3} for a False Accept Rate of 10^{-3}. More detailed results can be found on the project website.

The next section describes how to use MOC for biometric identification.

A Step by Step Implementation

Entities Involved

The system architecture depicted here tends to combine the efficiency of biometric recognition and the physical security of a hardware-protected component. In practice, we build a **biometric terminal**, (cf. Figure VII.1), that includes distinct entities:

- a main processing unit,
- a sensor,
- some non-volatile memory. This memory contains what we call the **encrypted database** which contains the encryption of all the templates of registered users,
- a **SAM** dedicated to the terminal. It can be physically attached to the terminal as a chip with connections directly weld to the printed circuit board of the terminal. Another possibility is to have a SIM card and a SIM card reader inside the terminal.

Afterwards, when we mention the computations of the biometric terminal, we designate those made by its main processor.

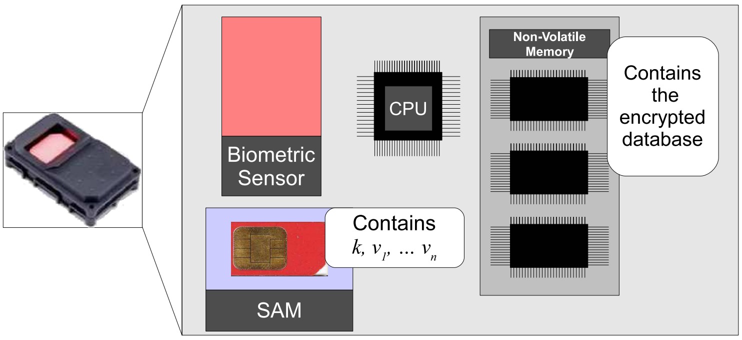

Biometric Sensor
CPU
Non-Volatile Memory
Contains the encrypted database
Contains $k, v_1, \ldots v_n$
SAM

Figure VII.1: Our Biometric Terminal

Remark 40. Coming back to the analogy with the payment terminals, we consider in the following that our terminal is tamper-evident [186]. Therefore, attempts of physical intrusions will be detected after.

Setup

We choose a symmetric encryption scheme, such as, for instance, the AES. It requires a cryptographic key κ which is kept inside the **SAM**. The **SAM** thus performs the encryption and decryption. The encryption of x under the key

κ is denoted by $\mathsf{Enc}(x)$ (we omit κ in order to lighten the notations). Note that no user owns the key κ: there is only one key, independent of the user.

We ensure the confidentiality of the templates by encrypting the content of the database under the key k. For n registered users, the database of the terminal stores their n encrypted templates $\{\mathsf{Enc}(b_1), \ldots, \mathsf{Enc}(b_n)\}$.

Identification through our proposal is made in two steps. To identify the owner of a biometric template b', we first roughly select a list of the most likely templates $(b_{i_1}, \ldots, b_{i_c})$ from the database, $c < n$. This is done by comparing quantized templates, as the comparison of these binary vectors is much faster than a MOC comparison. In a second step, the identification is comforted by doing the c matching operations on the MOC.

Enrolment Procedure

The enrolment of a user u_i associated to a (classical) template b_i takes two steps:

1. Compute and store the encryption of the template $\mathsf{Enc}(b_i)$ into the database,

2. then, compute and store a quantized template v_i into the **SAM** memory.

Although not encrypted, the quantized templates are stored in the **SAM** memory, and are thus protected from eavesdroppers.

Access-Control Procedure

When a user u_j presents his biometrics to the sensor, he is identified in this way:

1. The processor encodes the biometric feature into the associated template b'_j.

2. The processor computes the quantized template v'_j and sends it to the **SAM**.

3. The **SAM** compares v'_j with the stored $v_1, \ldots, v_i, \ldots, v_n$. He gets a list of c candidates $v_{i_1}, \ldots, v_{i_c}$ for the identification of b'_j.

4. The **SAM** sequentially requests each of the $\mathsf{Enc}(b_i)$ for $i \in \{i_1, \ldots, i_c\}$, and decrypts the result into b_i.

5. The **SAM** completes its task by doing the c MOC comparisons, and finally validating the identity of the owner of b'_j if one of the MOC comparisons leads to a match.

Proposition VII.1. *As the biometric information of the enrolled users remains either in the* ***SAM****, or encrypted outside the* ***SAM*** *and decrypted only in the* ***SAM****, this access-control biometric terminal architecture ensures the privacy of the registered users.*

Performances of this Scheme

The main observation is that the MOC comparison is the most costly operation here as an identification executes n of them. Based on this fact, we reduce the number of comparisons, and focus on selecting the best candidates.

For an identification, we in fact switch the timing needed for n MOC comparisons within the **SAM** for the (Hamming) comparison with n quantized templates followed by a sorting for the selection of the c best candidates, and at most c MOC comparisons.

Let μ_{MOC} (resp. $\mu_{HD}(k)$; $\mathsf{Sort}(k, n)$; μ_{Dec}) be the computation time for a MOC comparison (resp. Hamming distance computation of k-bits vectors; sorting n integers of size k; template decryption). Additionally, the feature extraction and quantization of the fresh biometric image is managed outside the **SAM** by the main processor of the terminal.

Neglecting extraction and quantization, the pre-screening of candidates through quantized biometrics will improve the identification time as soon as $(n - c) \cdot (\mu_{MOC} + \mu_{Dec}) > n \cdot \mu_{HD}(k) + \mathsf{Sort}(k, n)$. Assuming that $\mu_{HD}(k)$ is 2ms for $k \leq 1000$ and that the comparison of two integers of size k takes 2ms as well, then it yields $(n - c) \cdot (\mu_{MOC} + \mu_{Dec}) > 2(n + n \cdot \log_2(n))$ms. μ_{MOC} is generally within 100ms-500ms; assume that $\mu_{MOC} + \mu_{Dec}$ takes 200ms here. Then for instance with $n = 100$ and $c = 10$, it leads to an improvement by a factor 5.6.

A Practical Example

To confirm our solution to enhance the security of an access control terminal, we run experiments through different fingerprints dataset based on a slight modification of Algorithm 3 described hereafter. Some of our results are highlighted here on the fingerprint FVC2000 second database [116].

Remark 41. Algorithm 3 takes as input a fingerprint picture I, an output length n', a mean vector μ got through training on a reference dataset, and a set $W \subset [\![1, n]\!]$ of size n'. The set W usually depends on the user $\mathcal{U}_i$, but in an identification perspective, we make W independent of the users, by selecting the components that have the highest ratio $\frac{\sigma^{(k)}_{intra}}{\sigma^{(k)}_{inter}}$ of intra-class variance with respect to inter-class variance.

With respect to Section VII.2, this procedure enables us to manage the enrolment of a set of users $u_1, \ldots, u_n$ and outputs for each user a quantized template $v_i = (V_i, VM_i)$. V_i is a k-long binary quantized vector for u_i, and

VM_i is its mask. As for the access-control procedure, when a new fingerprint image is captured for a user u_j, a minutiae-based fingerprint template b'_j is extracted together with the pattern-based template Y_j based following Algorithm 1, then the quantization handles the quantized vector $Q(Y_j)$, through Algorithm 3, to construct the vector $v'_j = (V'_j, VM'_j)$ by keeping only the indexes contained in W. We stress again that all these computations are performed by the main processor unit of the terminal.

Performances. On the second FVC2000 database, for the 100 users, with $M = 6$ images randomly selected per user for enrolment and the 2 remaining for the identification tests, we construct binary vectors v_i of length 128 ($i = 1, \dots, 100$) at enrolment and for each v'_j obtained at the access-control step, and we observe the rank of the good candidates by sorting the v_i with respect to an adapted Hamming distance between v'_j and v_i. This distance is computed as the number of differences plus half the number of positions where no value is known. In that case, 90% of the good candidates are among the 8 closest results and almost all are reached before rank 20. To reduce further the number of MOC comparisons needed, we can increase the length of the quantized templates. The experiments validate this, for 256-bit long templates: 81% of good candidates are reached on rank 2 and 90% on rank 5. The list of candidates is then almost always consolidated by very few MOC comparisons. Figure VII.2 illustrates the results with a quantization on 256 bits and 128 bits.

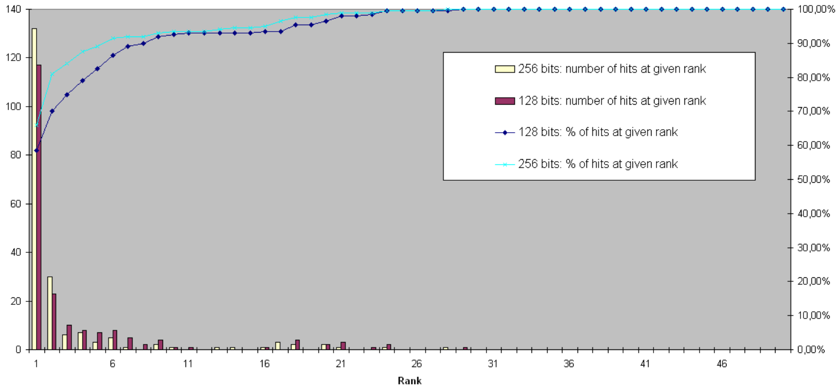


Figure VII.2: Accuracy with 128 bits and 256 bits

Remark 42. We can go further and change the scale of the setting. Indeed, the same idea can be applied at a system level. We only need to replace our Match-On-Card **SAM** by a more powerful hardware component, such as, for instance, Hardware Secure Module (HSM) [5].

However, this is interesting only if the quantization of the templates provide a scalable infrastructure, which requires to study the evolution of algorithms' accuracy for growing databases.

VII.3 Conclusion

This part presented different methods to achieve biometric identification, with cryptographic-level security. As any (modern) cryptographic construction, it went through the design of a model, security properties, and functional design. We recall here the three essential propositions that were made.

Asymmetric Cryptography Using public-key cryptography, we are able to design a system with strong privacy properties. The system is made of a server, which does not have any information on the content of the database, nor on the identity of the users that get identified. It is also possible to prevent the users from getting more information than what they ought to in this model. However this requires intensive cryptographic computation, to achieve all these properties.

Symmetric Cryptography The advances made in Symmetric Searchable Encryption enable to retrieve the closest template in a database. The privacy properties are not as protective as the asymmetric case, but the server is still unable to reconstruct a template (Adaptive Confidentiality), or to distinguish between requests (but in a non-adaptive way, *i.e.* he plays the game by the rules). The computations required for this construction are very light, and doable without extensive hardware.

Quantized- and classical- biometric combination Using a dedicated hardware architecture, identification is done with Match-on-Card technology. Because a pre-selection is made on the templates to be compared, the computation requirements are very low. The loss in terms of error rates is small, and this solution is well-suited for local identification.

This concludes the topic of applying cryptography to biometric identification. The next part deals with identification of devices, and explores the uses of identification codes.

Part 4

Identifying Wireless Devices

Chapter VIII

Wireless Communications and Privacy

> Mathematicians are like Frenchmen: whatever you say to them they translate into their own language and forthwith it is something entirely different.
>
> Johann Wolfgang von Goethe

VIII.1 Wireless Networks and Trivial Wisdom

This is now common knowledge: the transmission of a message using wireless electromagnetic waves is convenient, efficient, fast, and unfortunately, sensitive. Sending a message without protection is indeed the same as shouting in a crowded place: you make sure everyone in the neighbourhood can hear you.

In some applications of wireless communications, this is not an issue, but it is in all others. This is for example the case for military applications, where lives are at stake, but also in less critical situations, where a user's PC is connected to the Internet *via* a modem, and the connection between the PC and the modem is wireless (using for example Wi-Fi$^{\text{TM}}$). No one wants all their neighbours to hear their private conversations, nor to know the content of their e-mails, yet this is how it was done when most routers were initialized with an insecure encryption key, or no key at all.

To be more precise, here is a (non exhaustive) list of the specific threats on wireless communications.

Eavesdropping

This is the easiest way to get information: listening to what transits on the channel. We suppose that the communication protocol is known by the adversary, except for a key - as is often done in cryptography, following Kerckhoffs' principle. With this assumption, an attacker can, with very few requirements, transcript everything that transits over the air into his own memory.

This kind of threat is undetectable.

Eavesdropping is, however, useless if sufficient cryptographic protection was applied to the communicated content.

Intervening

We like to think of a channel as a wire with only two ends; this is unfortunately not the case for all of them. The classical technique of telephone tapping consists in adding a derivation to the phone line, so that an additional terminal can be inserted. This is different from eavesdropping, as someone who added a terminal to the channel can speak and try to impersonate one of the two speakers.

It is possible to find out if a phone line is tapped by physical means, but this is not the case on a wireless channel. Therefore, it is not possible *a priori* to know if a message comes from a trusted user or not, without the adequate protocol.

Jamming

It is sometimes more interesting for an attacker to simply block the communication between two parties, *e.g.* to force them to use a less secure channel. In the case of wireless communication, this is very easy to achieve, as it is possible for a malicious user to add electromagnetic noise to the wireless waves that carry information.

Such an adversarial model cannot be resolved: one can detect that someone is jamming the channel, but can do nothing to prevent it - except, obviously, finding the source of the noise.

Our Model For Electromagnetic Communication

From these observations, we deduce that the wireless channel is a public non-authenticated channel, in which an adversary can prevent the communication between two parties at a low cost. There are, however, operations that cannot be done:

Destructing: we suppose that both the emitter and receiver are tamper-proof. This strong assumption is justified, as there is no point in designing security protocols if one of the parties cannot communicate. This does not prevent an attacker from impersonating either the emitter or the receiver. In the case where a base station is communicating with several elements, an adversary can corrupt or destroy some, but not all of them.

Removing: a message is an electromagnetic wave, *i.e.* an electric and a magnetic field $\mathbf{E}(t)$ and $\mathbf{B}(t)$. *Removing* a message means clearing the field, in other words, ensuring that $\mathbf{E}(t) = \mathbf{E}_0(t)$ where $\mathbf{E}_0(t)$ is the electric field prior to the sending of a message. This can be achieved *via* two ways:

- A malicious user can stop the propagation of a wave at some point (using physical means, such as a Faraday shield); this is a physical operation equivalent to destroying one of the two communicating devices;
- He can add at some point a field $\mathbf{E}'(t)$ to $\mathbf{E}$ so that $\mathbf{E} + \mathbf{E}' = \mathbf{E}_0$. This is a very difficult operation, the easiest way to do so is by using interferences to null the signal at some point; however, as lasers are seldom used in such communication, the beams are often highly divergent.

Finally, it is hardly possible to null a signal, and thus to remove a message.

Note that this does not prevent an attacker from deliberately adding noise to a message; moreover, depending on the modulation, it is more or less easy for him to alter messages so that he can convey the message that he wants over the channel.

We shall no further consider the physical layer of message transmission, and use these hypotheses for the transmission of messages.

VIII.2 Lightweight Wireless Elements

The security problematic is more complex when the wireless elements are limited in resources. When the communicating devices are regular computing elements with virtually no memory, energy, or computation time limitations, then assuming that an element can do a cryptographic computation is standard. This is the case for laptop computers equipped with a wireless card, or for a car's embedded computer that communicates with a base station.

It is more difficult to design protocols if a device is limited in power or memory. For example, computing takes energy, and so does transmitting a message. If a communicating element does not have any other energy source than electromagnetic induction, then the amount of possible computation and communication is limited. As another example, a smartcard follows a computer-like architecture[1]. The challenge is to fill the few squared millimetres with a microprocessor, some flash memory, and specific components depending on the card use. As a consequence, designers sometimes have to do compromises in the elements in the card.

The RFID tags make an even better example of low-cost devices. Some of these elements are indeed only used for security measures: barcode-like RFIDs. These are not designed to face privacy threats. However, for some applications, it is desirable to have very low-cost wireless elements that are resistant against (non-physical) privacy attacks. For example, an electronic ID card (or passport) carries and communicates very sensitive information. This is also the case for Smartdust [98], a network of small micro-electromechanical systems equipped with wireless communications. Each speck of Smartdust is communicating, and the content of the communications need to be private enough. We demonstrate in the next chapters several techniques that can involve Smartdust.

There are many security protocols designed for RFID tags; among them, we can cite [29, 69, 131, 137, 139, 149, 154, 167, 186]. The protocol that we propose later on differs on the identification paradigm that we study, as it is the base station that first contact the devices and not the other way.

VIII.3 On Privacy

The question of privacy arises when it is possible for someone to listen to the communications, but does not understand its content[2]. From the data an eavesdropper hears, what can he deduce on the sender's and receiver's identity? Is he able to trace a device from one communication to the other?

This issue gave birth to dedicated security models, that formalize the privacy threat against a device, the different events that take place during the execution of a protocol, and the attacks of an adversary. Among these models, we can cite [58, 97, 123, 130], and suggest [81] for a more extensive list of references. We briefly recall hereafter the model for privacy, correctness and soundness described by Vaudenay in [182]. Our main concern is interrogation of devices, but it can be easily seen as an authentication protocol, so we use almost the same model.

Following [182], we consider that provers are equipped with ContactLess Device (CLD) to identify themselves. CLDs are transponders identified by a unique Serial Number (SN). During the identification phase, a random virtual serial number (vSN) is used to address them.

An identification protocol is defined as algorithms: First to setup the system made of a verifier and several CLDs, secondly to run a protocol between

[1] A Von Neumann structure.

[2] Otherwise, the question is that of the confidentiality

CLDs and verifiers. Note that we need an authority who publishes a mathematical structure.

Setup Algorithms

- $\textsc{SetupAuthority}(1^k) \mapsto (KA_s, KA_p)$ generates the system parameters defined by an authority (KA_s stands for the private parameters and KA_p for the parameters publicly available).

- $\textsc{SetupVerifier}_{KA_p}$ initializes a verifier. It may generate a private-public set of parameters (KV_s, KV_p), associated to the verifier.

- $\textsc{Setup}\mathrm{CLD}^b_{KA_p,KV_p}(\mathrm{SN})$ generates the parameters of the CLD identified by SN. This algorithm outputs a couple (s, I) where s denotes the secret (if any) parameters of the CLD, I its identity within the system. It enables to initialize the internal state of the CLD, which may be updated afterwards during an execution of the protocol. If $b = 1$, it also stores the pair (I,SN) in a database which may be made available to the verifier. If $b = 0$ it is an illegitimate device.

Communication Protocol $\mathcal{P}$ Along with these setup algorithms, the identification protocol between a CLD and a verifier consists of messages sent by the two parties. Protocol instances are hereafter denoted by π.

Oracles To formalize possible actions of an adversary, different oracles are defined to represent ways for him to interact with verifiers or CLDs, or to eavesdrop communications. The use of different oracles leads to different privacy levels.

Given a public set of parameters KV_p, the adversary has access to:

- $\textsc{Create}\mathrm{CLD}^b(\mathrm{SN})$: creates a CLD with serial number SN initialized via $\textsc{Setup}\mathrm{CLD}^b$. At this point, it is a free CLD, *i.e.* not yet in the system.

- $\textsc{Draw}\mathrm{CLD}(distr)\mapsto((\mathrm{vSN}_1,b_1),...,(\mathrm{vSN}_n,b_n))$ moves a subset of n CLDs from the set of free CLDs into the set of drawn CLDs in the system. The n CLDs are sampled from a given distribution. Virtual serial numbers vSN_i are used to refer to these CLDs. If b_i is one, this indicates whether a CLD is legitimate. This oracle creates and keeps a table of correspondences $\mathcal{T}$ where $\mathcal{T}(\mathrm{vSN})=\mathrm{SN}$. Adversary has no knowledge of this table $\mathcal{T}$.

- $\textsc{Free}(\mathrm{vSN})$: moves the drawn CLD vSN to the set of free CLDs, *i.e.* vSN cannot be used any more to query the CLD.

- $\textsc{Launch} \mapsto \pi$: makes the verifier launch a new protocol instance π.

- $\textsc{SendVerifier}(m, \pi) \mapsto m'$: sends the message m for the protocol instance π to the verifier who may respond m'.

- SendCLD$(m', \pi) \mapsto m$: sends the message m' to the CLD, which responds m.
- Result$(\pi) \mapsto x$: when π is a complete instance of $\mathcal{P}$, it returns $x = 1$ if the verifier succeeds in identifying a CLD from π and 0 otherwise.
- Corrupt(vSN)$\mapsto S$: returns the internal state S of the CLD vSN.

Types of Adversary

- **Strong** adversaries are allowed to use all of the above oracles.
- **Destructive** adversaries cannot use a corrupted CLD another time.
- **Forward** adversaries cannot use any oracle after one Corrupt query, *i.e.* destroys the system when he corrupts one CLD.
- **Weak** adversaries are not allowed to use the Corrupt oracle.
- **Narrow** adversaries are not allowed to use the Result oracle.

This defines 8 kinds of adversaries because a narrow adversary may also have restrictions on the use of the Corrupt oracle. For instance, an adversary can be narrow and forward, he is then denoted by narrow-forward.

Three security notions are defined in this model: correctness, resistance against impersonation and privacy.

Definition 14. A scheme is **correct** if the identification of a legitimate CLD fails only with negligible probability.

Resistance against Impersonation Attacks The definition of resistance against impersonation attacks (Definition 15) deals with active adversaries. Active adversaries may impersonate verifiers and CLDs, and eavesdrop and modify communications. This property of resistance against impersonation attacks has also repercussions on privacy properties (cf. Lemma VIII.1).

Definition 15. A scheme is **resistant against Impersonation Attacks** if any polynomially bounded **strong** adversary is not identified by a verifier except with a negligible probability. Adversaries are authorized to use different devices at the same time while they communicate with the verifier. Nevertheless, the resulting protocol transcript must neither be equal to the replay of a previous one between a legitimate CLD and the verifier nor lead to the identification of a corrupted CLD.

Remark 43. As a consequence, a scheme is not resistant against impersonation attacks if an adversary is able to modify on the fly outputs from a prover without affecting the identification result.

In addition to this definition, in order to mitigate replay attacks, a legitimate verifier should not output twice the same values in two complete protocol instances, except with a negligible probability.

Similarly, and as in [139], we introduce the **resistance against impersonation of verifier** where an adversary should not be able to be identified as a legitimate verifier by a non-corrupted CLD except by replaying an eavesdropped transcript. This is related to the notion of verifier authentication.

Privacy Privacy is defined as an advantage of an adversary over the system. To formalize this, [182] proposes to challenge the adversary once with the legitimate oracles and a second time with simulated oracles. In this setting, the adversary is free to define a game and an algorithm $\mathcal{A}$ to solve his game. If the two challenges results are distinguishable, *i.e.* if the system cannot be simulated, then there is a privacy leakage. A game with three phases is imposed. In the first phase, $\mathcal{A}$ has access to the whole system through oracles. In a second phase, the hidden table $\mathcal{T}$ of correspondences is transmitted to $\mathcal{A}$ (note that this table is never learned by the simulator). In a third phase, $\mathcal{A}$ is no longer allowed to use the oracles, and outputs its result.

Definition 16 (Privacy)**.** A scheme is defined as **private** if for any game, all adversaries are trivial.

The formal definition of a *trivial* adversary is provided in Definition 18.

Definition 17. A **blinded** adversary uses simulated oracles instead of the oracles Launch, SendVerifier, SendCLD and Result. Simulations are made using an algorithm called a **blinder** denoted $\mathcal{B}$.

To simulate oracles, a blinder has access neither to the provers secrets nor to the secret parameters KV_s. We denote $\mathcal{A}^{\mathcal{O}}$ the algorithm $\mathcal{A}$ when executed using legitimate oracles and $\mathcal{A}^{\mathcal{B}}$ the algorithm $\mathcal{A}$ when executed using the blinder.

Definition 18. An adversary is **trivial** if there exists a blinder $\mathcal{B}$ such that the difference $\left|\Pr\left[\mathcal{A}^{\mathcal{O}} \text{ wins}\right] - \Pr\left[\mathcal{A}^{\mathcal{B}} \text{ wins}\right]\right|$ is negligible.

Hence, to prove privacy, it is enough to prove that an adversary cannot distinguish between the outputs of the blinder $\mathcal{B}$ and outputs made by legitimate oracles. To the different kinds of adversaries enumerated above correspond accordingly as many notions of privacy. This definition of privacy is more general than anonymity and untraceability.

Note that Corrupt queries always leak information on the CLDs' identity. If an adversary systematically opens CLDs in order to track them, he is considered as a trivial one. Indeed, a blinded adversary will succeed in the same way, as the Corrupt oracle is not simulated. The aim of strong privacy is to ensure that CLDs cannot be tracked using their outputs even when their secrets are known.

The following lemma established by Vaudenay in [182] emphasizes the link between impersonation resistance and privacy:

Lemma VIII.1. *A scheme secure against impersonation attacks and narrow-weak (resp. narrow-forward) private is weak (resp. forward) private.*

The proof relies on the fact that an adversary is not able to simulate any CLD if the scheme is sound. This implies that the Result oracle is easily simulated.

[182] also proves that narrow-strong privacy implies the use of public key cryptography and that strong privacy is impossible in this model.

Organisation of this Part

The notion of identification code is central to this part. Introduced in chapter IX, the codes make possible, as a first application, to reduce the communication cost of an identification protocol. The toy example that motivates this study is the League problem, and the problematic of transmitting as little information as possible with known prior information.

Identification codes are also used in chapter X, as the keystone for an interrogating protocol. Many contactless devices are disseminated in an area, and a contactless sensor knows the identity of each of them. The protocol that we propose enables the sensor to beckon one of the devices in a private way. The security and privacy of this protocol are proved in a computational model using the hardness of the Polynomial Reconstruction Problem; the rest of the chapter is dedicated to finding an information theoretic limit on the parameters, within which the protocol remains secure.

Finally, we mention an application of coding theory to key establishment of wireless devices in Appendix D. The establishment of a secret key using public communication is a classical application of information-theory to secure communication. The adversary of this model is usually seen as passive, and we show a possible protocol to thwart active adversaries, using only lightweight devices.

Chapter IX

How not to use the Available Bandwidth

He who knows, does not speak.
He who speaks, does not know.

Lao Tzu

In 1990, Orlitsky introduced the following *League Problem* [132]. There are n football teams. Alice knows that the Pittsburgh Steelers and the Arizona Cardinals played against each other. Bob hears the name of the winning team, but not Alice. Unfortunately, he did not get the name of the loser.

This is an instance of a general problem in which Bob must send an identifier to Alice, in a smart way. Bob knows that Alice has some prior information - he even knows what kind of information that is, but does not have the information itself. Alice and Bob are then going to interact in order to identify the element efficiently.

The League Problem is to determine how many bits must Bob and Alice exchange in the worst case. If two interactions are allowed, there is a solution in $\lceil \log \log n \rceil + 1$ bits, which is optimal [132], even in the case where more than two messages are exchanged.

This chapter presents results published in [48]. This idea is to use identification codes to solve this League Problem.

Identification codes, described in section IX.2, were introduced by Ahlswede and Dueck [3] to enable Alice to know whether Bob sends a message indicating that a particular team (for instance, the Pittsburgh Steelers, Alice's favourite team) has won. These codes demand less bits than the traditional transmission codes which convey more general messages of the type "What team did Bob send?". With these identification codes, Alice can make two kinds of mistakes. First, she can believe that the team identified was not the one sent by Bob when it was. Second, she can conclude that this team was the one sent by Bob when it was not. In other words, false positives and false negatives, as in chapter II.2.

Coming back to our League Problem, we go even further. If we now allow Alice to sometimes take wrong decisions as it is the case when using identification codes, we exhibit a solution where only $\log\log\log n$ bits are required.

IX.1 The League Problem: a Specific Case of Two-Way Communications

Two-Way Communications

Orlitsky [133] explored many aspects of communication between two players. Player A knows X, player B knows Y, and we want to design a communication protocol between A and B such that, at the end of the protocol, A knows $f(X, Y)$ where f is a given function. The goal of this protocol is for A and B to send as few bits as possible.

This problem is pretty generic and obviously depends on the function f. We here suppose that n is a parameter for the length of the data X, Y (for example, Y is an element of $[\![1, n]\!]$), and we look for – asymptotic – optimal communication protocols. An upper bound on the number of bits to be sent is $\log n$, as it suffices that B sends Y to A for A to be able to compute the result.

Here, we explore the different existing possibilities for the League Problem, before exhibiting a new solution which, while allowing some errors on the result, outperforms the previously existing solutions.

Problem Statement

In a well-known league, n teams $t_1, \ldots, t_n$ are competing, until the final match where team t_α and team t_β are to play against each other. Alice knows t_α and t_β, but misses the result of the game. Bob knows who is the winner t, but not who was his opponent. How can Alice and Bob communicate so that Alice gets the result without using the channel more than is necessary?

We assume that the channel between Alice (A) and Bob (B) is two-way and noiseless, so that each sent bit is correctly received. We also assume that the ordering of the teams is known and shared between the two partners. In the following, log denotes the binary logarithm.

Practical Solutions

The trivial solution, without any interaction, is for Bob to send the name of the winning team to Alice. This takes $\lceil \log n \rceil$ bits to transmit, and is optimal in the lossless case (if Bob can transmit his message in k bits, in a lossless way, there is an injection between $\{0,1\}^k$ and $[\![1,n]\!]$; thus $2^k \geq n$). This gives upper-left cell *1W-0e* of Table IX.1.

If we allow interaction between A and B, then Orlitsky showed [132] a solution in $O(\log\log n)$ bits. First, A sends the position where the bit strings representing t_α and t_β differ - which takes $\lceil \log\log n \rceil$ bits, then B replies with the actual value of this bit – thus a $1 + \lceil \log\log n \rceil$-long solution. This solution is also shown to be optimal, and provides the upper-right cell *2W-0e*.

The problem widens if we allow some error to be made within a controlled probability. Let λ be the probability of the event "after the communication, A is mistaken about the winning team". This is referred to in [140] as "the ϵ-randomized model". The case $\lambda = 0$ leads to the previous results; we show in the following that $\lambda > 0$ leads to new interesting results.

Existing Bounds

The optimal solutions of this problem, as stated in section IX.1, satisfy some strict boundaries showed in [132]. Reusing the notations employed in that article, we note $C_m(Y|X)$ the *m-message complexity of* Y *knowing* X, *i.e.* the minimal number of bits required to transmit Y to a person who knows X, with m messages sent over the channel. Here, m is a natural number ($m \geq 1$). $C_m(Y|X)$ is a decreasing sequence in $\mathbb{N}$, whose limit is noted $C_\infty(Y|X)$.

Note that $C_m(Y|X)$ refers to the case where A knows without any doubt Y at the end of the protocol. In the case where A knows Y with probability $1-\lambda$, the corresponding quantity is noted $C_m^\lambda(Y|X)$.

With these notations, several bounds can be found in [132], among which we highlight the following two:

- (1) $C_\infty(Y|X) \geq \lceil \log C_1(Y|X) \rceil + 1$ with equality in the case of the League Problem;
- (2) $C_1^\lambda(Y|X) \leq 4C_\infty(Y|X) + 2\log\frac{1}{\lambda}$

Our work aims at improving the second bound for the League Problem; we show that allowing vanishing errors in the result enables to reduce the communication cost by a logarithmic factor. Moreover, we derive an inequality similar to the first one in the error case.

IX.2 Identification codes

Definition

Informally speaking, an identification code is a data representation that enables a receiver Bob to know, within a given error probability, if Alice sent a message $i \in [\![1,N]\!]$, or not. To be more specific, the following definition is commonly adopted.

Let $\mathcal{X}, \mathcal{Y}$ be two alphabets, and W^n a channel from $\mathcal{X}^n$ to $\mathcal{Y}^n$. W^n is defined as the probability to receive a message $y^n \in \mathcal{Y}^n$ given a transmitted message $x^n \in \mathcal{X}^n$. By extension, for a given subset $E \subset \mathcal{Y}^n$, $W^n(E|x^n)$ is the probability to receive a message belonging to E when x^n was transmitted.

Definition 19 (Identification Code, [3])**.** A $(n, N, \lambda_1, \lambda_2)$-identification code from $\mathcal{X}$ to $\mathcal{Y}$ is given by a family $\{(Q(\cdot|i), D_i)\}_{i \in [\![1,N]\!]}$ where:

- $Q(\cdot|i)$ is a probability mass function over $\mathcal{X}^n$, that encodes i,
- $D_i \subset \mathcal{Y}^n$ is the decoding set of i,
- λ_1 and λ_2 are the first-kind and second-kind error rate, with

  $$\lambda_1 \geq \sum_{x^n \in \mathcal{X}^n} Q(x^n|i) W^n(\overline{D_i}|x^n)$$

  and

  $$\lambda_2 \geq \sum_{x^n \in \mathcal{X}^n} Q(x^n|j) W^n(D_i|x^n)$$

  (where $W^n(D_i|x^n)$ is the probability to be in the decoding set D_i given a transmitted message x^n and $W^n(\overline{D_i}|x^n)$ the probability to be outside the decoding set)

for all $i, j \in [\![1, N]\!]$ such that $i \neq j$.

Given $Q(\cdot|i)$, the *encoding set* of i is defined as the set of messages x^n for which $Q(x^n|i) > 0$.

The first-kind error rate denotes the probability for a transmitted message not to be identified, and the second-kind error rate is the probability for a transmitted message to be falsely identified.

The relevant rate to consider in such a case is the *Identification Rate*, defined as $R_{ID} = \frac{1}{n} \log \log N$. This differs from the Transmission Rate of a code, which is $R_{Tr} = \frac{1}{n} \log N$. It is well known that the transmission is possible if and only if the Transmission Rate is lower than a number κ, called the (Shannon) capacity of the channel. The following theorem, which states that the Shannon capacity of a channel is also a bound for Identification codes, was shown in [3]:

Theorem IX.1 (Identification Capacity, [3])**.** *Let κ be the Shannon capacity of the channel W. Let $\epsilon > 0$.*

- *For each $0 < \lambda_1, \lambda_2 \leq 1$, there exist n, N and an $(n, N, \lambda_1, \lambda_2)$-identification code such that $\frac{1}{n} \log \log N \geq \kappa - \epsilon$;*
- *If there exists an $(n, N, \lambda_1, \lambda_2)$-identification code with $\lambda_1, \lambda_2 \leq 2^{-n\epsilon}$, then the rate of this code is such that $\frac{1}{n} \log \log N \leq \kappa$.*

This theorem basically states that for a given channel, the transmission capacity is the same as the identification capacity.

Constructing Identification Codes

There exist few constructions of identification codes. [3] uses constant-size sets as a general frame-work for identification codes. This idea was then applied by [108, 184], in constructions using constant-size codes as an instance of [3]. Another construction, based on prime numbers, is given in [4]. Finally, [124] designs an identification code based on Reed-Solomon codes, thus showing that it is possible to design such an ID-code thanks to the minimal distance of an error-correcting code. We will more particularly study this construction in chapter X.

Using Identification Codes to solve the League Problem

A first alternative way of solving the League Problem is to use Identification Codes. Instead of going through the two-round communications, B directly sends an identification tag for the winning team. As A must choose between two teams, she must check whether the received tag is identifying t_α or t_β.

To successfully achieve this goal, A and B agree beforehand on an $(m, n, \lambda_1, \lambda_2)$-identification code, where n is the number of teams and m the number of bits to be transmitted. As A knows t_α and t_β, she sets her target on D_{t_α}, then listens to B. Then B picks a message x^m according to $Q(\cdot|t)$ and sends it to A, who checks whether $x^m \in D_{t_\alpha}$.

To evaluate the error probability of such a construction, consider the following: either $t = t_\alpha$, or $t = t_\beta$. In the first case, the probability for A not to read t_α in x^m is $Q(\overline{D_{t_\alpha}}|t_\alpha)$, which is smaller than λ_1. In the second case, the probability for A to read t_α anyways is $Q(D_{t_\alpha}|t_\beta)$, which is smaller than λ_2. The overall error probability is thus $\lambda \leq \frac{\lambda_1+\lambda_2}{2}$.

Note that, according to Theorem IX.1, there exist identification codes such that m is about $\frac{1}{\kappa} \log\log n$. In our case, $\kappa = 1$. For all $\epsilon > 0$ and fixed error probability $\lambda > 0$, we therefore obtain a communication protocol for the League Problem in $\frac{\log\log n}{1-\epsilon}$ bits, a solution for the lower-left cell *1W-λe*, Table IX.1.

IX.3 Achieving a Triple-Log League Solution

We now allow two-way communications between A and B. In the errorless case, this reduced the communication complexity from $O(\log n)$ to $O(\log\log n)$. We here show that if we allow errors, we reduce the communication complexity from $O(\log\log n)$ to $O(\log\log\log n)$.

Going one Step Further

Our proposal starts with the original protocol from Orlitsky. To achieve the optimal two-way communication, [132] represents the set of all teams accord-

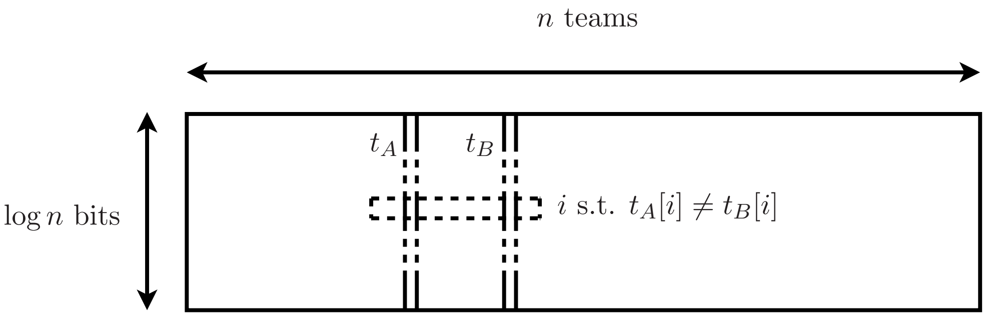


Figure IX.1: Lossless representation of n teams, and resulting two-way communication

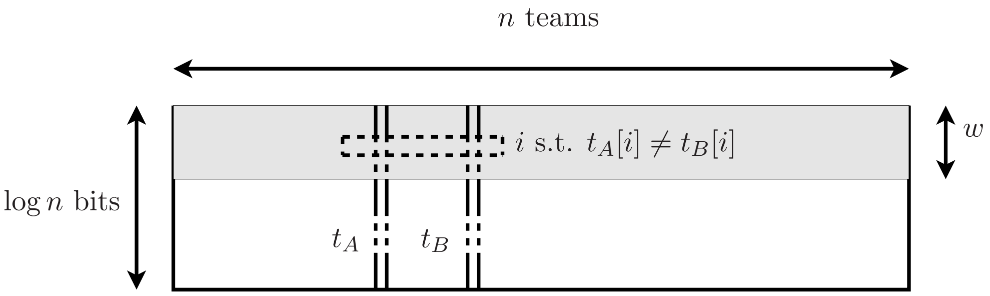


Figure IX.2: Representation of n teams on $w < \log n$ bits.

ing to an entropic coding, as is illustrated in Figure IX.1.

If we wish to achieve communication while enabling a (small) error probability, it suffices to relax the representation of Figure IX.1, and reduce the number of bits needed to represent each team from $\log n$ to w, see Figure IX.2. In doing so, A needs only send $\log w$ bits to B. Let h be an "entropic" hash function from $[\![1, n]\!]$ to $\{0, 1\}^w$; for example, $h(x)$ takes the first w bits of the representation of x in $\lceil \log n \rceil$ bits.

Indeed, as n elements are represented with a set of 2^w elements, for each w-long bit string, there will be $\frac{n}{2^w}$ elements that have the same representation. However, as we only wish to distinguish between any two elements, the probability for t_α and t_β to have the same representation is $\frac{1}{2^w}$.

From this fact, we deduce the probability for the protocol to fail, *i.e.* the probability for A not to correctly guess t between $\{t_\alpha, t_\beta\}$:

$$\Pr[\text{fail}] = \Pr[\text{fail}|h(t_\alpha) = h(t_\beta)] + \Pr[\text{fail}|h(t_\alpha) \neq h(t_\beta)]$$

As the protocol is always successful when t_α and t_β have different representations, we find the probability of error to be $\Pr[\text{fail}] = \frac{1}{2^{w+1}}$.

Triple-log Solution with Vanishing Error Probability

In the specific case where $w = \lceil \log\log n \rceil$, the protocol takes an overall length of $\lceil \log\log\log n \rceil + 1$ bits of communications with $\Pr[\text{fail}] = \frac{1}{2\log n}$. This result is worth noting in Table IX.1, lower-right cell *2W-λe*.

This error probability might seem non-negligible if stated under those terms - as $\frac{1}{\log x}$ slowly converges to 0. However, we emphasize the fact that this enables to solve the League Problem with a huge number of competing teams with very small communication between A and B.

Another way of stating this is that by sending $m+1$ bits over a channel, it is possible to identify $2^{2^{2^m}}$ teams, with an error probability of only $\frac{1}{2^{2^m+1}}$, which is actually negligible.

This result also beats the bound of Inequality (2):

$$C_1^\lambda(Y|X) \leq 4C_\infty(Y|X) + 2\log\frac{1}{\lambda}.$$

In this case, the upper bound is equal to $6\lceil \log\log n \rceil + 4$, which is still greater than $\lceil \log\log\log n \rceil + 1$.

Revisiting the Two-Way Communication Paradigm

Using the notations introduced in section IX.1, we here show that the triple-log result for C_∞^λ obtained is coherent with the double-log communication result for C_1^λ. This is shown in the following theorem:

Theorem IX.2. *For all (X, Y) pairs, for all $0 < \lambda < 1$, the following inequality holds:*

$$C_2^\lambda(Y|X) \geq \left\lceil \log C_1^\lambda(Y|X) \right\rceil$$

Proof It is similar to that of Inequality (1); we formalize a two-way protocol in this fashion:

- A sends a – possibly randomized – message $\sigma_A = \mathsf{createMessage}(X)$,
- B receives σ_A and replies with $\sigma_B = \mathsf{reply}(Y, \sigma_A)$, which can also be randomized;
- A receives σ_B and deduces $Y' = \mathsf{deduce}(X, \sigma_A, \sigma_B)$ such that $\Pr[Y' = Y] \geq 1 - \lambda$.

Assume that messages σ_A have (maximal) length l_A and messages σ_B have (maximal) length l_B. In this case, $f_B = \mathsf{reply}(Y, \cdot)$ is a function from $\{0,1\}^{l_A}$ to $\{0,1\}^{l_B}$, which enables to determine Y with probability $1 - \lambda$.

The graph of f_B is the set of all (σ_A, σ_B) such that $f_B(\sigma_A) = \sigma_B$, thus a subset of $\{0,1\}^{l_A} \times \{0,1\}^{l_B}$, and can be represented as a subset of $\{0,1\}^{l_A+l_B}$, *i.e.* an element of $\{0,1\}^{2^{l_A+l_B}}$.

In order to transform a two-ways protocol into a one-way protocol, it suffices for B to fully send his function reply, which he can do in $2^{l_A+l_B}$ bits.

Then, A can compute σ_A using createMessage, apply it to f_B, and deduce Y' such that $\Pr[Y'=Y] \geq 1-\lambda$.

This shows that if there exists a protocol with 2-message complexity C, then there exists a protocol with 1-message complexity 2^C; thus $C_2^\lambda(Y|X) \geq \lceil \log C_1^\lambda(Y|X) \rceil$. □

Remark 44. For the sake of clarity we dealt with protocols that only use 2 messages, but in fact our theorem can easily be extended to any number of messages greater than 2, using a function f_B that depends not only on σ_A, but on all previously sent messages. This shows that $C_\infty^\lambda(Y|X) \geq \lceil \log C_1^\lambda(Y|X) \rceil$.

We already showed that it is possible, using identification codes, to solve the League Problem with errors with a one-way communication cost of

$$\lceil \log\log(n) \rceil + 1.$$

Applying Theorem IX.2 shows that our result, namely a two-ways protocol for the League Problem with communication cost $\lceil \log\log\log n \rceil + 1$, is coherent.

Double-log One-Way Solution with Vanishing Error Probability Without Identification Codes

The result of the previous section incites us to apply the proof of Theorem IX.2 in order to find an efficient solution for the League Problem, in only one message.

Actually, instead of sending the graph of the function f_B as previously defined, it suffices to send, in an equivalent way, the first $w = \lceil \log\log n \rceil$ bits of the winning team t.

As the receiver A has only the choice between two teams, she fails exactly when both teams have the same $\lceil \log\log n \rceil$ first bits. This happens with probability $2^{-\lceil \log\log n \rceil} \approx \frac{1}{\log n}$.

This shows that the League Problem has a trivial one-way solution in $\lceil \log\log n \rceil$ with vanishing error-probability.

	One-Way Communication (1W)	Two-Way Communication (2W)
Errorless (0e)	$\lceil \log n \rceil$ Entropic coding *optimal*	$1+\lceil \log\log n \rceil$ [132] *optimal*
λ Errors (λe)	$\frac{\log\log n}{1-\epsilon}$ Identification Codes $\lceil \log\log n \rceil$ Section IX.3	$1+\lceil \log\log\log n \rceil$ Section IX.3

Table IX.1: Summary for the different cases. λ is the probability for Alice not to have the correct result.

Remark 45. For a given error-probability λ, an interesting question is to determine the minimal solution to the League Problem such that Alice gets the result with probability $1 - \lambda$. Our work provides an upper bound on the solution for one- and two-way communications; the minimal number of bits to be emitted is still an open problem.

IX.4 Unlocking Possible Extensions with Coding Theory

Communicating over a Noisy Channel

The problem of communicating over a noisy channel was introduced by Shannon and is well-known. Given two alphabets $\mathcal{X}$ and $\mathcal{Y}$, a channel from $\mathcal{X}^m$ to $\mathcal{Y}^m$ is a mass function $W : \mathcal{X}^m \times \mathcal{Y}^m \to [0, 1]$ which defines the output of a message x^m. The channel is noisy if W cannot be represented as the identity function.

Transmitting information over such a channel is always possible at a given rate if this rate is lower than the capacity of the channel $\kappa(W)$. This means in practice that in order to transmit k bits of information, one must send at least $m = k/R$ bits where $R < \kappa$.

Finding the optimal data structure to communicate over a noisy channel is an open problem, way beyond the scope of this chapter. In the following, we shall assume that the channel noise is overcome by classical coding techniques, and thus focus only on the problem of the information to transmit.

A League Problem with More than 2 Competing Teams

Consider a generalization of the initial problem, where Alice misses the result of the game between t_α and t_β, to the following: In the universe of the n teams competing, the final round involved $s + 1 \geq 2$ teams. How can now A get from B the identity of the winner? A trivial solution is to call $\binom{s+1}{2}$ times the initial ($s = 1$) protocol. One can however get a linear (in s) solution by making use of *separating* codes [52], defined as follows:

Definition 20. Let Q be an alphabet of size q, s, u integers. A subset $C \subset Q^m$ is (s, u)*-separating* if for any two disjoint subsets S, U of C with $|S| = s$, $|U| = u$, there is some coordinate $i \in [\![1, m]\!]$ such that for any $x \in S$ and any $y \in U$, we have $x_i \neq y_i$.

We only need here a specialization to the case $q = 2, u = 1$.

There exist asymptotic families of $(s, 1)$-separating codes with rate $R_s > 0$. An existential proof is easy to come up with; for constructions, one can resort to algebraic geometry codes on large alphabets, *e.g.* [189], and then concatenate to get binary codes. We do not elaborate on this topic here, since we only need to achieve a non zero rate R_s for our purpose.

The idea is the following: encode $n = 2^{R_s m}$ binary sequences (teams) on m bits using such a code: then, for any ordered $(s+1)$-subset of teams $(t_{i_1}, ...t_{i_{s+1}})$, there exists an index $j \in [\![1, m]\!]$ such that the j-th bit of t_{i_1} is 0 and all others t_i's have a 1, or the opposite. When A asks B for this bit, she identifies t_{i_1}; calling this protocol at most $s+1$ times is enough.

Real-Life Application

Imagine that a cloud of n Smartdust is released over a geographical zone. Some sensors are installed in this zone. During a kind of system setup, the sensors collect the identities of the different specks of Smartdust in their area of listening. We assume that each sensor possesses at most $s+1 < n$ specks in their area of listening. Using section IX.4, we encode each identifier t_i on $m = \lceil \frac{1}{R_s} \log n \rceil$ bits.

After that, the sensors periodically want to verify if a given speck of Smartdust is still working. We can imagine that sensors have to reduce the length of communications to a minimum; for instance to save needed energy of transmission.

To test the liveness of an element noted $t_e \in \{t_{i_1}, \ldots, t_{i_{s+1}}\}$, using section IX.4, sensors compute the index $j \in [\![1, m]\!]$ such that $t_e[j]$ is different from the other $t[j]$ for $t \in \{t_{i_1}, \ldots, t_{i_{s+1}}\}$. They then broadcast a message of type: $(j, t_e[j])$ where j is encoded over $\log m$ bits. Note that the total size of the message is $1 + \lceil \log \frac{\log n}{R_s} \rceil$. Each node of Smartdust which receives the message checks whether it is the one that has to answer to the sensor. In this case, it emits an acknowledgement sequence.

Conclusion

In this chapter, we show that allowing vanishing errors into the determination of the results can save an extra log factor in the communication cost of the League Problem. More generally, denoting by $\log^{(i)} n$ the i-th iterated logarithm, a straightforward extension of the results in section IX.3 yields that, if we code in length $w = \lceil \log^{(i)} n \rceil$, then the overall protocol length will be in $\lceil \log^{(i+1)} n \rceil$ and the error-probability less than $1/\log^{(i-1)} n$.

Chapter X

A Private Interrogation of Devices Protocol

> It ain't what you don't know that gets you into trouble. It's what you know for sure that just ain't so.
>
> Mark Twain

In the field of contactless communication, a verifier (often called a sensor or reader of devices) is used to identify the objects by verifying the validity of the attached contactless devices. This is the case for Radio Frequency IDentification (RFID) systems, where devices are attached to physical objects. The verification is realized through an authentication protocol between a device and the verifier. Once authenticated, the verifier manages the object and allows the owner of the object to access some service. Applications examples include in stock management application for real-time item identification and inventory tracking, e-passport applications, etc. Devices can also be part of a sensor network that gives information on the related infrastructure around a geographical zone.

In this context, a verifier has often to manage many devices at the same time in the same area. Main issues are then efficiency, security and cost, and, of course, the very specific issue to the field of contactless communication: privacy. This issue, discussed in chapter VIII, is an active research field, as the community providi models and solutions.

Contactless devices are generally assumed to respond automatically to any verifier scan. In this work, we follow an idea [142] that suggests that the verifier directly addresses the device with which it wants to communicate. To this aim, the verifier broadcasts the device identifier and then the corresponding device responds accordingly. However, the emission of the device identifier enables an eavesdropper to track it. We here look for a solution that does not require many computations and many communications efforts, while preventing an eavesdropper to be able to track a particular device. Changing the paradigm from the situation where a device initiates the protocol to a situation where the device identifies first the interrogation request enables to envisage new solutions.

We show that Identification Codes [3] perfectly fit our needs. Such a probabilistic coding scheme increases a lot the job of the eavesdropper as the same identifying bit string is not used twice except with a small probability. In particular, for the class of identification codes of [124], a reduction to the cryptographic assumption of [105] is possible.

We first describe a general scheme based on these identification codes and show that our scheme satisfies good security and privacy properties by analysing it in the privacy model defined in [182]. We then explain how the scheme is suited to very low-cost devices.

Note that the problematic of this chapter is not limited to interrogation of low-cost devices; in fact, we focus on interrogation protocols and any independent component that communicates over a noisy broadcasting channel is a potential target.

X.1 Identification Codes

We wish to communicate mainly with contactless devices, which means that all the communications are to pass through radio waves. As a direct consequence, a message sent over the channel is publicly available to any eavesdropper. In a realistic model where a verifier sequentially communicates with wireless devices, it is the verifier that will initiate the communication. To that purpose, the verifier first beckons the device with which it wants to communicate. The most efficient way of doing so is by using an identification code.

Informally, a $(\eta, N, \lambda_1, \lambda_2)$-identification code is given by a set of (probabilistic) coding functions from $[\![1, N]\!]$ to $\mathcal{X}^\eta$, along with (deterministic) decoding sets. The error rate λ_1 gives the probability of a false-negative, and λ_2, of a false-positive identification. The formal definition was provided in section IX.2.

We stress that the use of an identification code in our case is more interesting than using a transmission code for the following reasons:

- The efficiency in terms of information rate: the rate of such a code is defined as $R = \frac{1}{\eta} \log\log N$ and can (see Theorem IX.1) be made arbitrary close to the (Shannon) capacity of the channel. This means that it is possible to identify $N = 2^{2^{R\eta}}$ devices with a message of length η, with constant error rates (λ_1, λ_2). A regular **transmission** code permits only to identify $2^{R\eta}$ devices.

- The transmission of an element of D_i to identify the device i permits its identification without completely giving away the identity i. Indeed, an eavesdropper only gets the message sent $x^\eta \in Y^\eta$, not the associated index i. The use of an identification code is thus a good way to enhance privacy in the beckoning of wireless devices. This notion is formalized in Section X.2.

The proof of Theorem IX.1 is based on a generic construction, exhibited hereafter. Let $A_1, \dots, A_N \subset X^\eta$ be N subsets such that each A_i has cardinal n and each intersection $A_i \cap A_j$ for $i \neq j$ contains at most λn elements. The encoding distribution $Q(\cdot|i)$ is defined as the uniform distribution over A_i; in the noiseless case (the channel W^η is the identity function) the decoding sets are also the A_i's. Note that in that case the false-negative rate λ_1 is equal to 0 and the false-positive rate λ_2 is λ.

This theoretical construction gives way to multiple practical identification codes based on constant-weight codes, such as [65, 108, 184]. We focus on [124] which provides a simple though efficient identification code well suited to our application.

Moulin and Koetter Identification Codes Family

We here recall a simple construction of identification codes proposed by Moulin and Koetter [124].

The identification code detailed in [124] is based on an Error-Correcting Code C of length n, size $N = |C|$ and minimum distance d over some alphabet. For a word $c_i = (c_i^{(1)}, \dots c_i^{(n)}) \in C$, the corresponding set A_i is the collection of all $(u, c_i^{(u)})$, for $u \in [\![1, n]\!]$. Note that we indeed have sets A_i of constant size n; moreover, the intersection of two different sets $A_i \cap A_j$ contains at most $n - d$ elements, which induces $\lambda_2 = \frac{n-d}{n} = 1 - \frac{d}{n}$.

A Reed-Solomon code over a finite field $A = \mathbb{F}_q$, of length $n < q - 1$, and dimension k, is the set of the evaluations of all polynomials $P \in \mathbb{F}_q[X]$ of degree less than $k - 1$, over a subset $F \subset \mathbb{F}_q$ of size n ($F = \{\alpha_1, \dots, \alpha_n\}$). In other words, for each k-tuple $(x_0, \dots, x_{k-1}) \in \mathbb{F}_q^k$, the corresponding Reed-Solomon word is the n-tuple $(y_1, \dots, y_n)$ where $y_i = \sum_{j=0}^{k-1} x_j \alpha_i^j$. In the sequel, we identify a source word $(x_0, \dots, x_{k-1}) \in \mathbb{F}_q^k$ with the corresponding polynomial $P = \sum_{j=0}^{k-1} x_j X^j \in \mathbb{F}_q[X]$.

Definition 21 (Moulin-Koetter RS-Identification Codes)**.** Let $\mathbb{F}_q$ be a finite field of size q, $k \leq n \leq q-1$ and an evaluation domain $F = \{\alpha_1, \dots, \alpha_n\} \in \mathbb{F}_q$. Set $A_P = \{(j, P(\alpha_j)) |\, j \in [\![1, n]\!]\}$ for P any polynomial on $\mathbb{F}_q$ of degree at most $k - 1$.

The Moulin-Koetter RS-Identification Codes is defined by the family of encoding and decoding sets $\{(A_P, A_P)\}_{P \in \mathbb{F}_q[X], \deg P < k}$. This leads to a $(\log_2 n + \log_2 q, q^k, 0, \frac{k-1}{n})$-identification code from $\{0, 1\}$ to $\{0, 1\}$.

Application to our Setting

Back to our original problem of devices interrogation, here comes a brief description of a set-up that enables the use of identification codes to initiate a protocol between a verifier and a device. A more formal description is given in Section X.3.

A set of $M < q^k$ devices is constructed, and each of them is associated with a different random polynomial $p_l \in \mathbb{F}_q[X]$ of degree less than $k-1$. The memory of these devices is then filled with a set of $p_l(\alpha_j)$, for $\alpha_j \in F$, with F a public subset of $\mathbb{F}_q$, *i.e.* the devices contain the evaluation of p_l over a subset of $\mathbb{F}_q$. The verifier is given the polynomial p_l.

When the verifier wants to initiate communication with the device number l associated with the identifier p_l, it selects a random $\alpha_j \in F$ and sends $(j, p_l(\alpha_j))$ over the wireless channel. A device that receives this message checks whether the value stored in its memory at the corresponding address is equal to $p_l(\alpha_j)$, *i.e.* computes an equality test of two bit strings. If the test is successful, it replies and goes through the authentication protocol described in Section X.3. Otherwise, it remains silent.

Consequently, only a legitimate verifier can interrogate a specific device. Next sections emphasize the security properties reached thanks to this principle.

X.2 Vaudenay's Model for Privacy

The model for privacy, soundness and correctness in which we consider our solution, is described in [182], and was presented in section VIII.3. Our main concern is interrogation of devices, but it can be easily seen as an authentication protocol, so the same model - up to a single modification - still applies.

This model defines eight kinds of adversaries: 'strong', 'destructive', 'forward' and 'weak', as any such adversary can also be 'narrow'.

Remark 46. The notion of **destructive** adversary is an intermediate notion between **strong** and **forward** adversaries. As explained in [130], **destructive** notion is different from **forward** notion only when the system enables the introduction of some correlated secrets between CLDs. This is not our case in the sequel, so we will no further distinguish these two notions.

The definition of privacy in this model is based on the indistinguishability of the distribution of the output of the oracles an adversary can access (see Definition 16). The list of these oracles is given in chapter VIII. Note that following Remark 46, the Corrupt oracle will be useless for impersonation attacks against our scheme (as secret are not correlated between devices).

Similarly, and as in [139], we introduce the **resistance against impersonation of verifier** where an adversary should not be able to be identified as a legitimate verifier by a non-corrupted CLD except by replaying an eavesdropped transcript. This is related to the notion of verifier authentication. Note that we introduce a slight restriction in Section X.4 as our scheme aims only at ensuring validity of the verifier against a pre-fixed CLD.

X.3 Our Protocol for Interrogation

Our aim is for a CLD to recognize itself into a verifier request, but authentication of the CLD toward the verifier is handled as well. That is how we set-up the system:

- $\textsc{SetupAuthority}(1^\ell)$ generates a set of parameters KA_p defining two integers η, N, two alphabets $\mathcal{X}$, $\mathcal{Y}$, and two error rates λ_1, λ_2. No private parameter is defined.
- $\textsc{SetupVerifier}_{KA_p}$ constructs $\mathcal{IC} = \{(Q(\cdot|i), \mathcal{D}_i)\}_{i \in [\![1,N]\!]}$ an $(\eta, N, \lambda_1, \lambda_2)$-identification code from $\mathcal{X}$ to $\mathcal{Y}$ following Definition 19, and sets $KV_p = \mathcal{IC}$. $\mathcal{IC}$ is based on the Moulin-Koetter construction [124] (cf. Definition 21).
- $\textsc{SetupCLD}_{KV_p}(\mathrm{SN})$ first returns randomly chosen $(i, j) \in [\![1, N]\!]$, $i \neq j$ as the parameters of the CLD identified by SN. It then initializes the CLD with the storage of a description of the decoding set D_i of the identifier i and the description of $Q(\cdot|j)$, the encoding probability mass function for index j. It also stores $(i,\ j,\ \mathrm{SN})$ in the verifier database.

A verifier and a set of devices are set-up as above and the following steps are then processed to interrogate and authenticate a specific CLD.

- The verifier, who wants to interrogate the CLD of identifier SN, recovers its identifier i in the database and encodes it via $Q(\cdot|i)$ into a message $x \in \mathcal{X}^\eta$. The verifier broadcasts the message (ACK, x), where ACK is an acknowledgement number which will help the verifier to sort the received answers when it emits simultaneously several such messages.
- Any listening CLD that receives the message (ACK, y) uses its own decoding set $D_{i_{CLD}}$ to determine whether y encodes i_{CLD}.
- If a CLD identifies y as an encoding of its identifier i_{CLD}, then it sends the message (ACK, x') to the verifier, where ACK is the incoming acknowledgement number and x' is an encoding of j_{CLD} obtained via $Q(\cdot|j_{CLD})$.
- Upon receiving this message, the verifier then checks whether the received message y' is a member of the decoding set D_j of the aimed CLD. If so, then the CLD is declared as authenticated.

CLD identifiers p, p'	parameters $\mathbb{F}_q, (\alpha_1, \ldots, \alpha_n)$	Verifier (l, p_l, p'_l)
	$\xleftarrow{(ACK, j, a=p_l(\alpha_j))}$	Pick j
If $p(\alpha_j) = a$	$\xrightarrow{(ACK, b=p'(\alpha_j))}$	Check whether $p'_l(\alpha_j) = b$

Figure X.1: CLD identification via Moulin-Koetter identification codes

Note that here x' has to be chosen in relation with the value of y so that impersonation of a CLD is not easy.

Specifications using Reed-Solomon based Identification Codes

We now consider only the Moulin-Koetter setting, in particular for the security analysis in the next sections. The description is given below (see also Fig. X.1).

In this setting, a set of CLDs is constructed where each of them – say CLD_l – is associated with two different random polynomial identifiers p_l, $p'_l \in \mathbb{F}_q[X]$ of degree at most $k-1$. Here p_l and p'_l are good descriptions of the associated encoding functions and the decoding sets; they are both stored on the CLD side and on the verifier database.

When the verifier wants to initiate communication with CLD_l (with identifiers p_l, p'_l), it selects a random $\alpha_j \in F \subset \mathbb{F}_q[X]$ and broadcasts $(ACK, j, p_l(\alpha_j))$ over the wireless channel. A CLD with identifiers p, p' that receives this message checks whether the polynomial p stored in its memory evaluated in α_j is equal to $p_l(\alpha_j)$. If the test is successful, it responds with the value $(ACK, p'(\alpha_j))$. Otherwise, it remains silent. The verifier authenticates the CLD if the received value $p'(\alpha_j)$ is equal to $p'_l(\alpha_j)$.

Remark 47. As a practical assumption, our interrogation protocol works as a broadcast channel and we assume that a legitimate verifier is interrogating several CLDs during the same period. Although it might look restrictive, recall that our goal is to address applications where a verifier has to manage efficiently a cloud of CLDs. Our protocol does not aim to be private if one device is isolated. More formally, we assume that a cloud of M CLDs is present in the broadcast area of the verifier and that the verifier interrogates them uniformly in a random order. In particular, an adversary is not able to *a priori* distinguish the devices without trying to exploit the content of messages exchanged.

For privacy purposes, we do not want replay attacks to be possible at all. In order to avoid them, we add to each device a flag bit that tells if the α_j was already used or not; this bit is flipped on at the reception of $(j, p(\alpha_j))$; after that, a device no longer accepts such a message. This can be seen as coupons enabling a limited number of interrogations by a legitimate verifier.

X.4 Security Analysis

Remark first that the scheme is correct: In the Moulin-Koetter construction (cf. Section X.1) the false-negative error rate (λ_1) is zero, thus the correct CLD will always answer and be authenticated.

Assumptions

The security results of this scheme are linked to solving the problem of polynomial reconstruction (PR) [102, 103, 104, 105]:

Definition 22 ([105])**.** Given n, k, t such that $n \geq t \geq 1$, $n \geq k$ and $z, y \in \mathbb{F}_q^n$, with $z_i \neq z_j$ for $i \neq j$, output all (p, I) where $p \in \mathbb{F}_q[X]$, $\deg(P) < k$, $I \subset [\![1, n]\!]$, $|I| \geq t$, and $\forall i \in I, p(z_i) = y_i$. Such an instance of this problem is noted $PR^z_{n,k,t}$.

The Guruswami-Sudan algorithm [82] for the **list decoding** of Reed-Solomon codes gives a way to solve the polynomial reconstruction problem when $t \geq \sqrt{kn}$. However, no efficient solution to this problem exists when $t < \sqrt{kn}$ and it is reputed hard. If $t < k$, PR is unconditionally secure (in the information-theoretical meaning).

Based on the assumed intractability of PR, [105] derives the Decisional PR (DPR) problem which consists, given an instance y of $PR^z_{n,k,t}$ for which there exists a solution (p, I), in determining whether a given $i \in [\![1, n]\!]$ is in I. Thanks to the DPR assumption (hardness of the DPR problem), it is shown [105] that PR instances are pseudo-random and that they do not leak any partial information on the polynomial values.

Remark 48. In the sequel we assume that the PR and DPR problems remain hard (with respect to the security parameter ℓ) even in our setting – where the noise is generated by the other queries and responses. M will be chosen so that the DPR assumption holds when the noise is assumed to be random. To justify this choice, we refer to [83] which explains the link between Reed-Solomon list decoding and the previous works on polynomial reconstruction in the mixture model. An algorithm to reconstruct polynomials from mixed values is designed in [8]. When considering mixed evaluations of M polynomials of degree at most $k - 1$, it enables to reconstruct one of these polynomials when at least $M(k - 1)$ related values are available in the mixture. In the sequel, we set M greater than $\sqrt{\frac{n}{k}}$ so that $M(k - 1)$ is approximately greater than $\sqrt{nk}$, i.e. that we obtain the same bound as for the solvability of PR instances. This algorithm is the basis – although a bit simpler – of the list decoding algorithm

[82] and this fact suggests that when we get less than $M(k-1)$ values for each polynomial with M large, the problem of reconstructing one polynomial remains hard even without a perfectly random noise.

Effect of Passive Eavesdropping

When listening on the channel to the queries made by a legitimate verifier and the replies produced by legitimate CLDs, an eavesdropper sees messages of this kind:

$$\Big(ACK_i, j_i, p_{l_{j_i}}(\alpha_{j_i})\Big), \Big(ACK_i, p'_{l'_{j_i}}(\alpha_{j_i})\Big)$$

(for l'_j such that $p_{l'_j}(\alpha_j) = p_{l_j}(\alpha_j)$), for some number of i's (say $i \in [\![1, T]\!]$). Note that we may also have collisions on the α_j used (i.e. $j_i = j_{i'}$ may occur for some $i \neq i'$). This means that the adversary obtains a set S of several PR instances of length less or equal to n (the length of the overall code, see Section X.1). Targeting a specific CLD, of identifier p and p', then there are at least two corresponding PR instances, $PR^{z_1}_{n_1,k,t_1}$ and $PR^{z_2}_{n_2,k,t_2}$ where p is one solution of the first one and p' a solution of the latter, among the set S of all those PR instances. One difficulty for the adversary is to sort the different messages and to deal with the collisions to extract such instances. If we assume that there is no collision (then necessarily $T \leq n$) and that the verifier queries uniformly the M CLDs (cf. Remark 47), then it implies that the adversary can recover these instances, but with $t_i \approx \frac{n_i}{M}$. So if M is greater than $\sqrt{\frac{n}{k}}$ then the PR instances are hard.

Moreover, when the number of received messages is large, the t_i's above may be greater than $\sqrt{kn}$ but the adversary has to deal with the collisions and to try all the different instances until the recovery of a solvable instance. Another strategy is to see the problem as one longer PR instance. This is related to the **list recovery problem** which is analysed in [152]. This is hard as well given some restriction on the number of eavesdropped messages.

Proposition X.1. *Assume that the number M of devices simultaneously queried by the verifier is such that $\sqrt{q} \geq M \geq e\sqrt{\frac{n}{k}}$ (with $e = exp(1)$). Then a passive adversary, who eavesdrops at most T requests with $T < M^2k$, cannot reconstruct the polynomial identifiers, except with a negligible probability.*

> **Proof** Assume that the adversary has eavesdropped T different requests with $T/M \geq \sqrt{kn}$, then there may exist solvable PR instances. Now he has to find these solvable instances among all possible instances. Following Remark 47 on uniformity of the queries made by a verifier, we assume that the number of different requests to each device is exactly $t = T/M$. (Due to the false-positive error rate of the underlying identification code, one request will address several additional devices and imply as many replies. In fact, as the polynomials are chosen independently and uniformly, the number of devices addressed by one query is strictly greater

than 1 only if there is a collision during the evaluation of several polynomials. The assumption $M \leq \sqrt{q}$ enables us to neglect this point, but the result is easily generalizable to the case $M > \sqrt{q}$.)

Let $M \geq \gamma\sqrt{\frac{n}{k}}$ where γ will be determined later. Note that if $T/M < k$ then it is unconditionally secure and if $T < \gamma n$ then $T/M < \sqrt{nk}$ so that the PR instances are hard. Assume that $T \geq \gamma n$, thus the number of collisions per α_j is expected to be about T/n (note that $T/M \leq n$ as each device is linked to at most n different requests). To make computation more tractable, we assume below that the number of collisions per α_j is exactly T/n.

A first strategy for the adversary is to find a solvable PR instance in the classical meaning, i.e. without any collision. The number of possible PR instances is then expected to be $B = \left(\frac{T}{n}\right)^n$ whereas the number of solvable instances is $A = M \times \binom{T/M}{\lceil\sqrt{kn}\rceil}\left(\frac{T}{n}\right)^{n-\lceil\sqrt{kn}\rceil}$. If the ratio $\rho = \frac{A}{B}$ of the number of solvable instances over the number of all possible instances is negligible then the adversary would not find a solvable instance in polynomial time. In fact ρ is equal to

$$M\binom{T/M}{\lceil\sqrt{kn}\rceil}\left(\frac{T}{n}\right)^{-\lceil\sqrt{nk}\rceil}.$$

To approximate ρ, note $R = \frac{k}{n}$ the rate of the Reed-Solomon code as eavesdropped by the adversary. We also introduce $\theta > 1$ such as $\frac{T}{M} = \theta\sqrt{kn}$. The notations give $M = \frac{\gamma}{\sqrt{R}}$ and $\frac{T}{n} = \theta\gamma$. A good approximation of $\binom{T/M}{\lceil\sqrt{kn}\rceil}$ is, for $\theta > 2$, $2^{\frac{T}{M}h_2\left(\frac{M\sqrt{kn}}{T}\right)} = 2^{n\sqrt{R}\theta h_2(\frac{1}{\theta})}$ where h_2 is the binary entropy function. This shows that ρ can be fairly approximated by

$$\rho \approx \frac{\gamma}{\sqrt{R}}2^{n\sqrt{R}\left(\theta h_2(\frac{1}{\theta})-\log_2(\theta\gamma)\right)}.$$

Taking a closer look at the exponent, we see that $\theta h_2(\frac{1}{\theta}) - \log_2(\theta\gamma) = (\theta - 1)\log_2(\frac{\theta}{\theta-1}) - \log_2(\gamma)$ is negative only if $\gamma > \left(1 + \frac{1}{\theta-1}\right)^{\theta-1}$. As $\forall x \in \mathbb{R}^\star, \log(1 + \frac{1}{x}) < \frac{1}{x}$, we deduce that if $\gamma \geq e$, then $\theta h_2(\frac{1}{\theta}) - \log_2(\theta\gamma) < 0$. Thus, $\rho \leq M2^{-n\sqrt{R}\log_2(\frac{\gamma}{e})}$ is negligible.

This gives a negligible probability for the adversary to find a solvable instance. This conclusion can be generalized to non-constant number of collisions as soon as the j picked by the verifier is chosen uniformly and independently among the different requests.

A second strategy is to apply the list recovery technique [152] derived from the list decoding algorithm [82]. This becomes tractable as soon as T/M is greater than $\sqrt{nk \times l}$ with l the maximum number of collisions per α_j (roughly, this corresponds to solving a PR instance of length nl).

Here $l = T/n$ and the condition $T/M \geq \sqrt{nkl} = \sqrt{Tk}$ is equivalent to the condition $T \geq M^2k$. Due to our hypothesis on the number of eavesdropped messages, the algorithm cannot be applied. □

Remark 49. In practice, the cloud of devices is dynamic, some devices may exit or enter the cloud around a verifier, so that the difficulty for the attacker can only increase.

Following this proposition and via the DPR problem, then a passive adversary cannot distinguish the answers as soon as the same interrogation request does not appear twice.

Proposition X.2. *Assume* $\sqrt{q} \geq M \geq e\sqrt{\frac{n}{k}}$ *and* $T < M^2k$. *A passive adversary cannot determine whether two requests correspond to the same CLD except if there is a collision, that happens only with probability* $1/\sqrt{n}$.

Security Against Impersonation

In our protocol, a CLD replies to the verifier only if it believes that the verifier is legitimate. It is thus close to mutual authentication – although here the authentication of the verifier is only probabilistic with respect to the false-positive error rate of an identification code. It is a weaker result than general verifier authentication: a verifier cannot be impersonated in order to interrogate a pre-fixed CLD.

Proposition X.3. *Assume* $\sqrt{q} \geq M \geq e\sqrt{\frac{n}{k}}$ *and* $T < M^2k$. *In our scheme, given a non-corrupted CLD, an adversary cannot impersonate a verifier to interrogate this specific CLD, without replaying an eavesdropped transcript, except with probability* $\frac{1}{q}$.

Proof To interrogate a CLD, the only useful information for an adversary are the requests made by the verifier. Proposition X.1 implies that this does not give an efficient solution to the adversary for obtaining information on one identifier.

Hence, the remaining solution to interrogate a CLD is to try at random to initiate a communication without prior knowledge of its identifier. The question is what is the probability to succeed out of a random couple (j, a)? If a specific CLD with identifier p is targeted, this probability is equal to $\Pr[p(\alpha_j) = a] = \frac{1}{q}$. □

Of course, if no specific CLD is fixed, then impersonation of an interrogation towards a member of a large set of CLDs is easier. With M CLDs, the probability to reach one of them correctly is $\frac{M}{q}$.

Given this difficulty of impersonating a verifier against a chosen CLD and the uselessness of eavesdropping (cf. Proposition X.1), we deduce the resistance of CLDs against impersonation attacks.

Proposition X.4. *Assume $\sqrt{q} \geq M \geq e\sqrt{\frac{n}{k}}$ and $T < M^2k$. Our scheme is secure against impersonation of a CLD, i.e. an adversary will fail with probability $1 - \frac{1}{q}$.*

Proof As stated in the previous proposition, impersonation of a verifier is not possible except with probability $\frac{1}{q}$ and an adversary would need to succeed at least k times to reconstruct the p' polynomial of a CLD. Moreover, eavesdropping the devices responses does not give a solution to reconstruct an identifier or to obtain information on an identifier, as stated in Proposition X.1. Furthermore corruption is not useful here as identifiers are not correlated between CLDs (following Definition 15, the adversary is not allowed to impersonate a corrupted CLD). The best choice for an adversary is thus to try at random. □

Replay attacks on the verifier side are not important from a security point of view as replaying a query does not give additional information to the adversary. However, they are prevented in the scheme to maintain privacy (with replay attacks, an adversary could track a device).

Privacy

Proposition X.5. *If $\sqrt{q} \geq M \geq e\sqrt{\frac{n}{k}}$ and $T < M^2k$, then our scheme is weak private.*

Proof We first prove the narrow-weak privacy; then, Lemma VIII.1 together with Proposition X.4 enables us to conclude. It is clear that all oracles are easy to simulate except SendCLD and SendVerifier (Result is not simulated in the narrow case). Concerning the latter, SendVerifier is used to generate an interrogation request; it is simulated simply by sending a random value. As PR instances are not distinguishable from random sequences (cf. [105]), an adversary cannot distinguish the requests from non-simulated ones.

Concerning SendCLD, the simulator needs to simulate the output of a CLD. For this, it can answer only on average to one request over M with a random value. As the adversary cannot impersonate a verifier, he cannot determine if a CLD is answering when beckoned or not. Neither can he distinguish the answered values from PR instances as above. □

Moreover, even if not forward private, as the identifiers are independently chosen among devices, the corruption of one device directly affects only this device. Although, this level of privacy could seem low, it is exactly what we intended to achieve and it is important to notice that contrary to the protocols described in [182], devices do not need the use of any internal random number generator to implement the protocol.

X.5 Advantages for very Low-Cost Devices

For low-cost devices, instead of storing the two polynomial identifiers p, p', we store directly the values $p(\alpha_1), \ldots, p(\alpha_n)$ and $p'(\alpha_1), \ldots, p'(\alpha_n)$ within the device. So doing, no computation is needed on the device side. Depending on the amount of memory available per device, we can also limit the number of such values by restricting ourselves to a basis of evaluation of size $L < n$, e.g. $(\alpha_1, \ldots, \alpha_L)$.

An additional advantage is that the scheme can be adapted simply to work over a noisy channel by storing encoded versions – through some error-correcting code – of these values $p(\alpha_1), \ldots, p(\alpha_L)$ and $p'(\alpha_1), \ldots, p'(\alpha_L)$ and the corresponding index $1, \ldots, L$. The devices will only have to compute the distance between the received message and the stored one.

Remark 50. It is also possible to further extend the scheme toward reaching forward privacy (equivalent to destructive privacy in this context of non-correlated identifiers): we store $L < k$ values for each identifier p, p' of degree at most $k - 1$ and erase the values $p(\alpha_j)$ and $p'(\alpha_j)$ after replying to the associated query. Because we erase the values after, a corruption will not give direct access to these values and because $L < k$, it is unconditionally impossible for an adversary to recover the missing values by polynomial interpolation. Hence, the destructive privacy is fulfilled. In this case, the false-positive rate should be quite small to avoid quick waste of the coupons of the devices.

X.6 Practical Parameters

For real-life low-cost CLDs, we can imagine a non-volatile memory of about $2^{18} = 256$k bits. We aim at a field size $q = 2^{64}$, which permits to store $2^{12} = 4096$ fields elements in the memory, *i.e.* 2048 evaluations of the two polynomials p_l, p'_l (which implies that the length $n \leq q-1$ of the corresponding code is $n = 2^{11}$).

With these parameters, we suggest the use of polynomials of dimension $k = 2^8$. Using such a dimension permits to define $q^k = 2^{64\times 256}$ possible polynomials; the number M of devices needed in the cloud around a verifier has then to be greater than $e \times \sqrt{\frac{n}{k}}$, i.e. at least 8. With $M = 256$, this leads to the restriction $T < 2^{24}$, which is automatically satisfied here as $T \leq Mn = 2^{19}$.

These parameters enable 2048 interrogations of the same device without compromising the device identity - both in terms of impersonation and of weak privacy.

Remark 51. We can suppress the identification-code structure, and replace it with a random one (*i.e.* replace $p(\alpha_i), p'(\alpha_i)$ by random $\beta_i, \beta'_i \in \{0,1\}^{\log_2 q}$).

However, instead of storing $k \cdot \log_2 q$ bits per device at the verifier's side, we need to store for each device the $n \cdot \log_2 q$ bits that are stored in it. With these parameters, this implies a storage space 8 times larger.

X.7 On the Threshold of Maximum-Distance Separable Codes

The security of the protocol of section X.3 is based on the hardness a well-studied decision problem [105]. Algorithms for polynomial reconstruction are of great interest, for cryptography, but also for coding theory. As it was shown in [121], the Guruswami-Sudan list-decoder [83] outputs a list limited to only one element in most cases when the number of correct coordinates is greater than $\sqrt{nk}$. This means that it should be possible to decode more than $n-\sqrt{nk}$ errors with a list-decoder, but no such algorithm exists yet.

This section aims at finding a threshold above which we know for certain that it is no longer possible to decode. This means that we switch from a computational assumption to information-theoretic security. This approach consists in looking at a usually ignored side of list-decoding. For a certain class of words x that are far enough from the code, we look at the radii r such that list-decoding x with radius r provides a list that is always lower-bounded by a large enough number. This differs from the literature concerning list-decoding, which usually looks for radii for which the size is always upper-bounded by a maximum list size, or tries to exhibit a counter-example.

The "large enough" list size can be obtained easily by imposing that Maximum-Likelihood Decoding to be most improbable. Indeed, if it is possible to list-decode a vector into a list of subexponential size, then the maximum-likelihood probability of error is bounded by the inverse of the list-size; conversely, if the maximum-likelihood probability of error is exponentially close to 1, then for almost all received vectors, the list size is huge. We therefore focus on the all-or-nothing behaviour of the ML decoder. Inspired by percolation theory [80], and code-applied graph theory [175], we show how it is possible to conservatively estimate, before, after, and around a threshold, the all-or-nothing probability of ML decoding.

We work on a n-dimensional space H; the weight of $x \in H$ is the number of non-zero coordinates $w(x) = d(x, 0)$, and its support is the set of all its non-zero coordinates: $supp(x) = \{i \in [\![1, n]\!] : x_i \neq 0\}$ (in other words, $w(x) = |supp(x)|$. The Hamming ball of radius r centred around $x \in H$ is the set of all vectors at a distance to x less than r, and is noted $B(x, r)$. The volume of such a ball is independent of x, and is noted $V(r)$. For a subset $U \subset H$, $\overline{U}$ is its complementary $\overline{U} = \{x \in H : x \notin U\}$.

The Threshold of a Code

The existence of a threshold is motivated by the classical question of percolation : given a graph, with a source, and a sink, and given the probability p for a "wet" node of the graph to "wet" an adjacent node, *what is the probability for the source to wet the sink*? It appears that this probability has a threshold effect; in other words, there exists a limit probability p_c such that, if $p > p_c$, then the sink is almost surely wet, and if $p < p_c$, then the sink is almost never wet. The threshold effect is illustrated in Fig. X.2.

This question can be transposed into the probability of error-correcting a code. Given a proportion of errors p, with a decoding algorithm, what is the probability of correctly recovering the sent codeword? It was shown in [192] that for every binary code, and every decoding algorithm, this probability also follows a threshold.

In this paper, we show that this property also applies to q-ary codes. In the following part, we show that the threshold behaviour that was seen on binary codes can be obtained again.

The Margulis-Russo Identity.

The technique used to derive threshold effects in discrete spaces is to integrate an isoperimetric inequality; for that, the Margulis-Russo identity is required.

If $H = \{0,1\}^n$ is the (binary)Hamming space, consider the measure $\mu_p : H \to [0,1]$ defined by $\mu_p(x) = p^{w(x)}(1-p)^{n-w(x)}$.

The number of limit-vectors of a subset $U \subset H$ is a function defined as

$$h_U(x) = |B(x,1) \cap \overline{U}| \text{ for } x \in U. \tag{X.7.1}$$

For $U \subset H$ such that U is increasing (*i.e.* if $x \in U$, and $y \geq x$, then $y \in U$ with $\geq$ defined component-wise), Margulis and Russo showed :

$$\frac{d\mu_p(U)}{dp} = \frac{1}{p}\int_U h_U(x)d\mu_p(x)$$

Let $q \in \mathbb{N}, q > 2$. The following shows that this equality also holds in $H_q = \{0, ...q-1\}^n$.

We redefine the measure function $\mu_p(x)$ over H_q by $\mu_p(x) = \left(\frac{p}{q-1}\right)^{w(x)}(1-p)^{n-w(x)}$. This definition is consistent with a measure, as

$$\mu_p(H_n) = \sum_{x \in H_q} \mu_p(x) = 1.$$

Note the inclusion $\subset$ to be the relation between a set and a (general) subset (*i.e.* for all X, $X \subset X$). The support inclusion generalises the component-wise $\leq$ that was used in the binary case.

Lemma X.1 (Margulis-Russo Identity over q-ary alphabets)**.** *Let U be an increasing subset of H_q, i.e. such that if $y \in U$, for all $x \in H_q$ such that $supp(y) \subset supp(x)$, then $x \in U$. Then*

$$\frac{d\mu_p(U)}{dp} = \frac{1}{p}\int_U h_U(x)d\mu_p(x)$$

where h_U is defined by (X.7.1).

Proof The proof of this lemma is an adaptation of Margulis' proof in [118]. For this, we use the notation:

- $[U,V] = |\{x,y\} \in U \times V : d(x,y) = 1|$ where $U,V \subset H_q$, is the number of links from U to V
- for $k \in [\![0,n]\!]$, $Z_k = \{x \in H_q : w(x) = k\}$,
- for $U \subset H_q$, $U_k = U \cap Z_k$ (U is the union of the U_k);
- $D_k = \sum_{x \in U_k} h_U(x)$ is the number of limit-vectors next to elements of weight k.

Trivially, $D_k = [U_k, Z_{k+1} - U_{k+1}] + [U_k, Z_{k-1} - U_{k-1}] + [U_k, Z_k - U_k]$. We now note that :

- $[U_k, Z_{k-1}] = |U_k| \cdot k$, as to go from U_k to Z_{k-1}, the only way (in one move) is to put one coordinate to 0;
- $[U_k, Z_{k+1}] = |U_k| \cdot (n-k)(q-1)$ with the same reasoning;
- $[U_k, Z_k - U_k] = [U_k, Z_{k+1} - U_{k+1}] = 0$ as U is increasing.
- Combining these equalities, we get $[U_k, U_{k+1}] = |U_k|(n-k)(q-1)$;
- $[U_k, Z_k] = 0$ as it is necessary to switch a non-zero coordinate to 0 and a zero to $\{1, ...q-1\}$.

Finally $D_k = [U_k, Z_{k-1}] - [U_k, U_{k-1}] = k|U_k| - (n-k+1)(q-1)|U_{k-1}|$ for $k > 0$ and $D_0 = 0$ (or $U = H_q$).

Back to the identity desired, we observe that

$$\begin{aligned}\int_U h_U(x) d\mu_p(x) &= \sum_{k=0}^{n} \sum_{x \in U_k} h_U(x) (\frac{p}{q-1})^k (1-p)^{n-k} \\ &= \sum_{k=0}^{n} D_k \left(\frac{p}{q-1}\right)^k (1-p)^{n-k}\end{aligned}$$

$$= \sum_{k=1}^{n} (k|U_k| - (n-k+1)(q-1)|U_{k-1}|) \cdot \left(\frac{p}{q-1}\right)^k (1-p)^{n-k}$$

$$= \textstyle\sum_{k=0}^{n} |U_k|(k - p\frac{n-k}{1-p}) \left(\frac{p}{q-1}\right)^k (1-p)^{n-k}$$

on the other hand,

$$\textstyle\frac{d\mu_p(U)}{dp} = \sum_{k=0}^{n} |U_k| \frac{d}{dp} \left(\left(\frac{p}{q-1}\right)^k (1-p)^{n-k} \right)$$

$$= \textstyle\sum_{k=0}^{n} |U_k| \left(\frac{p}{q-1}\right)^k (1-p)^{n-k} \left(\frac{k}{p} + \frac{-(n-k)}{1-p}\right)$$

Hence the identity. □

This lemma shows that the Margulis-Russo identity is also true on $\{0...(q-1)\}^n$; it was the keystone of the reasoning done in [175] to show an explicit form of the threshold behaviour of Maximum-Likelihood Error Correction.

A Threshold for Error-Decoding q-ary codes.

In the following, we use $\varphi(t) = \frac{1}{\sqrt{2\pi}}e^{-\frac{t^2}{2}}$ the normal distribution, $\Phi(x) = \int_{-\infty}^{x} \varphi(t)dt$ the accumulate normal function, and $\Psi(x) = \varphi(\Phi^{-1}(x))$ (so that $\forall x, \Psi(x) \cdot \Phi'^{-1}(x) = 1$).

A monotone property is a set $U \subset H_q$ such that U is increasing, or $\overline{U}$ is increasing.

Theorem X.1. *Let U be a monotone property of H_q. Suppose that $\exists \Delta \in \mathbb{N}^\star$: $\forall x \in U, h_U(x) = 0$ or $h_U(x) \geq \Delta$.*

Let $\theta \in [0,1]$ be (the unique real) such that $\mu_\theta(U) = \frac{1}{2}$. Let $g_\theta(p) = \Phi\left(\sqrt{2\Delta}(\sqrt{-\ln\theta} - \sqrt{-\ln p})\right)$.

Then the measure of U, $\mu_p(U)$ is bounded by :

$$\begin{array}{lcll} \mu_p(U) & \leq & g_\theta(p) & \textit{for } p \in (0;\theta] \\ \mu_p(U) & \geq & g_\theta(p) & \textit{for } p \in [\theta;1) \end{array}$$

Proof The proof is exactly the same as the one from [175]. The whole idea is to derive the upper-range:

$$\int_U \sqrt{h_U} d\mu_p \geq \sqrt{2\ln\frac{1}{p}}\Psi(\mu_p(U))$$

The integration of this equation, together with the Margulis-Russo lemma, gives the result. □

We remark that the non-decoding region of a given point, for a q-ary code, is an increasing region of $\mathbb{F}_q^n$. For linear codes, this non-decoding region can always be translated to that of 0 without loss of generality; let $U_0 = \{x \in \mathbb{F}_q^n \text{ s.t. } \exists c \in C, c \neq 0 : d(x,c) \leq d(x,0)\}$. The probability of error decoding of C is then $\mu_p(U_0)$.

For $x \in U_0$, we show that either $h_{U_0}(x) = 0$, or $h_{U_0}(x) \geq \frac{d}{2}$, where d is the minimal distance of C.

Indeed, if $h_{U_0}(x) > 0$, then there exists $c \in C, c \neq 0$ such that $d(x,c) \leq d(x,0)$, and $x_1 \in \overline{U_0}$ at Hamming distance 1 from x. The monotonic property of A_0 provides $|w(x_1) - w(x)| = 1$, and as x is further from 0 than x_1, $w(x_1) = w(x) - 1$. Then all the vectors obtained by replacing one of the coordinates of x by 0 are out of U_0; in particular, $h_{U_0}(x) \geq w(x)$. Let $d_c = w(c) \geq d$ be the weight of c; as x is nearer to c than to 0, $w(x) \geq \frac{d_c}{2}$. Thus the previous assertion.

Combining the previous results, we just showed that for any q-ary code, the probability of error is, as for binary codes, bounded by a threshold function. This can be expressed by the following theorem, which has the same form as the one showed in [175]:

Theorem X.2. *Let C be a code of any length, and of minimal distance d. Over the q-ary symmetric channel, with transition probability p, the probability of decoding error $P_e(p)$ associated with C is such that there exists a unique $p_c \in (0;1)$ such that $P_e(p_c) = \frac{1}{2}$, and P_e is bounded by:*

$$P_e(p) \lesseqgtr 1 - \Phi(\sqrt{d}(\sqrt{-\ln(1-p_c)} - \sqrt{-\ln(1-p)}))$$

The upper-bound ($\leq$) is true when $p \in]0;p_c]$; the lower-bound ($\geq$) is true when $p \in [p_c;1[$.

Even though linearity was asked so that all decoding regions are isometric, it is not a requirement for this theorem. Indeed, the bounding equations are true for every codeword c by replacing d by $\min_{c' \in C, c` \neq c} d(c,c')$. Assuming that the codewords sent are distributed in a uniform way over C, we thus obtain this result.

The behaviour of this function is illustrated in Fig X.2. Around $p \approx 0$ (actually, for all $p < p_c - \epsilon$, ϵ being the gap between the transitional to the stable behaviour of P_e), P_e is extremely flat above its limit 0; around $p \approx 1$ (and, symmetrically, for all $p > p_c + \epsilon$, P_e is extremely flat below its limit 1. Finally, around the threshold p_c, the slope is $\frac{\sqrt{d}}{\sqrt{2\pi}(1-p_c)}$, which is almost vertical when the minimal distance d is large.

Explicit Computation of the Threshold for Maximum-Distance Separable Codes

Here we only take interest in linear codes over $\mathbb{F}_q^n$.

Another Estimation of the Decoding Threshold.

By linearity, we can again without loss of generality assume that the sent codeword was the all-zero vector 0. It is possible to have a rough estimation of the probability of wrongly decoding with crossover probability p correctly a vector by computing the proportion of vectors $x \in \mathbb{F}_q^n$ of weight less or equal to np that are closer to a non-null codeword than to 0. Let $g(p)$ be this proportion.

$$g(p) = \frac{\mid \{x : \text{ s.t. } \exists c \in C, c \neq 0 : d(x,c) \neq w(x) \leq np\} \mid}{\mid \{x : w(x) \leq np\} \mid}.$$

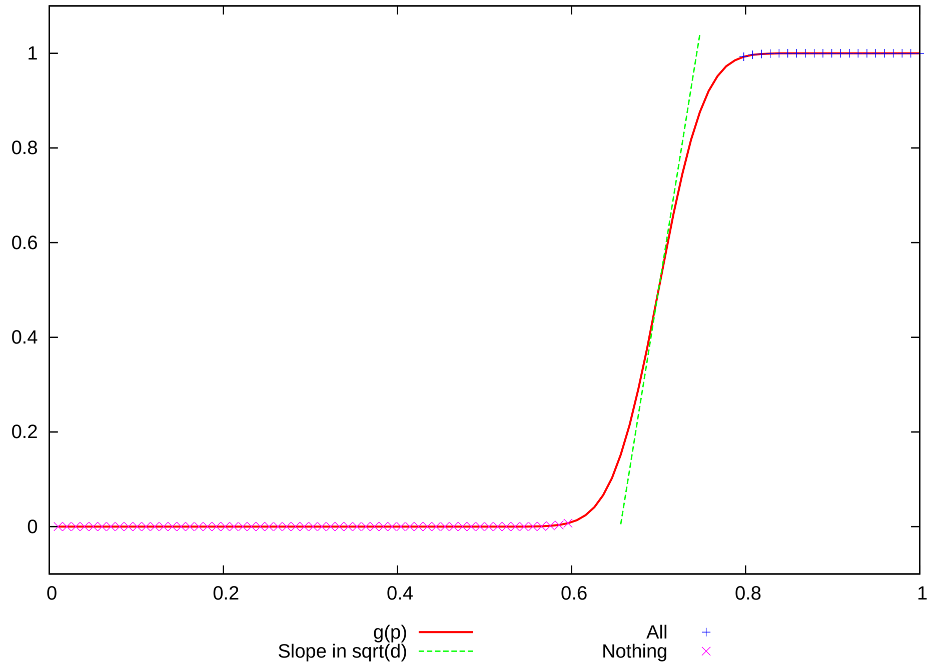


Figure X.2: Illustration of the threshold effect, $d = 400$, $p_c = 0.7$

Let $vol(q,n,t) = \frac{1}{n}\log_q(V(t))$. It is well known that when $t \leq n(1-\frac{1}{q})$, $vol(q,n,t) = H_q(\frac{t}{n}) + o_n(1)$, where $H_q(x) = -x\log_q x - (1-x)\log_q(1-x) + x\log_q(q-1)$ is the q-ary entropy of $x \in [0,1]$.

To compute the numerator, we suggest, for each codeword $c \in C$ that has a weight between d and $2pn$, to compute the number of vectors x that are nearer to c than to 0. This number actually only depends on the weight of c, and will be noted $\nu_{pn}(w(c))$. As there are $A_{w(c)}$ codewords of weight $w(c)$ in the code (with the standard notation), the function $g(p)$ can be approximated by:

$$g(p) \leq \frac{\sum_{l=d}^{2pn} A_l \nu_{pn}(l)}{q^{nvol(q,n,pn)}} \tag{X.7.2}$$

The different quantities used in this equation are illustrated in Fig X.3.

The number $\nu_t(w)$ is obtained in the following combinatorial way. Let c be a codeword of weight w. Let $x \in \mathbb{F}_q^n$ be a vector with the following constraints:

- $d(x,0) \leq t$, *i.e.* x is the result of the transmission of 0 with at most t errors.

- $d(x,0) \geq d(x,c)$, *i.e.* x is wrongly decoded.

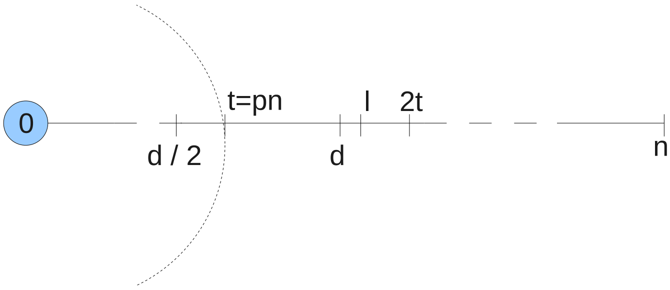


Figure X.3: Different quantities used in Eq X.7.2

We note α the number of coordinates i in x such that $x_i \neq c_i$ and $x_i = 0$; β is the number of coordinates i such that $x_i \neq c_i$ and $x_i \neq 0$; γ is the number of coordinates i such that $x_i \neq c_i$ and $c_i = 0$.

The previous constraints on x can be rewritten into the system (S):

$$(S) : \begin{cases} 1) & 0 \leq \alpha, \beta \leq w \\ 2) & 0 \leq \gamma \leq n - w \\ 3) & \gamma \leq t + \alpha - w \\ 4) & \beta + \gamma \leq t \\ 5) & 2\alpha + \beta \leq w \end{cases}$$

We then obtain

$$\nu_t(w) = \sum_{\alpha,\beta,\gamma} \binom{w}{\alpha + \beta} \binom{\alpha + \beta}{\beta} (q-2)^{\beta} \binom{n-w}{\gamma} (q-1)^{\gamma}.$$

Remark 52. It is easy to see that $\nu_t(w)$ is at most the volume of a ball of radius $w - \frac{d}{2}$; this estimation will be used in the next part.

Application to MDS codes.

Maximum-Distance Separable (**MDS**) Codes are codes such that their dimension k and minimal distance d fulfil the Singleton bound, so that:

$$k + d = n - 1.$$

A well known family of MDS codes are the Reed-Solomon codes, for which a codeword is made of the evaluation of a degree $k - 1$ polynomial over n field elements $\alpha_1, \ldots, \alpha_n$. Reed-Solomon codes over $\mathbb{F}_q$ can have a length up to $q - 1$, but shorter such codes are also MDS.

For MDS codes, the number A_l of codewords of given weight is known. This number is:

$$A_{n-i} = \sum_{j=1}^{n-1} (-1)^{j-i} \binom{n}{j} \binom{j}{i} (q^{k-j} - 1)$$

From this identity, it is easy to derive the more usable formula:

$$A_l = \binom{n}{l} \sum_{j=0}^{l-d} (-1)^j \binom{l}{j} (q^{1+l-d-j} - 1) \tag{X.7.3}$$

It is now possible to approximate quite nicely the error probability while under the threshold - indeed, the numerator and denominator are correct as long as a vector x is not close to 2 different codewords with a weight in the range $[\![d, pn]\!]$, *i.e.* as long as the list of codewords at a distance less than pn from x is reduced to a single element.

Short MDS Codes over Large Fields.

We now focus on the specific problem presented in section X.6. This setting is characterized by the following:

- The underlying code is a Reed-Solomon over a field $\mathbb{F}_q$;
- The field size q is very large for cryptographic reasons;
- The code length n is very short (with respect to q) as nq is the size of embedded low-cost devices' memory.

This application fits into the framework depicted in the previous sections. Moreover, the information "n much smaller than q" ($n = o(q)$) enables to compute an asymptotic first order estimation of the threshold in such codes.

Indeed, if $g(p) \leq f(p)$, then $g^{-1}(\frac{1}{2}) \geq f^{-1}(\frac{1}{2})$. We now compute an upper bound on $g(p)$, to derive an estimation on the threshold θ. More precisely, we aim at computing $\iota(p)$ the first-order value of $\log_q(g(p))$; then, $\iota^{-1}(0)$ is a lower-approximation of the threshold.

To estimate the weight enumerator A_l, we use formula $(X.7.3)$ to derive

$$A_l \leq \binom{n}{l} 2^l q^{1+l-d} \leq 2^{n+l} q^{1+l-d}.$$

The number of targeted vectors for each codeword $\nu_t(l)$ is not easy to evaluate; we note its first order development $\log_q \nu_t(l) := n\mu(l,t) + o_q(1)$, so that $\nu_t(l) \leq q^{n\mu(l,t)} \cdot o_q(q)$. (Here, the term $o(q)$ is a bounded by a polynomial in n.) We know that

$$0 \leq n\mu(l,t) \leq l - \frac{d}{2} \tag{X.7.4}$$

Combining these elements with equation (X.7.2), we obtain

$$g(p) \leq \sum_{l=d}^{2pn} q^{(n+l)\log_q(2)+1+l-d+n\mu(l,pn)-nvol(q,n,pn)}.$$

As $vol(q,n,t) = H_q(\frac{t}{n}) + o_n(1) = \frac{t}{n} + o_q(1)$, the first order of $g(p)$ is bounded by: $\log_q g(p) \leq \max_{l\in[d,pn]}\left(1 + l - d - pn + n\mu(l,pn)\right) + o_q(1)$.

The bounding (X.7.4) of μ shows that the right-hand side of this inequality is between $1+pn-d$ and $1+3pn-\frac{3d}{2}$, which shows that the threshold $g^{-1}(\frac{1}{2})$ is asymptotically between $\frac{\delta}{2}$ and δ.

Unfortunately, a more precise evaluation of μ strongly depends on the context. Indeed, according to Section X.7,

$$\nu(l,t) = o_q(q) \cdot \max_{\alpha,\beta,\gamma:(S)} q^{\beta+\gamma}\binom{n-l}{\gamma}\binom{l}{\alpha+\beta}\binom{\alpha+\beta}{\beta}.$$

This maximum can be obtained by evaluating the term to be maximized on all vertices of the polytope defined by the system (S) ((S) is made of 9 inequalities of 3 unknown, the vertices are obtained by selecting 3 of these equations, thus at most $\binom{9}{3} = 84$ vertices); however, it is not possible to exhibit here a general answer as the solution depends on the minimal distance of the code, *i.e.* on the rate of the Reed-Solomon code.

Numerical Application to a $(2048, 256, 1793)_{2^{64}}$ MDS Code.

In the case of a code over a finite field of reasonable dimension, it is possible to exactly compute the ratio that approximates the Maximum Likelihood threshold. However, the exact threshold cannot be easily computed yet; it is still an open problem related to the list-decoding capacity of Reed-Solomon codes.

We therefore used the NTL open-source library [165] to compute the values A_l, $\nu_t(l)$ and $|B(t)|$ in order to have an accurate enough approximation of the the function $g(p)$ described earlier. The parameters are those that were proposed in section X.6, and show that the decoding threshold of such a code is between 0.8 and 0.875.

The slope around the threshold is around 115, so for p "small" (in fact, a bit smaller than p_c) $g(p)$ is very near to 0, while as p goes to 1, $g(p)$ is much greater than the maximum probability of 1. This was predicted earlier, and expresses the fact that the list-size of radius pn is always greater than 1. The threshold value $g^{-1}(\frac{1}{2}) \approx \iota^{-1}(0)$ is a lower-bound for the threshold of the code, though the intuition says that this lower-bound is pretty near to the real threshold.

New Security Considerations

The initial goal was to revise the conditions of security of the construction depicted in section X.3, and it can be rephrased in these terms: from a received vector x of $\mathbb{F}_q^n$, for what parameters is the size of the list of radius pn exponentially large?

This problem can be reduced to that of the threshold probability of a linear error-correcting code. Indeed, below the threshold of the code, when the minimal distance of the code is large enough, the error decoding probability of the code is exponentially small, and it is exponentially close to 1 above the threshold. For our class of parameters, ensuring that the error rate is above the threshold is enough to show the security of the scheme. The threshold behaviour is demonstrated or q-ary codes as well as for binary codes, and we can only lower-bound the threshold of the MDS codes.

Applying these results to the initial problem, we show that the threshold for a (highly) truncated Reed-Solomon code over a finite field $\mathbb{F}_{2^{64}}$ is very near to normalized the minimal distance $d = n - k + 1$ of this code. This result is coherent with the estimation of [121], and with the NP-hardness stated in [84].

We conclude that to switch from an algorithmic assumption (the hardness of the Polynomial Reconstruction Problem, see [103]) to Information-Theoretical security, we recommend to raise the dimension k of the underlying code. This lowers the decoding threshold of the code; the downside is that storage of a codeword is more costly.

Concluding Remarks

> I am not young enough to know everything.
>
> James M. Barrie

This thesis uses elements from many fields of research to construct security protocols for identification. We will end this document by summing up the constructions that we proposed first, and then the contribution of each topic in our research.

New Constructions

We essentially provided constructions for biometric (Part 3) and for device (Part 4) identification.

We based the biometric identification protocols on modern cryptography, which has three high-level aspects:

- Public-Key Cryptography. The dichotomy between public and secret key applies well to our case of use, for which the end-users and the server have very asymmetric roles. In the model of biometric databases and identification, the security of the data and the privacy of the users coincide; the use of secure public key primitives is a good way of handling our problems.

- Symmetric Cryptography. The interesting notion of Searchable Symmetric Encryption provides a ready-to-use infrastructure for biometric identification. One of our main contribution was to adjust the data to the primitive. Nevertheless, one should not overlook the fact that we also designed a satisfying security model for the kind of data we consider. For this model, the sheer association of classical blocs cannot lead to a biometric identification protocol respectful of the users' privacy. Our contribution relies on a clever composition of encryption and non-cryptographic functions.

- Hardware-based Security. Using Match-on-Card is not a novel idea as far as biometrics are concerned, but MOC-based identification would take frightening delays without a speed-up. Once again, we had to work on this primitive while keeping in mind the personal aspect of all the data that would transit. We especially did an architectural effort to assemble the elements in the correct order.

We focused on a particular device-based identification protocol, based on identification codes. For any given identification code, the protocol is easily implementable, and applying it to Reed-Solomon based codes provides a whole range of tools studied in the fields of cryptography and coding theory. We were therefore able to show that for the parameters of interest, it is possible to transform this identification protocol into a mutual authentication protocol, as secure as the hardness of the polynomial reconstruction.

This last point caught our attention, and we showed that the hardness of the polynomial reconstruction problem is linked, by counting arguments, to the decoding threshold of the whole class of maximum-distance separable codes. For the reader interested in a precise estimation of this threshold, we provided a simple enough formula that estimates the threshold for truncated Reed-Solomon codes.

Perspectives

This work answers some questions, but many doors were also opened on some question that need answering.

The protocols we provided for a secure biometric identification take as input biometric templates as binary strings of a fixed length, which use as a resemblance score the Hamming distance. This model is successful on the iris, but the algorithm used on the fingerprints are not as efficient as one could wish. The first open problem that needs to be answered is the following:

Problem 3 (Efficient Minutiae-Based Fingerprint Quantization)**.** Let $\mathcal{M} = \{(x_1, y_1, \theta_1), \dots (x_n, y_n, \theta_n)\}$ be a set of minutiae.

From $\mathcal{M}$, compute $b \in \{0,1\}^N$ such that the matching of two templates b and b' is done by computing $d(b, b')$.

The algorithms presented in chapter II do not offer a solution to this problem, as the quantization methods described take as input a picture and use the pattern to derive a binary string, and not the minutiae. During our work, we studied the solutions proposed, *e.g.* [66, 64, 63, 170]. However, [64, 63] is based on a graphical model for minutiae, and that graphical model is either too realistic, in which case the message-passing algorithm underneath is too costly, or not realistic enough, which renders poor results. [170] is a practical solution for minutia quantization, but is not resistant to fingerprint misalignment.

[66] provides a solution that uses well the local structure of the minutiae map, but does not require a global coherence to discriminate fingerprint templates. A mitigated solutions that would take into account both the local and global coherence would be a good candidate to solve such a problem, and would serve well the protocols of chapters V, VI and VII.

As we mentioned in chapter I, biometric fusion enables to improve the biometric error rates very efficiently, since biometric traits are supposed to be independent (it is commonly assumed that the iris and the fingerprint patterns have no common origin). To use both templates in a single cryptographic scheme would, for a low increase in the computational time and storage space, make the system much more practical and acceptable by the users. This is the second opening of this thesis: to find an efficient way to do biometric fusion in the encrypted domain, for authentication or identification.

Problem 4 (Biometric Fusion in Cryptographic Protocols)**.** Let $\mathcal{B}_1, \mathcal{B}_2$ be two sets of biometric templates of different sorts, with matching algorithms $\mathsf{m}_1 : \mathcal{B}_1 \times \mathcal{B}_1 \to \mathbb{R}$, $\mathsf{m}_2 : \mathcal{B}_2 \times \mathcal{B}_2 \to \mathbb{R}$. Let $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be a fusion rule, acting as a score classifier for templates from $\mathcal{B}_1$ and $\mathcal{B}_2$.

Design an authentication or identification protocol that uses biometric templates from $\mathcal{B}_1$ and $\mathcal{B}_2$ using the classifier f.

The cryptographic state of the art is rich and offers many possibilities, only a fraction of which is used in the protocols designed to handle biometrics of part 3. In the field of public key cryptography, an interesting primitive is the Identity-Based Encryption (IBE) [162]. This primitive allows a sender to encrypt a message with the public key of the receiver, the public key being the receiver's identity (for example, his e-mail address). The idea of using biometrics as input to IBE was proposed by Sahai and Waters [155], and several constructions were put forward to match this "Fuzzy Identity-Based Encryption".

How this primitive could be applied to biometric identification is still an open subject that needs further investigating.

The design of cryptographic protocols goes through specifications, models, and the use of building blocs the security of which is based on a hard problem. In part 4 we designed a protocol based on the difficulty of Polynomial Reconstruction, a classical problem, used in cryptographic protocols, but also in other primitives like threshold-secret sharing [161]. The difficulty of this problem is therefore of a great interest. In chapter X we investigated a combinatoric way to find out which instances of the Polynomial Reconstruction problem were hard, and our results were close to those found in [121], but the question remains: *can we do better than this estimation?*

Problem 5 (Hardness of the Polynomial Reconstruction Problem)**.** Let $k, n, q \in \mathbb{N}^\star$, q a prime power, $1 \leq k \leq n \leq q-1$, $\{\alpha_1, \ldots, \alpha_n\} \subset \mathbb{F}_q$.

Determine $H \subset \mathbb{F}_q^n$ the set of all vectors x for which the list-decoding of x in the $[n, k, n-k+1]$-Reed-Solomon code defined by the α_i.

Kiayias and Yung's hypothesis [103] state that H is the set of points at a distance greater than $n - \sqrt{nk}$ from the code. Guruswami and Vardy [84] established that decoding the code's black holes (points at distance $n-k$ from the code) is NP-hard.

The status of the still missing range needs further study.

Bibliography

[1] J. J. Abrams. Star Trek. Paramount Pictures, May 2009.

[2] Michael Adjedj, Julien Bringer, Hervé Chabanne, and Bruno Kindarji. Biometric identification over encrypted data made feasible. In *Fifth International Conference on Information Systems Security*, Dec. 2009.

[3] Rudolf Ahlswede and Gerhard Dueck. Identification via channels. *Information Theory, IEEE Transactions on*, 35(1):15–29, Jan 1989.

[4] Rudolf Ahlswede and Bart Verboven. On identification via multiway channels with feedback. *Information Theory, IEEE Transactions on*, 37(6):1519–1526, Nov 1991.

[5] Ross Anderson, Mike Bond, Jolyon Clulow, and Sergei Skorobogatov. Cryptographic processors – a survey. *Proceedings of the IEEE*, 94(2):357–369, 2006.

[6] Alex Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Commun. ACM*, 51(1):117–122, 2008.

[7] Russell Ang, Reihaneh Safavi-Naini, and Luke McAven. Cancelable key-based fingerprint templates. In Boyd and Nieto [22], pages 242–252.

[8] Sigal Ar, Richard J. Lipton, Ronitt Rubinfeld, and Madhu Sudan. Reconstructing algebraic functions from mixed data. *SIAM J. Comput.*, 28(2):487–510, 1998.

[9] A. M. Bazen and Raymond N. J. Veldhuis. Likelihood-ratio-based biometric verification. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1):86–94, 2004.

[10] Asker M. Bazen and Sabih H. Gerez. Systematic methods for the computation of the directional fields and singular points of fingerprints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):905–919, 2002.

[11] Asker M. Bazen and Raymond N. J. Veldhuis. Detection of cores in fingerprints with improved dimension reduction. In *4th IEEE Benelux Signal Processing Symposium(SPS-2004), Hilvarenbeek, The Netherlands*, pages 41–44, 2004.

[12] Mihir Bellare, Alexandra Boldyreva, and Adam O'Neill. Deterministic and efficiently searchable encryption. In *CRYPTO'07*, pages 535–552, 2007.

[13] Mihir Bellare and Phillip Rogaway. Entity authentication and key distribution. In *CRYPTO '93: Proceedings of the 13th annual international cryptology conference on Advances in cryptology*, pages 232–249, New York, NY, USA, 1993. Springer-Verlag New York, Inc.

[14] J.M. Berger. A note on error detection codes for asymmetric channels. *Information and Control*, 4(1):68 – 73, 1961.

[15] Toby Berger. *Rate-Distortion Theory, A Mathematical Basis for Data Compression.* Prentice-Hall, Inc., 1971.

[16] John Bethencourt, Dawn X. Song, and Brent Waters. New constructions and practical applications for private stream searching (extended abstract). In *IEEE Symposium on Security and Privacy*, pages 132–139. IEEE Computer Society, 2006.

[17] Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*, 13(7):422–426, 1970.

[18] Ruud M. Bolle, Jonathan H. Connell, and Nalini K. Ratha. Biometric perils and patches. *Pattern Recognition*, 35(12):2727–2738, 2002.

[19] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In *EUROCRYPT'04*, pages 506–522, 2004.

[20] Dan Boneh, Eyal Kushilevitz, Rafail Ostrovsky, and William. E. Skeith III. Public key encryption that allows PIR queries. In Alfred Menezes, editor, *CRYPTO*, volume 4622, pages 50–67. Springer, 2007.

[21] Raj Chandra Bose and Dwijendra Kumar Ray-Chaudhuri. On a class of error correcting binary group codes. *Information and Control*, 3(1):68–79, March 1960.

[22] C. Boyd and J. M. González Nieto, editors. *Information Security and Privacy, 10th Australasian Conference, ACISP 2005, Brisbane, Australia, July 4-6, 2005, Proceedings*, volume 3574 of *LNCS*. Springer, 2005.

[23] Xavier Boyen. Reusable cryptographic fuzzy extractors. In *CCS '04: Proceedings of the 11th ACM conference on Computer and communications security*, pages 82–91, New York, NY, USA, 2004. ACM.

[24] Gilles Brassard and Louis Salvail. Secret-key reconciliation by public discussion. In *EUROCRYPT '93: Workshop on the theory and application of cryptographic techniques on Advances in cryptology*, pages 410–423, Secaucus, NJ, USA, 1994. Springer-Verlag New York, Inc.

[25] J. Bringer and H. Chabanne. An authentication protocol with encrypted biometric data. In Serge Vaudenay, editor, *AFRICACRYPT*, volume 5023 of *Lecture Notes in Computer Science*, pages 109–124. Springer, 2008.

[26] Julien Bringer, Hervé Chabanne, Gérard Cohen, and Bruno Kindarji. RFID key establishment against active adversaries. In *First IEEE International Workshop on Information Forensics and Security*, 2009.

[27] Julien Bringer, Hervé Chabanne, Gérard Cohen, Bruno Kindarji, and Gilles Zemor. Theoretical and practical boundaries of binary secure sketches. *Information Forensics and Security, IEEE Transactions on*, 3(4):673–683, Dec. 2008.

[28] Julien Bringer, Hervé Chabanne, Gérard Cohen, Bruno Kindarji, and Gilles Zémor. Optimal iris fuzzy sketches. In *First IEEE International Conference on Biometrics: Theory, Applications and Systems*, 2007.

[29] Julien Bringer, Hervé Chabanne, and Thomas Icart. Improved privacy of the tree-based hash protocols using physically unclonable function. In Rafail Ostrovsky, Roberto De Prisco, and Ivan Visconti, editors, *Proceedings of the 6th International Conference on Security and Cryptography for Networks – SCN'08*, volume 5229 of *Lecture Notes in Computer Science*, pages 77–91, Amalfi, Italy, August 2008. Springer.

[30] Julien Bringer, Hervé Chabanne, Tom A.M. Kevenaar, and Bruno Kindarji. Extending match-on-card to local biometric identification. In *Biometric ID Management and Multimodal Communication*, 2009.

[31] Julien Bringer, Hervé Chabanne, and Bruno Kindarji. The best of both worlds: Applying secure sketches to cancelable biometrics. *Science of Computer Programming*, 74:43–51, December 2008.

[32] Julien Bringer, Hervé Chabanne, and Bruno Kindarji. Anonymous identification with cancelable biometrics. In *International Symposium on Image and Signal Processing and Analysis*, 2009.

[33] Julien Bringer, Hervé Chabanne, and Bruno Kindarji. Error-tolerant searchable encryption. In *Communications, 2009. ICC '09. IEEE International Conference on*, pages 1–6, June 2009.

[34] Julien Bringer, Hervé Chabanne, and Bruno Kindarji. Identification with encrypted biometric data. *Security and Communication Networks*, to appear, 2010.

[35] Julien Bringer, Hervé Chabanne, Gérard Cohen, and Bruno Kindarji. Private interrogation of devices via identification codes. *INDOCRYPT '09: Proceedings of the 10th International Conference on Cryptology in India*, pages 272–289, 2009.

[36] Julien Bringer, Hervé Chabanne, Malika Izabachène, David Pointcheval, Qiang Tang, and Sébastien Zimmer. An application of the Goldwasser-Micali cryptosystem to biometric authentication. In J. Pieprzyk, H. Ghodosi, and E. Dawson, editors, *ACISP*, volume 4586 of *Lecture Notes in Computer Science*, pages 96–106. Springer, 2007.

[37] Julien Bringer, Hervé Chabanne, David Pointcheval, and Qiang Tang. Extended private information retrieval and its application in biometrics authentications. In Feng Bao, San Ling, Tatsuaki Okamoto, Huaxiong Wang, and Chaoping Xing, editors, *CANS*, volume 4856 of *Lecture Notes in Computer Science*, pages 175–193. Springer, 2007.

[38] Julien Bringer, Hervé Chabanne, David Pointcheval, and Sébastien Zimmer. An application of the Boneh and Shacham group signature scheme to biometric authentication. In Kanta Matsuura and Eiichiro Fujisaki, editors, *IWSEC*, volume 5312 of *Lecture Notes in Computer Science*, pages 219–230. Springer, 2008.

[39] Julien Bringer, Hervé Chabanne, and Qiang Tang. An application of the Naccache-Stern knapsack cryptosystem to biometric authentication. In *AutoID*, pages 180–185. IEEE, 2007.

[40] Ileana Buhan, Jeroen Breebaart, Jorge Guajardo, Koen de Groot, Emile Kelkboom, and Ton Akkermans. A quantitative analysis of crossmatching resilience for a continuous-domain biometric encryption technique. In *First International Workshop on Signal Processing in the EncryptEd Domain, SPEED'09*, 2009.

[41] Ileana Buhan, Jeroen Doumen, Pieter Hartel, Qiang Tang, and Raymond Veldhuis. Embedding renewable cryptographic keys into continuous noisy data. In *Information and Communications Security*, volume 5308 of *Lecture Notes in Computer Science*, pages 294–310. Springer Berlin / Heidelberg, 2008.

[42] Ileana Buhan, Jeroen Doumen, Pieter Hartel, and Raymond Veldhuis. Fuzzy extractors for continuous distributions. *ASIACCS '07: Proceedings of the 2nd ACM symposium on Information, computer and communications security*, pages 353–355, 2007.

[43] Jin W. Byun, Dong H. Lee, and Jongin Lim. Efficient conjunctive keyword search on encrypted data storage system. In Andrea S. Atzeni and Antonio Lioy, editors, *EuroPKI*, volume 4043, pages 184–196. Springer, 2006.

[44] Christian Cachin and Ueli M. Maurer. Linking information reconciliation and privacy amplification. In *Advances in Cryptology – EUROCRYPT'94*, volume 950 of *Lecture Notes in Computer Science*, pages 266–274. Springer Berlin / Heidelberg, 1995.

[45] Mario Cagalj, Srdjan Capkun, RamKumar Rengaswamy, Ilias Tsigkogiannis, Mani Srivastava, and Jean-Pierre Hubaux. Integrity (I) codes: Message integrity protection and authentication over insecure channels. *IEEE Symposium on Security and Privacy*, 0:280–294, 2006.

[46] Srdjan Capkun, Mario Cagalj, Ramkumar Rengaswamy, Ilias Tsigkogiannis, Jean-Pierre Hubaux, and Mani Srivastava. Integrity codes: Message integrity protection and authentication over insecure channels. *IEEE Transactions on Dependable and Secure Computing*, 5(4):208–223, Oct.-Dec. 2008.

[47] CASIA. Chinese academy of science, institute of automation. URL : http://www.sinobiometrics.com/Database.htm.

[48] Hervé Chabanne, Gérard Cohen, and Bruno Kindarji. On iterated logarithm solutions to identification protocols. In *IEEE Information Theory Workshop*, 2010.

[49] Hervé Chabanne and Guillaume Fumaroli. Noisy cryptographic protocols for low-cost RFID tags. *IEEE Transactions on Information Theory*, 52(8):3562–3566, Aug. 2006.

[50] Yan-Cheng Chang and Michael Mitzenmacher. Privacy preserving keyword searches on remote encrypted data. In *Applied Cryptography and Network Security Conference (ACNS)*, 2005.

[51] Benny Chor, Eyal Kushilevitz, Oded Goldreich, and Madhu Sudan. Private information retrieval. *J. ACM*, 45(6):965–981, 1998.

[52] Gérard Cohen and Hans G. Schaathun. Upper bounds on separating codes. *IEEE Transactions on Information Theory*, 50(6):1291–1294, June 2004.

[53] Reza Curtmola, Juan A. Garay, Seny Kamara, and Rafail Ostrovsky. Searchable symmetric encryption: Improved definitions and efficient constructions. In *CCS '06: Proceedings of the 13th ACM conference on Computer and communications security*, pages 79–88. ACM, 2006.

[54] Joan Daemen and Vincent Rijmen. *The Design of Rijndael*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2002.

[55] John Daugman. High confidence visual recognition of persons by a test of statistical independence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1148–1161, 1993.

[56] John Daugman. The importance of being random: statistical principles of iris recognition. *Pattern Recognition*, 36(2):279–291, 2003.

[57] George I. Davida, Yair Frankel, and Brian J. Matt. On enabling secure applications through off-line biometric identification. *IEEE Symposium on Security and Privacy*, 0:0148, 1998.

[58] Robert H. Deng, Yingjiu Li, Andrew C. Yao, Moti Yung, and Yunlei Zhao. A new framework for RFID privacy. Cryptology ePrint Archive, Report 2010/059, 2010.

[59] Tim Dierks and Eric Rescorla. The transport layer security (TLS) protocol version 1.1. RFC 4346, Internet Engineering Task Force, April 2006.

[60] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 38(1):97–139, 2008.

[61] Yevgeniy Dodis and Adam Smith. Correcting errors without leaking partial information. In *STOC '05: Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 654–663, New York, NY, USA, 2005. ACM.

[62] Vivekanad Dorairaj, Natalia A. Schmid, and Gamal Fahmy. Performance evaluation of non-ideal iris based recognition system implementing global ICA encoding. In *IEEE International Conference on Image Processing, 2005*, volume 3, pages III – 285–8, sept. 2005.

[63] Stark C. Draper, Ashish Khisti, Emin Martinian, Anthony Vetro, and Jonathan S. Yedidia. Secure storage of fingerprint biometrics using Slepian-Wolf codes. In *Information Theory and Applications Workshop*, 2007.

[64] Stark C. Draper, Ashish Khisti, Emin Martinian, Anthony Vetro, and Jonathan S. Yedidia. Using distributed source coding to secure fingerprint biometrics. Technical Report TR 2007-005, Mitsubishi Electrical Research Laboratories, January 2007.

[65] Krishnan Eswaran. Identification via channels and constant-weight codes. `http://www.eecs.berkeley.edu/~ananth/229BSpr05/Reports/KrishEswaran.pdf`.

[66] Faisal Farooq, Ruud M. Bolle, Tsai-Yang Jea, and Nalini Ratha. Anonymous and revocable fingerprint recognition. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.

[67] Michael L. Fredman, János Komlós, and Endre Szemerédi. Storing a sparse table with O(1) worst case access time. *ACM*, 31, 1984.

[68] Yves Fregnac and Michel Imbert. Development of neuronal selectivity in primary visual cortex of cat . *Physiological Reviews*, 64(1):325–434, 1984.

[69] Benjamin C.M. Fung, Khalil Al-Hussaeni, and Ming Cao. Preserving RFID data privacy. In *IEEE International Conference on RFID – RFID 2009*, Orlando, Florida, USA, April 2009.

[70] Fingerprint verification competition. `http://bias.csr.unibo.it/fvc2000/`.

[71] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *CRYPTO'84*, pages 10–18, 1984.

[72] Martin J. Gander and Ueli M. Maurer. On the secret-key rate of binary random variables. In *IEEE International Symposium on Information Theory, 1994.*, pages 351–, Jun-1 Jul 1994.

[73] William I. Gasarch. A survey on private information retrieval. http://www.cs.umd.edu/ gasarch/pir/pir.html.

[74] Craig Gentry and Zulfikar Ramzan. Single-database private information retrieval with constant communication rate. In Luís Caires, Giuseppe F. Italiano, Luís Monteiro, Catuscia Palamidessi, and Moti Yung, editors, *ICALP*, volume 3580 of *Lecture Notes in Computer Science*, pages 803–815. Springer, 2005.

[75] Yael Gertner, Yuval Ishai, Eyal Kushilevitz, and Tal Malkin. Protecting data privacy in private information retrieval schemes. In *ACM Symposium on Theory of Computing*, pages 151–160, 1998.

[76] Philippe Godlewski and Gérard Cohen. Some cryptographic aspects of womcodes. In *Lecture notes in computer sciences; 218 on Advances in cryptology—CRYPTO 85*, pages 458–467, New York, NY, USA, 1986. Springer-Verlag New York, Inc.

[77] Alwyn Goh and David Ngo Chek Ling. Computation of cryptographic keys from face biometrics. In Antonio Lioy and Daniele Mazzocchi, editors, *Communications and Multimedia Security*, volume 2828 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 2003.

[78] Eu-Jin Goh. Secure indexes. Cryptology ePrint Archive, Report 2003/216, 2003. `http://eprint.iacr.org/2003/216/`.

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

[80] Geoffrey R. Grimmett. Percolation. Springer, 1997.

[81] Information Security Group. RFID security and privacy lounge. http://www.avoine.net/rfid/.

[82] Venkatesan Guruswami and Madhu Sudan. Improved decoding of Reed-Solomon and algebraic-geometry codes. *IEEE Transactions on Information Theory*, 45(6):1757–1767, 1999.

[83] Venkatesan Guruswami and Madhu Sudan. Reflections on "improved decoding of Reed-Solomon and algebraic-geometric codes". IEEE Information Theory Newsletter, 2002.

[84] Venkatesan Guruswami and Alexander Vardy. Maximum-likelihood decoding of Reed-Solomon codes is NP-hard. In *SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 470–478, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.

[85] Gael Hachez, Francois Koeune, and Jean-Jacques Quisquater. Biometrics, access control, smart cards: a not so simple combination. In *Proceedings of the fourth working conference on smart card research and advanced applications on Smart card research and advanced applications*, pages 273–288, Norwell, MA, USA, 2001. Kluwer Academic Publishers.

[86] Neil M. Haller. The S/KEY one-time password system. In *Proceedings of the Internet Society Symposium on Network and Distributed Systems*, pages 151–157, 1994.

[87] Feng Hao, Ross Anderson, and John Daugman. Combining crypto with biometrics effectively. *IEEE Transactions on Computers*, 55(9):1081–1088, 2006.

[88] Feng Hao, John Daugman, and Piotr Zielinski. A fast search algorithm for a large fuzzy database. *IEEE Transactions on Information Forensics and Security*, 3(2):203–212, June 2008.

[89] Amir Herzberg, Stanislaw Jarecki, Hugo Krawczyk, and Moti Yung. Proactive secret sharing or: How to cope with perpetual leakage. In Don Coppersmith, editor, *CRYPTO*, volume 963, pages 339–352. Springer, 1995.

[90] Karen Hollingsworth, Kevin W. Bowyer, and Patrick J. Flynn. All Iris Code bits are not created equal. In *IEEE International Conference on Biometrics: Theory, Applications, and Systems, 2007. BTAS 2007.*, Sept 2007.

[91] Sumin Hong, Woongryul Jeon, Seungjoo Kim, Dongho Won, and Choonsik Park. The vulnerabilities analysis of fuzzy vault using password. In *FGCN '08: Proceedings of the 2008 Second International Conference on Future Generation Communication and Networking*, pages 76–83, Washington, DC, USA, 2008. IEEE Computer Society.

[92] Piotr Indyk. Nearest neighbors in high-dimensional spaces. In Jacob E. Goodman and Joseph O'Rourke, editors, *Handbook of Discrete and Computational Geometry, chapter 39*. CRC Press, 2004. 2rd edition.

[93] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Symposium on the Theory Of Computing*, pages 604–613, 1998.

[94] Andrew Teoh Beng Jin, David Ngo Chek Ling, and Alwyn Goh. An integrated dual factor authenticator based on the face data and tokenised random number. In David Zhang and Anil K. Jain, editors, *ICBA*, volume 3072 of *Lecture Notes in Computer Science*, pages 117–123. Springer, 2004.

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

[96] Ari Juels and Martin Wattenberg. A fuzzy commitment scheme. In *ACM Conference on Computer and Communications Security*, pages 28–36, 1999.

[97] Ari Juels and Stephen A. Weis. Defining strong privacy for RFID. In *PERCOMW*, pages 342–347. IEEE Computer Society, 2007.

[98] Joseph M. Kahn, Y Howard Katz, and Kristofer S. J. Pister. Emerging challenges: Mobile networking for smart dust. *Journal of Communications and Networks*, 2:188–196, 2000.

[99] Jens-Peter Kaps, Kaan Yuksel, and Berk Sunar. Energy scalable universal hashing. *IEEE Transactions on Computers*, 54(12):1484–1495, Dec. 2005.

[100] Tom A.M. Kevenaar, Geert-Jan Schrijen, Michiel van der Veen, Anton H.M. Akkermans, and Fei Zuo. Face recognition with renewable and privacy preserving binary templates. In *AUTOID '05: Proceedings of the Fourth IEEE Workshop on Automatic Identification Advanced Technologies*, pages 21–26, Washington, DC, USA, 2005. IEEE Computer Society.

[101] Dalia Khader. Public key encryption with keyword search based on K-resilient IBE. In Marina L. Gavrilova, Osvaldo Gervasi, Vipin Kumar, Chih J. K. Tan, David Taniar, Antonio Laganà, Youngsong Mun, and Hyunseung Choo, editors, *ICCSA (3)*, volume 3982, pages 298–308. Springer, 2006.

[102] Aggelos Kiayias and Moti Yung. Polynomial reconstruction based cryptography. In Serge Vaudenay and Amr M. Youssef, editors, *Selected Areas in Cryptography*, volume 2259 of *Lecture Notes in Computer Science*, pages 129–133. Springer, 2001.

[103] Aggelos Kiayias and Moti Yung. Cryptographic hardness based on the decoding of Reed-Solomon codes. In Peter Widmayer, Francisco Triguero Ruiz, Rafael Morales Bueno, Matthew Hennessy, Stephan Eidenbenz, and Ricardo Conejo, editors, *International Colloquium on Automata, Languages and Programming*, volume 2380 of *Lecture Notes in Computer Science*, pages 232–243. Springer, 2002.

[104] Aggelos Kiayias and Moti Yung. Cryptographic hardness based on the decoding of Reed-Solomon codes with applications. In *Electronic Colloquium on Computational Complexity (ECCC)*, 2002.

[105] Aggelos Kiayias and Moti Yung. Cryptographic hardness based on the decoding of Reed-Solomon codes. *IEEE Transactions on Information Theory*, 54(6):2752–2769, June 2008.

[106] Adam Kirsch and Michael Mitzenmacher. Distance-sensitive Bloom filters. In *Algorithm Engineering & Experiments*, Jan 2006.

[107] Adams Kong, King-Hong Cheung, David Zhang, Mohamed Kamel, and Jane You. An analysis of biohashing and its variants. *Pattern Recogn.*, 39(7):1359–1368, 2006.

[108] Kaoru Kurosawa and Takuya Yoshida. Strongly universal hashing and identification codes via channels. *Information Theory, IEEE Transactions on*, 45(6):2091–2095, Sep 1999.

[109] Eyal Kushilevitz, Rafail Ostrovsky, and Yuval Rabani. Efficient search for approximate nearest neighbor in high dimensional spaces. In *Symposium on the Theory Of Computing*, pages 614–623, 1998.

[110] Yongjin Lee, Yongki Lee, Yunsu Chung, and Kiyoung Moon. One-time templates for face authentication. In *ICCIT '07: Proceedings of the 2007 International Conference on Convergence Information Technology*, pages 1818–1823, Washington, DC, USA, 2007. IEEE Computer Society.

[111] Stan Z. Li and Anil K. Jain, editors. *Encyclopedia of Biometrics*. Springer US, 2009.

[112] Jean-Paul. M. G. Linnartz and Pim Tuyls. New shielding functions to enhance privacy and prevent misuse of biometric templates. In J. Kittler and M. S. Nixon, editors, *Audio- and Video-Based Biometric Person Authentication*, volume 2688 of *LNCS*, pages 393–402. Springer, 2003.

[113] Helger Lipmaa. An oblivious transfer protocol with log-squared communication. In Jianying Zhou, Javier Lopez, Robert H. Deng, and Feng Bao, editors, *ISC*, volume 3650, pages 314–328. Springer, 2005.

[114] Xiaomei Liu, Kevin W. Bowyer, and Patrick J. Flynn. Iris recognition and verification experiments with improved segmentation method. In *AutoID'2005, 17-18 October 2005, Buffalo, New York*, 2005.

[115] F. Jessie MacWilliams and Neil J. A. Sloane. *The Theory of Error-Correcting Codes (North-Holland Mathematical Library)*. North Holland, January 1983.

[116] Dario Maio, Davide Maltoni, Raffaele Cappelli, Jim L. Wayman, and Anil K. Jain. FVC2000: fingerprint verification competition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):402–412, 2002.

[117] Davide Maltoni, Dario Maio, Anil K. Jain, and Salil Probhakar. *Handbook of Fingerprint Recognition*, volume 22 of *Springer, New York, 2003*. Cambridge University Press, New York, NY, USA, 2004.

[118] Gregori A. Margulis. Probabilistic characteristics of graphs with large connectivity. *Problemy Peredači Informacii*, 10(2):101–108, 1974.

[119] Ueli M. Maurer. Information-theoretically secure secret-key agreement by not authenticated public discussion. In *EUROCRYPT'97*, 1997.

[120] Ueli M. Maurer and Stefan Wolf. Secret-key agreement over unauthenticated public channels II: the simulatability condition. *IEEE Transactions on Information Theory*, 49(4):832–838, 2003.

[121] Robert J. McEliece. On the average list size for the Guruswami-Sudan decoder. In *ISCTA03*, 2003.

[122] Jennifer L. Mnookin. The Achilles' heel of fingerprints. *The Washington Post*, May 29:A27, 2004.

[123] David Molnar and David Wagner. Privacy and security in library RFID: issues, practices, and architectures. In *CCS*, pages 210–219. ACM, 2004.

[124] Pierre Moulin and Ralf Koetter. A framework for the design of good watermark identification codes. In Edward J. Delp III and Ping Wah Wong, editors, *SPIE*, volume 6072, page 60721H. SPIE, 2006.

[125] D.E. Muller. Application of boolean algebra to switching circuit design and to error detection. *EEE Transactions on Electronic Computers*, 3:6–12, 1954.

[126] Karthik Nandakumar, Abhishek Nagar, and Anil K. Jain. Hardening fingerprint fuzzy vault using password. In *Advances in Biometrics*, volume 4642 of *Lecture Notes in Computer Science*, pages 927–937. Springer Berlin / Heidelberg, 2009.

[127] National Institute of Standards and Technology (NIST). MINEX II - an assessment of match-on-card technology. `http://fingerprint.nist.gov/minex/`.

[128] National Institute of Standards and Technology (NIST). Iris Challenge Evaluation. `http://iris.nist.gov/ICE`, 2005.

[129] Sam Newfield. Fingerprints don't lie, February 1951.

[130] Ching Yu Ng, Willy Susilo, Yi Mu, and Reihaneh Safavi-Naini. RFID privacy models revisited. In Sushil Jajodia and Javier López, editors, *ESORICS*, volume 5283 of *Lecture Notes in Computer Science*, pages 251–266. Springer, 2008.

[131] Miyako Ohkubo, Koutarou Suzuki, and Shingo Kinoshita. RFID privacy issues and technical challenges. *Commun. ACM*, 48(9):66–71, 2005.

[132] Alon Orlitsky. Worst-case interactive communication - I. two messages are almost optimal. *IEEE Transactions on Information Theory*, 36(5):1111–1126, Sep 1990.

[133] Alon Orlitsky. Worst-case interactive communication - II: Two messages are not optimal. *IEEE Transactions on Information Theory*, 37(4):995–1005, 1991.

[134] Rafail Ostrovsky and William E. Skeith III. Algebraic lower bounds for computing on encrypted data. Cryptology ePrint Archive, Report 2007/064, 2007. `http://eprint.iacr.org/`.

[135] Rafail Ostrovsky and William E. Skeith III. A survey of single database PIR: Techniques and applications. Cryptology ePrint Archive: Report 2007/059, 2007.

[136] Rafail Ostrovsky and Victor Shoup. Private information storage (extended abstract). In *ACM Symposium on Theory of Computing*, pages 294–303, 1997.

[137] Khaled Ouafi and Raphael C.-W. Phan. Traceable Privacy of Recent Provably-Secure RFID Protocols. In *Proceedings of the 6th International Conference on Applied Cryptography and Network Security — ACNS 2008*, volume 5037 of *Lecture Notes in Computer Science*, pages 479–489, New York City, New York, USA, June 2008. Springer.

[138] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In J. Stern, editor, *Advances in Cryptology, Proceedings of EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238. Springer, 1999.

[139] Radu-Ioan Paise and Serge Vaudenay. Mutual authentication in RFID: security and privacy. In Masayuki Abe and Virgil D. Gligor, editors, *ASIACCS*, pages 292–299. ACM, 2008.

[140] King F. Pang and Abbas El Gamal. Communication complexity of computing the Hamming distance. *SIAM Journal on Computing*, 15(4):932–947, 1986.

[141] Ying-Han Pang, Andrew Teoh Beng Jin, and David Ngo Chek Ling. Two-factor cancelable biometrics authenticator. *J. Comput. Sci. Technol.*, 22(1):54–59, 2007.

[142] PEARS. Privacy Ensuring Affordable RFID System. European Project.

[143] P. Jonathon Phillips, Hyeonjoon Moon, Patrick Rauss, and Syed A. Rizvi. The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, 2000.

[144] Rainer Plaga. On biometric keys, their information content and proper use. Conference on Biometric Feature Identification and Analysis, Göttingen, Sept. 2007.

[145] Nalini K. Ratha, Jonathan H. Connell, and Ruud M. Bolle. Enhancing security and privacy in biometrics-based authentication systems. *IBM Systems Journal*, 40(3):614–634, 2001.

[146] Nalini K. Ratha, Jonathan H. Connell, Ruud M. Bolle, and Sharat Chikkerur. Cancelable biometrics: A case study in fingerprints. In *International Conference on Pattern Recognition*, pages 370–373. IEEE Computer Society, 2006.

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

[148] Irving S. Reed. A class of multiple-error-correcting codes and their decoding scheme. *IEEE Transactions on Information Theory*, 4:38–42, 1954.

[149] Melanie R. Rieback. *Security and Privacy of Radio Frequency Identification.* PhD thesis, Vrije Universiteit, Amsterdam, The Netherlands, 2008.

[150] Ronald L. Rivest, Len Adleman, and Michael L. Dertouzos. On data banks and privacy homomorphisms. In *Foundations of Secure Computation*, pages 169–177. Academic Press, 1978.

[151] Chris Roberts. Biometric attack vectors and defences. *Computers & Security*, 26(1):14–25, 2007.

[152] Atri Rudra. *List Decoding and Property Testing of Error Correcting Codes.* PhD thesis, University of Washington, 2007.

[153] Eun-Kyung Ryu and Tsuyoshi Takagi. Efficient conjunctive keyword-searchable encryption. In *AINA Workshops*, pages 409–414. IEEE Computer Society, 2007.

[154] Ahmad-Reza Sadeghi, Ivan Visconti, and Christian Wachsmann. User privacy in transport systems based on RFID e-tickets. In *Workshop on Privacy in Location-Based Applications – PILBA'08*, Malaga, Spain, October 2008.

[155] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In R. Cramer, editor, *EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, pages 457–473. Springer, 2005.

[156] Marie Sandström. *Liveness Detection in Fingerprint Recognition Systems.* PhD thesis, Linköping University, Department of Electrical Engineering, 2004.

[157] Berry Schoenmakers and Pim Tuyls. Efficient binary conversion for Paillier encrypted values. In Serge Vaudenay, editor, *EUROCRYPT*, volume 4004, pages 522–537. Springer, 2006.

[158] Walter J. Schreier and Terrance E. Boult. Cracking fuzzy vaults and biometric encryption. In *Proceedings of the Biometric Symposium*, 2007.

[159] Ridley Scott. Blade Runner. Warner Bros, June 1982.

[160] Saeed Sedghi, Peter van Liesdonk, Jeroen M. Doumen, Pieter H. Hartel, and Willem Jonker. Adaptively secure computationally efficient searchable symmetric encryption. Technical Report TR-CTIT-09-13, Centre for Telematics and Information Technology, University of Twente, April 2009.

[161] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[162] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Proceedings of CRYPTO 84 on Advances in cryptology*, pages 47–53, New York, NY, USA, 1985. Springer-Verlag New York, Inc.

[163] Claude E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 623–656, 1948.

[164] Claude E. Shannon. Communication theory of secrecy systems. *The Bell System Technical Journal*, 28:656–715, 1949.

[165] Victor Shoup. NTL: A library for doing number theory.

[166] Koen Simoens, Pim Tuyls, and Bart Preneel. Privacy weaknesses in biometric sketches. In *Security and Privacy, 2009 30th IEEE Symposium on*, pages 188 –203, may 2009.

[167] Sarah Spiekermann and Sergei Evdokimov. Privacy Enhancing Technologies for RFID - A Critical Investigation of State of the Art Research. In *IEEE Privacy and Security*, 2009.

[168] Steven Spielberg. Minority Report. DreamWorks, 20th Century Fox, June 2002.

[169] Zhe-Nan Sun, Tie-Niu Tan, and Xian-Chao Qiu. Graph matching iris image blocks with local binary pattern. In David Zhang and Anil K. Jain, editors, *International Conference on Biometrics*, volume 3832, pages 366–372. Springer, 2006.

[170] Y. Sutcu, S. Rane, J.S. Yedidia, S.C. Draper, and A. Vetro. Feature extraction for a slepian-wolf biometric system using LDPC codes. *IEEE International Symposium on Information Theory, 2008. ISIT 2008*, pages 2297–2301, July 2008.

[171] Yagiz Sutcu, Shantanu Rane, Jonathan S. Yedidia, Stark C. Draper, and Anthony Vetro. Feature extraction for a Slepian-Wolf biometric system using LDPC codes. *IEEE International Symposium on Information Theory, 2008.*, ISIT 2008.:2297–2301, July 2008.

[172] Yagiz Sutcu, Shantanu Rane, Jonathan S. Yedidia, Stark C. Draper, and Anthony Vetro. Feature transformation of biometric templates for secure biometric systems based on error correcting codes. *Computer Vision and Pattern Recognition Workshop*, 0:1–6, 2008.

[173] Qiang Tang, Julien Bringer, Hervé Chabanne, and David Pointcheval. A formal study of the privacy concerns in biometric-based remote authentication schemes. In *Information Security Practice and Experience*, volume 4991 of *Lecture Notes in Computer Science*, pages 56–70. Springer Berlin / Heidelberg, 2008.

[174] R. M. Tanner. A recursive approach to low-complexity codes. *IEEE Trans. on Information Theory*, 27:533–547, 1981.

[175] Jean-Pierre Tillich and Gilles Zémor. Discrete isoperimetric inequalities and the probability of a decoding error. *Comb. Probab. Comput.*, 9(5):465–479, 2000.

[176] Pim Tuyls, Anton H. M. Akkermans, Tom A. M. Kevenaar, Geert Jan Schrijen, Asker M. Bazen, and Raymond N. J. Veldhuis. Practical biometric authentication with template protection. In Takeo Kanade, Anil K. Jain, and Nalini K. Ratha, editors, *AVBPA*, volume 3546, pages 436–446. Springer, 2005.

[177] Pim Tuyls and Jasper Goseling. Capacity and examples of template-protecting biometric authentication systems. In D. Maltoni and A. K. Jain, editors, *ECCV Workshop BioAW*, volume 3087 of *Lecture Notes in Computer Science*, pages 158–170. Springer, 2004.

[178] Yoshifumi Ueshige and Kouichi Sakurai. A proposal of one-time biometric authentication. In Hamid R. Arabnia and Selim Aissi, editors, *Security and Management*, pages 78–83. CSREA Press, 2006.

[179] Umut Uludag and Anil K. Jain. Attacks on biometric systems: a case study in fingerprints. In *SPIE-EI 2004, Security, Seganography and Watermarking of Multimedia Contents VI*, 2004.

[180] Umut Uludag and Anil K. Jain. Fuzzy fingerprint vault, 2004.

[181] Maneesh Upmanyu, Anoop M. Namboodiri, K. Srinathan, and C. V. Jawahar. Efficient biometric verification in encrypted domain. In Massimo Tistarelli and Mark S. Nixon, editors, *ICB*, volume 5558 of *Lecture Notes in Computer Science*, pages 899–908. Springer, 2009.

[182] Serge Vaudenay. On privacy models for RFID. In Kaoru Kurosawa, editor, *ASIACRYPT*, volume 4833 of *Lecture Notes in Computer Science*, pages 68–87. Springer, 2007.

[183] Evgeny Verbitskiy, Pim Tuyls, Chibuzo Obi, Berry Schoenmakers, and Boris Skoric. Key extraction for general non-discrete signals. `http://eprint.iacr.org/2009/303`, 2009.

[184] Sergio Verdu and Victor K.-W. Wei. Explicit construction of optimal constant-weight codes for identification via channels. *Information Theory, IEEE Transactions on*, 39(1):30–36, Jan 1993.

[185] Markus Weber. Frontal face dataset. `http://www.vision.caltech.edu/html-files/archive`, California Institute of Technology, 1999.

[186] Stephen A. Weis, Sanjay E. Sarma, Ronald L. Rivest, and Daniel W. Engels. Security and privacy aspects of low-cost radio frequency identification systems. In *Security in Pervasive Computing*, pages 201–212. Springer, 2003.

[187] Niclas Wiberg. *Codes and Decoding on general Graphs*. PhD thesis, Linkoping University, Linkoping, Sweden, 1996.

[188] the free encyclopedia Wikipedia. Privacy. http://en.wikipedia.org/wiki/Privacy.

[189] Chaoping Xing. Asymptotic bounds on frameproof codes. *Information Theory, IEEE Transactions on*, 48(11):2991–2995, Nov 2002.

[190] Andrew Teoh Beng Jin Ying-Han Pang and David Ngo Chek Ling. Replaceable biometrics authenticator. In *M2USIC 2005*, 2005.

[191] Kaan Yuksel. Universal hashing for ultra-low-power cryptographic hardware applications. Master's thesis, Worcester Polytechnic Institute, 2004.

[192] Gilles Zémor. Threshold effects in codes. In Gérard D. Cohen, Simon Litsyn, Antoine Lobstein, and Gilles Zémor, editors, *Algebraic Coding*, volume 781 of *Lecture Notes in Computer Science*, pages 278–286. Springer, 1993.

Appendix A

Information Theory Tools

> It is a very sad thing that nowadays there is so little useless information.
>
> Oscar Wilde

Information Theory was born with Claude Shannon's works on communication [163]. His problematic was to define a mathematical measure on the information that is transmitted from one person to the other. To answer this question, Shannon introduced the notion of entropy, inspired by thermodynamics.

A.1 Measures on Probabilistic Sources

Let $\mathcal{X}$ be the (finite) set of messages that a player Alice wants to send; a random variable X that takes values in $\mathcal{X}$ denotes the repartition of the messages' frequencies.

Definition 23. The *information* carried by a message $x \in \mathcal{X}$ is

$$-\log_2 \Pr\left[X = x\right].$$

The *entropy* of the random variable X, noted $H(X)$, is the average information of X:

$$H(X) = \mathbb{E}_x(-\log_2 \Pr\left[X = x\right]).$$

The entropy of a source is measured in bits.

For example, if $\mathcal{X} = \{0, 1\}$ and a random variable X takes value 1 with probability p and 0 with probability $1 - p$, then the information carried by a 1 is $-\log_2 p$, the information carried by a 0 is $-\log_2(1 - p)$ and the entropy of X is $H(X) = -p\log_2 p - (1 - p)\log_2(1 - p)$. This particular case gives the definition of the Shannon binary entropy.

Definition 24. Let $p \in [0,1]$. The Shannon binary entropy of p, noted $h_2(p)$[1], is given by the expression:

$$h_2(p) = -p \log_2 p - (1-p)\log_2(1-p)$$

The entropy is maximal for $p = \frac{1}{2}$, in which case $h_2(p) = 1$. In other words, the entropy of a binary source is maximal if the source takes value 1 or 0 with the same probability. In that case, it is a 1-bit source.

If $X = (X_1, X_2)$ is a random variable over $\mathcal{X}_1 \times \mathcal{X}_2$ such that X_1 and X_2 are independent random variables, the entropy of X is given by $H(X) = H(X_1) + H(X_2)$.

A particular case is for $\mathcal{X} = \{0,1\}^n$, when all the bits of X are independent and identically distributed, with $\Pr\left[X^{(i)} = 1\right] = (1-p)$. In that case, the entropy of X is $H(X) = nh_2(p)$. This entropy is maximal, once again, if X covers uniformly the space $\{0,1\}^n$.

Let us introduce briefly the other notions of entropy introduced in the document:

Definition 25. Let X, Y be random variables over $\mathcal{X}$ and $\mathcal{Y}$ respectively. For $y \in \mathcal{Y}$, the variable $(X|y)$ ("X knowing y") is a random variable with distribution law $\Pr\left[(X|y) = (x|y)\right] = \Pr\left[X = x|y\right]$; the entropy of $(X|y)$ is $H(X|Y=y)$.

The conditional entropy of X knowing Y, noted $H(X|Y)$, is defined by:

$$H(X|Y) = \mathbb{E}_y\left(H(X|Y=y)\right)$$

Conditional entropy provides a measure on the missing information on X when we know Y. If X and Y are independent, then the distribution law of $(X|y)$ is the same for all y, and is the law of X. That means that the conditional entropy $H(X|Y)$ is equal to $H(X)$ if X and Y are independent.

If fact, $H(X|Y) \leq H(X)$ for all random variables X and Y and the equality holds only if X and Y are independent.

Definition 26. Let X be a random variable over $\mathcal{X}$. The *min-entropy* of X, noted $H_\infty(X)$, is the information carried by the most probable realisation of X:

$$H_\infty(X) = -\log_2 \max_x \Pr\left[X = x\right]$$

This definition is useful in cryptography. Indeed, an adversary is more likely to look for the weakest link in the scheme that try to attack the general case[2].

[1] In this document, this entropy might be noted in another fashion to avoid confusion with hash functions.

[2] In a cryptanalysis, if the messages are not uniformly distributed, then the most probable messages carry more information than the others

Definition 27. Let X, Y be random variables over $\mathcal{X}$ and $\mathcal{Y}$ respectively. The *average min-entropy of X knowing Y* is given by:

$$\overline{H}_\infty(X|Y) = -\log_2 \mathbb{E}_y(\max_x \Pr[X = x|Y = y]).$$

This quantity, used *e.g.* in Secure Sketches, denotes the maximal information that leaks on X given Y.

Like any entropy, the average min-entropy of a random variable X knowing Y can only decrease when a deterministic function is applied; in other words:

Proposition A.1. *Let X be a random variable over $\mathcal{X}$ and $f : \mathcal{X} \to \mathcal{Z}$ be a deterministic function. Let Y be a random variable over $\mathcal{Y}$. In this setting,*

$$\overline{H}_\infty(X|Y) \geq \overline{H}_\infty(f(X)|Y).$$

Proof Let $x \in \mathcal{X}$, then $\Pr[X = x] \leq \Pr[f(X) = f(x)]$. In particular, for all $y \in \mathcal{Y}$, $\Pr[X = x|Y = y] \leq \Pr[f(X) = f(x)|Y = y]$, which leads to $\max_x \Pr[X = x|Y = y] \leq \max_x \Pr[X = x|Y = y]$.

The proposition comes from the non-increasing character of $-\log_2 \mathbb{E}$. □

As a similar property, if there is a $\lambda \geq 1$ such that $\forall x \in \mathcal{X}, \Pr[X = x] \leq \frac{\Pr[f(X)=f(x)]}{\lambda}$, then the average min-entropy of X knowing Y can be lower-bounded by:

$$\overline{H}_\infty(X|Y) \geq \overline{H}_\infty(f(X)|Y) + \log_2 \lambda.$$

The proof of this statement is the same to that of Proposition A.1.

A.2 Shannon's Theorem

Shannon describes a channel between two people as a transition probability law $W : \mathcal{Y} \times \mathcal{X} \to [0, 1]$. For a message $x \in \mathcal{X}$ to be sent, the probability for a receiver to read $y \in \mathcal{Y}$ is $W(y, x)$.

Let X be a random variable over $\mathcal{X}$. The result of the transmission of X in the channel W is a random variable Y; the mutual information of X and Y is defined as follow.

Definition 28. Let X, Y be random variables over $\mathcal{X}$ and $\mathcal{Y}$ respectively. The *mutual information* of X and Y is defined as:

$$I(X : Y) = H(X) - H(X|Y)$$

If X and Y are independent, then their mutual information is zero.

From the mutual information of X and Y the definition of the channel capacity follows.

Definition 29. Let $\mathcal{X}, \mathcal{Y}$ be two finite alphabets. Let $W : \mathcal{Y} \times \mathcal{X} \to [0,1]$ be a transition law from $\mathcal{X}$ to $\mathcal{Y}$. For X a random variable over $\mathcal{X}$, we note Y_X the random variable whose probability law is:

$$\Pr[Y = y] = \sum_{x \in \mathcal{X}} W(y,x) \Pr[X = x].$$

The *capacity of the channel* W, noted $\kappa(W)$, is the maximal mutual information between $\mathcal{X}$ and $\mathcal{Y}_{\mathcal{X}}$.

$$\kappa(W) = \max_X I(X : Y_X).$$

The binary symmetric channel (BSC) of transition probability p is the channel from $\{0,1\}$ to $\{0,1\}$ defined by $W_p(0,0) = W_p(1,1) = 1 - p$ and $W_p(1,0) = W_p(0,1) = p$. A classical result is that the capacity of W_p is $\kappa(W_p) = 1 - h_2(p)$.

We finally state Shannon's fundamental theorem of communication theory.

Theorem A.1. *Let W be a channel from $\mathcal{X}$ to $\mathcal{Y}$.*

- *Let $\epsilon > 0$. For a sufficiently large N, it is possible to transmit a message from $\mathcal{X}^N$ to $\mathcal{Y}^N$ with maximal probability of block error less than ϵ, provided that the set of messages X is such that $\frac{1}{N}\log_2 |X| < \kappa(W)$.*
- *if $\frac{1}{N}\log_2 |X| > \kappa(W)$ then for all decoding algorithms, the probability of block error is arbitrarily close to 1.*

This fundamental theorem states that there exist error-correcting codes of rate near the capacity that enable vanishing error probability after decoding. Finding these codes and corresponding decoding algorithms is an open problem.

Appendix B

Notions on Error-Correcting Codes

> How is an error possible in mathematics?
>
> Henri Poincaré

This appendix only recalls well-known notions on Error-Correcting Codes that are used throughout this manuscript. Interested readers should refer to a complete document such as [115].

B.1 Preliminaries

The Hamming distance is defined over any alphabet:

Definition 30. Let A be a finite set, $n \in \mathbb{N}^\star$; the *Hamming distance* $d_H : A^n \times A^n \to [\![0, n]\!]$ counts the number of differences between vectors $x = (x_i)_i$ and $y = (y_i)_i$:

$$\forall x, y \in A^n, d_H(x, y) = \sum_{i=1}^{n} \chi_{x_i}(y_i).$$

When the context is clear, the Hamming distance d_H will simply be noted d.

B.2 Definitions

An Error-Correcting Code (**ECC**, sometimes simply called "code") is, *stricto sensu*, a set of vectors of a finite field. Practical Error-Correcting Codes have however encoding and decoding functions, that allows to correct a number of errors. The original definition of an ECC follows.

Definition 31. A (n, m) *Error-Correcting Code* over an alphabet A of cardinality q is a subset C of A^n, of cardinality q^m. Elements of C are called codewords.

An encoder E is a deterministic function from A^m to A^n; a decoder D is a deterministic function from A^n to $A^m \cup \perp$ where the symbol $\perp$ means that the decoding failed.

E and D must verify the identity $\forall x \in A^m, D(E(x)) = x$.

C has minimal distance $d = \min_{c_1, c_2 \in C} d_H(c_1, c_2)$.

If C is a (n, m) code with minimal distance d, then a possible decoder D is the decoder that returns a codeword c on input x if $d_H(x, c) \leq \frac{d-1}{2}$, and $\perp$ otherwise.

Linear Error-Correcting Codes

The most useful class of ECC are the linear ECC. The base alphabet for these is a finite field $\mathbb{F}$ of cardinality q. In this case, the definition of a linear Error-Correcting Code is adapted in the following way:

Definition 32. A $[n, k, d]_q$ *linear Error-Correcting Code* is a linear subspace C of the vector space $\mathbb{F}^n$, of *dimension* k. The *minimal distance* d of C is the minimal weight of non-null vectors $c \in C$.

One advantage of using linear Error-Correcting Codes is that many properties follow from linear algebra. In particular, for each linear ECC C, there exists a full-rank *generating matrix* $G \in \mathbb{F}^{n \times k}$ that provides easily an encoding method from $\mathbb{F}^k$ to $\mathbb{F}^n$. Moreover, there exists a full-rank *parity-check* matrix $H \in \mathbb{F}^{n \times (n-k)}$ that defines the *dual code* $C^\top$, and that nullifies all elements of C.

B.3 Boundaries on Error-Correcting Codes

We here recall some well known limits on codes.

A boundary that appear in this document is the Singleton bound; it is a classical combinatoric bound on the size of codes:

Proposition B.1 (Singleton Bound)**.** *If C is a (n, K) q-ary code of minimal distance d, then its size K is such that:*

$$K \leq q^{n-d+1}.$$

For a linear code of dimension k, this can be rewritten as:

$$k \leq n - d + 1.$$

Codes that fulfil the Singleton bound are called Maximum-Distance Separable codes, among which a well-known family are the Reed-Solomon codes.

This bound can be easily refined to get the following expression.

Proposition B.2 (Hamming Bound)**.** *If C is a (n, K) q-ary code of minimal distance d, $t = \lfloor \frac{d-1}{2} \rfloor$, then its size K is such that:*

$$K \leq \frac{q^n}{\sum_{i=0}^{t} \binom{n}{k} (q-1)^k}.$$

In particular, if C is a $[n, k, d]$ linear binary code,

$$2^k \leq \frac{2^k}{\sum_{i=0}^{t} \binom{n}{k}}.$$

Codes fulfilling this bound are called perfect. For these codes, the spheres of radius t around the codewords cover entirely the space A^n.

There are many other bounds that can be found in the literature. Codes are a useful tool for theoretic and practical reasons, but considering these bounds enable to design theoretical constructions knowing the ins and outs.

B.4 Classical Codes

Here are some constructions of codes that are used in this document. This section does not intend to present an exhaustive list of the codes available in the literature, but to give key elements on the design of codes.

Reed and Muller Codes

Reed and Muller Codes [148, 125] are binary codes that are defined by evaluating all multivariate polynomials of a given degree over all the possible (binary) values of their inputs.

Definition 33 (Reed-Muller Codes)**.** Let m, r be non-null integers. Define $k = \sum_{i=0}^{r} \binom{m}{i}$.

The Reed-Muller code of order r in m variables is the $[2^m, k, 2^{m-r}]$ code consisting of the evaluations of all multivariate polynomials $P \in \mathbb{F}_2[x_1, \ldots, x_m]$ of degree less than r over $\mathbb{F}_2^m$.

In particular, a Reed-Muller code of order 1 in m variables is a $[2^m, m+1, \frac{2^m}{2}]$ code.

Reed and Solomon Codes

These codes are q-ary, and have the property of fulfilling the Singleton bound.

Definition 34 (Reed-Solomon Codes)**.** Let $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{F}_q^n$ be n different non-null elements of the base field. Let $k \leq n$.

The Reed-Solomon code defined over α is the $[n, k, (n-k)+1]_q$ code defined by the codewords $c_p = (p(\alpha_1), \ldots, p(\alpha_n))$ where $p \in \mathbb{F}_{\shortparallel}[X]$ is a polynomial of degree less than $k - 1$.

From this definition, it follows that the maximal length of a q-ary Reed-Solomon code is $q - 1$.

It is possible to decode these codes using an algebraic method, such as the Berlekamp-Massey algorithm, or to list-decode a received vector, with *e.g.* the Guruswami-Sudan algorithm. The latter enables to decode many vectors beyond the $\frac{d-1}{2}$ correction capacity.

LDPC Codes

These codes are defined by their parity-check matrix H. They are usually binary, though q-ary LDPC are being studied.

Definition 35 (LDPC Codes)**.** Let $H \in \mathbb{F}_2^{n\times(n-k)}$ be a *sparse* binary matrix. Then $C = \{x \in \mathbb{F}_{\not\succeq}{}^n$ s.t. $H.x = 0$ is the *Low Density Parity Check* code defined by H.

LDPC are particularly interesting because there exist an iterative decoding algorithm that enable to decode with only the parity matrix H. The *belief propagation* algorithm returns a solution that is very near to Maximum-Likelihood Decoding.

This algorithm is a graph-based message-passing algorithm, that looks for the Maximum-A-Posteriori codeword given a received vector.

Concatenated and Product Codes

It is possible to create a new codes out of two different codes. We here describe two similar constructions: concatenated codes and product codes.

Concatenated codes are obtained by applying an *inner* and *outer* code to some message. This results to a code of larger length and larger dimension (the overall rate is smaller), for which there exists better decoder than just the naïve two-pass decoder.

Product codes are concatenated codes that can be represented in a specific way. A codeword of a product code is a matrix whose lines are codewords of the first code and whose columns are codewords of the second code. Once again, there exists algorithms that go beyond the naive decoding by line then by column. Chapter IV provides such an algorithm for binary product codes.

Appendix C

Basic Notions of Cryptography

> The best weapon of a dictatorship is secrecy, but the best weapon of a democracy should be the weapon of openness.
>
> Niels Bohr

Facing the task of providing enough cryptographic background to the completeness of this document, we provide in this appendix some key elements to understand the cryptographic primitives and considerations mentioned.

C.1 Cryptographic Primitives

Public-Key Cryptosystems

A cryptosystem is given by three functions:

- $(pk, sk) \leftarrow \mathsf{Setup}(1^\ell)$ generates the keys used by the encryption and decryption functions; pk is the public key and sk the secret key. ℓ is the security parameter.
- $c \leftarrow \mathsf{Enc}_{pk}(m)$ is the encryption of the message m into the cipher c, with the public key pk. This function can be deterministic or probabilistic. In order to lighten the notations, pk is often omitted.
- $m' \leftarrow \mathsf{Dec}_{sk}(c)$ is the deterministic decryption of the cipher c with the secret key sk. Here again, sk is often omitted.

Many public key cryptosystems were designed since the 1970s and this appendix does not aim at exhaustively enumerate them. We however will give as example the El Gamal cryptosystem, as it is used in some of this thesis' constructions.

Definition 36 (El Gamal Cryptosystem [71])**.** The system is defined by the three functions:

- Setup: Let G be a cyclic group of order p where p is a ℓ-bit prime, and $g \in G$ be a generator of G. Let $x \in [\![0, p-1]\!]$ be a random integer and $h = g^x$ an element of G. $\mathsf{Setup}(1^\ell)$ returns $pk = (G, p, g, h)$ and $sk = x$.

- Enc_{pk}: take as input a message $m \in [\![0, p-1]\!]$ and outputs its encryption $c = (g^y, m \cdot h^y)$ where $y \in [\![0, p-1]\!]$ is a random value.

- Dec_{sk}: take as input a cipher $c = (c_1, c_2)$ and returns $m' = c_2 \cdot c_1^{-x}$.

If $c = (c_1, c_2)$ is an encryption of m, then $m' = (m \cdot h^y) \cdot (g^y)^{-x} = m \cdot g^{xy} \cdot g^{-xy} = m$. This ensures the completeness of the scheme.

Homomorphic Encryption

A public-key cryptosystem can have a homomorphic property.

Definition 37 (Homomorphic Encryption)**.** Let $\mathcal{M}$ be the message space, and $\mathcal{C}$ be the cipher space. Let $*$ be a binary operation over $\mathcal{M}$ and $\odot$ a binary operation over $\mathcal{C}$.

A public-key cryptosystem $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$ is homomorphic from $*$ to $\odot$ if, for all messages $m_1, m_2 \in \mathcal{M}$,

$$\mathsf{Dec}_{sk}\left(\mathsf{Enc}_{pk}(m_1) \odot \mathsf{Enc}_{pk}(m_2)\right) = m_1 * m_2.$$

In other words, it is possible to do operations on the cleartext messages with their ciphers, without knowledge of the secret key.

The El Gamal cryptosystem has this property, for the multiplication from the message to the cipher domains.

Symmetric Cryptography

Symmetric cryptography pre-empted public-key cryptography, and denotes the encryption and decryption methods that rely on a secret key for encryption and decryption. It essentially comes in three variants: block ciphers, stream ciphers and hash functions.

Block Ciphers convert blocs of bits into an encrypted form. The arch-bloc cipher widely used is the Advanced Encryption Standard (AES) [54], selected by the US National Institute of Standards and Technology (NIST).

Stream Ciphers operate on bit strings of indefinite length and encrypt them gradually, by combining them with a pseudo-random bit string.

Hash Functions take bit strings of arbitrary length and outputs a fixed-size bit vector. Cryptographic hash functions are designed to resist to first- and second-preimage (finding a bit string that gives a given hash value) and to collisions (finding two bit strings with the same hash). The Secure Hash Algorithms (SHA) are the algorithms selected by the NIST; the SHA-3 competition is currently underway.

C.2 Security Considerations

Security is considered as the probability of success of an attacker against a system. We want this probability to be negligible.

Definition 38. A function f is said to be *negligible* if for all non-constant polynomial P, and for all sufficiently large k, we have $f(k) < \frac{1}{|P(k)|}$.

When the security parameter of a scheme is ℓ, a probability is negligible if it is negligible in ℓ.

Security Models and Proofs

The security of a cryptographic protocol is proved by showing that for any probabilistic polynomial-time algorithm, the probability to break the scheme is negligible. Different security notions arise depending on what we allow the algorithm to do. As an example, we state the notion of indistinguishability for a public-key cryptosystem:

Condition 17. A public-key cryptosystem is said to be *indistinguishable* if, for all polynomial-time probabilistic algorithm $\mathcal{A}$, the probability of success of A is negligible in the following experiment:

$$\begin{array}{l} \mathsf{Exp}_{\mathcal{A}}^{\mathrm{Ind}} \\ \left| \begin{array}{llcl} 1. & (pk, sk) & \leftarrow & \mathsf{Setup}(1^l) \\ 2. & \{m_0, m_1\} & \leftarrow & \mathcal{A}(pk) \\ 3. & e & \overset{R}{\leftarrow} & \{0,1\} \\ 4. & e' \in \{0,1\} & \leftarrow & \mathcal{A}(\mathsf{Enc}_{pk}(m_e)) \end{array} \right. \end{array}$$

where the success of $\mathcal{A}$ is defined as $|\Pr[e = e'] - \frac{1}{2}|$.

In a first step (1.) the parameters of the system are generated, and $\mathcal{A}$ is given the public key. Based on this public key, $\mathcal{A}$ chooses two messages m_0 and m_1 on which $\mathcal{A}$ believes to have an advantage. One of them m_e is randomly chosen among the two messages, encrypted, and transmitted to $\mathcal{A}$. If the adversary guesses which one it was with a significant probability then $\mathcal{A}$ is successful.

The experiment of indistinguishability comes in different versions, depending on the possibilities of adversary. If $\mathcal{A}$ has access to an encryption oracle before selecting m_0 and m_1, then the condition becomes *Indistinguishability for Chosen Plaintext Attack*, or IND-CPA. If $\mathcal{A}$ has access to a decryption oracle, the condition becomes *Indistinguishability for Chosen Plaintext Attack*, or IND-CCA. The models also differs if $\mathcal{A}$ can make oracle queries before making his choice e' or not.

To prove a security property on a cryptosystem, it suffices to prove that if an adversary is able to break a security property, then he is also able to resolve a "hard problem" [164] as defined hereafter. The reduction of the property to the problem must be in polynomial time.

Hard Problems

Security proofs are based on intractability problems. These are problems that were asked but not resolved, for which no efficient algorithm is known, and that are believed to be difficult.

As an example, let us state the Computational and Decisional Diffie-Hellman problems:

Definition 39 (Diffie-Hellman triple)**.** Let G be a cyclic group generated by $g \in G$.

A triple $(X, Y, Z) \in G^3$ is a Diffie-Hellman triple if there exist $x, y \in \mathbb{N}$ such as $X = g^x$, $Y = g^y$ and $Z = g^{xy}$

Problem 6 (Computational Diffie-Hellman Problem)**.** Let G be a cyclic group generated by $g \in G$.

On input $(X, Y) \in G^2$, generate $Z \in G$ such that (X, Y, Z) is a Diffie-Hellman triple.

Problem 7 (Decisional Diffie-Hellman Problem)**.** Let G be a cyclic group generated by $g \in G$. Let $H_{DH} = \{(g^x, g^y, g^{xy}), x, y \in \mathbb{N}\}$ be the set of all Diffie-Hellman triple, and $H_3 = G^3$.

On input (X, Y, Z) randomly taken from H_{DH} or from H_3 with the same probability, output 1 if (X, Y, Z) is a Diffie-Hellman triple, and 0 otherwise.

The Computational Diffie-Hellman (CDH) assumption states that there is no probabilistic polynomial-time algorithm that solves problem 6 with non-negligible probability. The Decisional Diffie-Hellman assumption, DDH, states that there is no polynomial-time algorithm that correctly solves problem 7 with probability significantly different than $\frac{1}{2}$.

Note that if there is an algorithm that solves the CDH problem, then it also trivially solves the DDH problem.

The Decisional Diffie-Hellman assumption leads to a classical security property:

Proposition C.1. *The El Gamal cryptosystem is IND-CPA under the Decisional Diffie Hellman assumption.*

This example's goal was to detail the formalism used in the document, especially in part 3.

C.3 Advanced Primitives

In addition to encryption and decryption, one of cryptography's contribution is to propose models and constructions for protocols that have specific requirements.

We here give more information on the Private Information Retrieval and Storage protocols.

Private Information Retrieval Protocols

A primitive that enables privacy-ensuring queries to databases is Private Information Retrieval protocol (PIR) [51]. Its goal is to retrieve a specific information from a remote server in such a way that he does not know which data was sent. This is done through a method $\mathsf{Query}^{PIR}_{\mathcal{Y},\mathsf{S}}(a)$, that allows $\mathcal{Y}$ to recover the element stored at index a in S by running the PIR protocol.

Suppose a database contains M bits $X = x_1, ..., x_M$. To be secure, the protocol should satisfy the following properties [75]:

- **Soundness:** When the user and the database follow the protocol, the result of the request is exactly the requested bit.
- **User Privacy:** For all $X \in \{0,1\}^M$, for $1 \leq i, j \leq M$, for any algorithm used by the database, it cannot distinguish with a non-negligible probability the difference between the requests of index i and j.

Among the known constructions of computational secure PIR, block-based PIR – *i.e.* working on block of bits – allows to efficiently reduce the cost. The best performances are from Gentry and Ramzan [74] and Lipmaa [113] with a communication complexity polynomial in the logarithm of M. Surveys of the subject are available in [73, 136].

Some PIR protocols are called Symmetric Private Information Retrieval, when they comply with the **Data Privacy** requirement [75]. This condition states that the querier cannot distinguish between a database that possesses only the information he requested, and a regular one; in other words, that the querier does not get more information than he asked for.

Private Information Storage (PIS) Protocols

PIR protocols enable to retrieve information of a database. A Private Information Storage (PIS) protocol [136] is a protocol that enables to write information in a database with properties that are similar to that of PIR. The goal is to prevent the database from knowing the content of the information that is being stored; for detailed description of such protocols, see [20, 134].

Such a protocol provides a method $\mathsf{update}(val, index)$, which takes as input an element and a database index, and puts the value val into the database entry $index$. To be secure, the protocol must also satisfy the Soundness and User Privacy properties, meaning that 1. $\mathsf{update}_{\mathsf{BF}}$ does update the database with the appropriate value, and 2. any algorithm run by the database cannot distinguish between the writing requests of (val_i, ind_i) and (val_j, ind_j).

Appendix D

Establishing a Session Key even With Active Adversaries

> The single biggest problem in communication is the illusion that it has taken place.
>
> George Bernard Shaw

In the perspective of securing the communications between contactless devices, a key step is the establishment of a common key. As the communicating channel is public in this case, this is not an easy task. This appendix presents a method, published in [26], to strengthen a very low cost solution for key agreement with a RFID device. Starting from a work which exploits the inherent noise on the communication link to establish a key by public discussion, we show how to protect this agreement against active adversaries. For that purpose, we unravel integrity (I)-codes suggested by Cagalj et al. No preliminary key distribution is required.

The amount of computation possible in RFID tags is somewhat limited, due to constraints on cost, size and power consumption of such devices. For that reason, protocols involving RFID devices must focus on the complexity of computation on the device side; which puts aside asymmetric cryptography. Under this constraint, even symmetric cryptography settings must be thought thoroughly.

[49] uses public discussion over a noisy channel for two wireless devices to agree on a key, and shows how to realize such a protocol with low-cost tags. An eavesdropper listening to such a protocol would not gain information on the key. As a natural extension to that work, we show how to shield such a protocol in order to thwart active adversaries. The additional tools required for this additional protection are reduced to a minimal complexity.

In order to formally introduce the essential notions referred to hereafter, Section D.1 describes the channels that we use. Section D.2 explains how

Key Agreement through Public Discussion works. Section D.3 details (I)-codes, a tool that enables us to protect the Key Agreement against active adversaries. Finally, Section D.4 presents our protocol for Key Agreement through presence.

D.1 A Description of the Devices, the Channel, and the Problematic

As it is often the case in cryptographic protocols, two entities Alice (**A**) and Bob (**B**) wish to communicate securely over some channel, while an adversary Eve (**E**) wants to counter their objectives, by either preventing the establishment of a key, or by discovering the key so that the communication is no longer confidential.

We focus on wireless devices. This means that they communicate using radio frequency; a direct consequence is that all messages sent by these devices are public. Moreover, there is noise over the channel. This noise can be caused by

1. physical causes such as interferences, Doppler effect, *etc.*
2. the emission of other wireless devices, that can be genuinely communicating over the same frequency, or can willingly emit in order to alter the communication.

The presence of noise over the channel leads us to the use of Error Correcting Codes (ECC) (that enable to reduce the noise). In other terms, we have two formal channels over which the devices are able to communicate.

1. A noisy channel C_p that inherently induces errors in the transmitted messages. We here suppose that p_{AB} is a non-null error probability describing a Binary Symmetric Channel (BSC) between **A** and **B**. Moreover, we also suppose that the transmission from **A** to **E** is done through a BSC of parameter p_{AE} which can be different than p_{AB}. (see Fig. D.1).
2. A noiseless channel C_0 obtained by correcting errors over C_p.

Both channels are public, *i.e.* **E** can listen to the channel, send some messages, and even alter sent messages by adding noise.

A and **B** want to establish a common key, that would be unknown by **E**. Our constraints are for **A** and **B** to be low-cost devices, which means that no sophisticated computation is allowed, and that we aim at very few logical gates to implement the protocol. As we prove in Section D.4, we do this by constructing a noiseless channel that detects intrusion of an active adversary, in other words, a “shielded” noiseless channel.

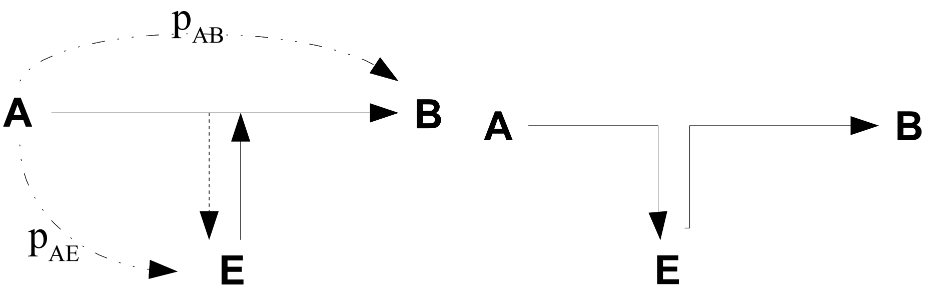


Figure D.1: The noisy channels $C_{p_{AB}}$, $C_{p_{AE}}$, and the noiseless channel C_0.

D.2 Previous Results on Key Agreement

The classical approach to key agreement by public discussion over a noisy channel was explored by [49] to apply it on low-cost devices such as RFID. This approach follows the three steps of **Advantage Distillation** [72], **Information Reconciliation** [24] and **Privacy Amplification** [44]. We recall in a few lines the main ideas behind these steps.

Advantage Distillation

A and **B** first exchange noisy data over the channel C_p (for example, **A** sends N_0 bits to **B**, and **B** receives a noisy version of those bits). Then, by public discussion over C_0, **A** and **B** select $N_1 < N_0$ bits out of the N_0 bits that were first exchanged, in such a way that the average error between the N_1-long bit string owned by **A** and the one owned by **B** is strictly less than p.

Advantage Distillation is designed in such a way that the error probability of the channel from **A** to **B** decreases more quickly than the error probability of the channel from **A** to **E** (and from **B** to **E**). A notorious example of Advantage Distillation protocol is the Bit Pair Iteration protocol; **A** and **B** send over C_0 the parity of each pair of bits of the data they own. When the parity is the same, they retain the first bit; in the other case, they discard the whole pair.

The distillation is made several times until the information sent is likely to have been sent from **A** to **B** through a BSC channel C_ϵ with ϵ small enough, and the information that **E** gets was sent through a channel C_λ with $\epsilon < \lambda$. After k iterations, **A** and **B** share N_k bits with error probability ϵ.

Information Reconciliation

After the step of Advantage Distillation, the bit strings that **A** and **B** own still differ. Information Reconciliation aims at correcting these errors by public discussion over C_0. [49] shows how to modify the Information Reconciliation protocol Cascade [24] to reduce its hardware implementation to fit into resource-constrained environment. In a nutshell, the Cascade protocol requires **A** and **B** to send the parity of blocs of data of increasing size, in such a way that they can correct the few errors remaining with high probability.

Privacy Amplification

A and **B** now agree on a bit string S of length N_k with very high probability. The aim of Privacy Amplification is to derive a shorter key out of the shared data, on which Eve has no information. For that purpose, **A** and **B** agree on a universal hash function from a predefined family of functions, and compute the hash of the bit string. This gives a shorter key K which is the result of the Key Agreement protocol; [44] proves that **E** finally does not get any information on K.

For practical purposes, the Universal Hash Functions defined in [99] are suited for low hardware requirements.

Summary

These three steps are well known, and enable Key Agreement over a noisy and public channel. However, such a construction is only valid for a passive adversary, *i.e.* when Eve just listens to messages that were sent over the air. In the era of wireless communication, anyone can temper with the data that was sent over a wireless channel, which is the base of packet injection attacks.

The next sections describes our contribution: how to adapt this scheme so that the key establishment protocol described above is resistant to active attacks?

D.3 Integrity (I)-codes

In a wireless environment, there is no existing mechanism that prevents an adversary to jam all communication between two devices. Indeed, a powerful white noise can make a Signal-to-Noise Ratio as low as possible. Thus, our goal is not to ensure that no one jams the communication, but to prevent an active adversary to obtain a significant advantage against one of the devices. The sole detection of an attack is thus enough in our model.

We therefore describe a protection system made to detect all intrusion attempts in the communications between **A** and **B**, called Integrity Code. These were introduced in [45, 46], and make use of physical means to protect the communication.

Integrity (I)-code bits are transmitted in such a way that an adversary can hardly change a bit “1” into a “0”. Moreover, information is coded in order to detect the remaining possible bit flipping: from “0” to “1”. Putting these 2 protections together, an adversary cannot modify a message without having a high probability of being detected.

Remark 53. Our use of integrity (I)-codes enables us to fulfil the non-Simulatability Condition introduced in [120].

Physical Transmission

The bits are transmitted using the **On-off keying** technique (OOK). Signal is divided in time-periods of length T. Each bit "1" is transmitted as a non-null signal of duration T. Each bit "0" corresponds to the absence of signal during the same amount of time T.

As the elimination of a non-null electromagnetic signal is very costly, this satisfies the first constraint: preventing the flipping from a "1" to a "0".

Assumption 1. It is impossible for an adversary to alter the transmission of a binary "1" using OOK.

Unidirectional Coding

In order to detect the flipping from a "0" to a "1", information is coded using a **Unidirectional Error-Detecting Code** [14]:

Definition 40. A Unidirectional Error-Detecting Code is a triple (S, C, α), satisfying the following conditions:

1. S is a finite set of possible source states,
2. C is a finite set of binary codewords,
3. α is a source encoding rule $\alpha : S \to C$, such that:
   - α is an injective function,
   - C respects the "non-inclusive supports" property, *i.e.* it is not possible to convert codeword $c \in C$ to another codeword $c' \in C$, such that $c' \neq c$, without switching at least one bit 1 of c to bit 0.

The "non-inclusive supports" property can be restated this way: if $c \in C$ is a binary codeword of length n, and $\mathsf{supp}(c) = \{i \in \{1, \dots n\} | c_i = 1\}$ is the support of c, then $\forall c, c' \in C$, the supports of c and c' are not included one into the other, *i.e.* $\mathsf{supp}(c) \not\subset \mathsf{supp}(c')$ and $\mathsf{supp}(c') \not\subset \mathsf{supp}(c)$.

The Manchester coding which encodes bit "1" into 10 and bit "0" into 01 is a very simple example of unidirectional error-detecting code. When combined with On-Off Keying, its error-detection rule simply consists in verifying that a codeword contains an equal number of symbols "0" and "1".

More generally, any binary immutable WOM-code (codes dedicated to Write-Once Memory) permits unidirectional coding. A Write-Once Memory is an array of bits such that once a bit was set to "1" it can never be unset again; immutable WOM-codes prevent the rewriting of a message on a Write-Once Memory. To improve the Manchester code, which has a rate of $\frac{1}{2}$, and following [76], we suggest the use of the Berger code. To encode a word x of length l, we add $\lceil \log l \rceil$ bits of redundancy in the following way: the binary weight $w(x) = \sum_{i=1}^{l} x_i$ is computed, and represented in its binary version $w_1, \ldots, w_{\lceil \log l \rceil}$. The coded version of x is the concatenation of x with $\overline{w_1, \ldots, w_{\lceil \log l \rceil}}$, *i.e.* $\left(x_1, \ldots, x_l, \overline{w_1}, \ldots, \overline{w_{\lceil \log l \rceil}}\right)$ [1]. The Berger code works because if $\mathsf{supp}(x) \subset \mathsf{supp}(x')$, then $w(x) \leq w(x')$, and $\mathsf{supp}\left(\overline{w_1}, \ldots, \overline{w_{\lceil \log l \rceil}}\right) \not\subset \mathsf{supp}\left(\overline{w'_1}, \ldots, \overline{w'_{\lceil \log l \rceil}}\right)$.

Remark 54. The idea of unidirectional coding was introduced by [119] in the same context.

D.4 Key Agreement Through Presence

The Model

Here is the description of the model for which we design the protocol. It is based on the facts described previously: communication between wireless devices is public, any adversary can make the communication unreadable, it is not possible to make expensive computation with cheap devices. Therefore, the following hypotheses are made:

- **A** is a low-cost device with limited computation and memory possibilities;
- **B** is a wireless sensor *i.e.* a communicating device that has reasonable computing hardware;
- The two devices **A** and **B** are *in presence*, which means that they are communicating with each other, and not with a third party **E**;
- **E** can hear everything that **A** and **B** send;
- **E** is able to emit at the same time an electromagnetic signal.

This last item is the main difference between the existing protocols and the following: we here consider *active adversaries*.

Definition 41. Let C be a channel between **A** and **B**, and **E** be an adversary such that:

- Transmission of a message $s = (x_1, \ldots, x_n) \in \{0, 1\}^n$ from **A** to **B** without interference of **E** is noiseless.

[1] The notation $\overline{a}$ is the binary negation of a.

- Transmission of s from **A** to **B** with intervention of **E** leads to the reception of $\Phi_E(s) = s' = (x'_1, \ldots, x'_n)$.
- A failed transmission leads to a state $\perp$ for **A** and **B**.

C is ϵ-resistant against an active adversary if except with probability less than ϵ, $\forall s \in \{0,1\}^n, s = \Phi_E(s)$ or **A** and **B** are in the state $\perp$.

Such a channel is such that, after a transmission, either **A** and **B** possess the same message s, or **A** and **B** know that the transmission was a failure.

Rewriting the Three Steps

As we mentioned it in Section D.1, there are two channels for **A** and **B** to communicate. The first one is C_p, the second C_0.

1. The messages that are sent over the channel C_0 are error-less thanks to error correction techniques. To eliminate an active adversary's chances of tempering with this channel, we add a fourth step called **Integrity Verification** after the three enumerated in Section D.2, described hereafter.

2. In the classical key agreement protocol, the channel C_p between **A** and **B** (resp. **A** and **E**) is usually modeled as a BSC channel with error probability p_{AB} (resp. p_{AE}). If the adversary is active during the first phase, then the effect is an increase of p_{AB} without a change on p_{AE}. However, the Advantage Distillation step finally leads to a new error probability p'_{AB} that is lower than p_{AE} independently of the initial situation. Therefore, thanks to the final Integrity Verification, an active adversary cannot gain an advantage at this step.

Validating the Agreement

The final verification step permits to ensure that the key agreement protocol was not perturbed by an active adversary. For that, the idea is to check that all the messages sent and received by **A** and **B** were the same, using a protection technique on the verification message.

Note $\mathcal{M}$ the set of all the messages that were emitted by both devices, in their order of apparition. We expect **B** to continuously save $\mathcal{M}$. At the end of the protocol, **A** will send to the wireless sensor **B** the identifier of a function h taken from a family of hash functions, together with $\alpha\left(h(\mathcal{M})\right)$ where α is the source encoding rule defined in Definition 40.

To reduce memory usage, **A** can compute $h(\mathcal{M})$ in an incremental way, by $x_{n+1} = h(x_n || m_{n+1})$ with m_i the i-th message transmitted over the channel, and x_i the hash of the i first elements. $||$ is the concatenation operator.

We therefore suggest the following order for the global scheme, which is illustrated in Fig. D.2.

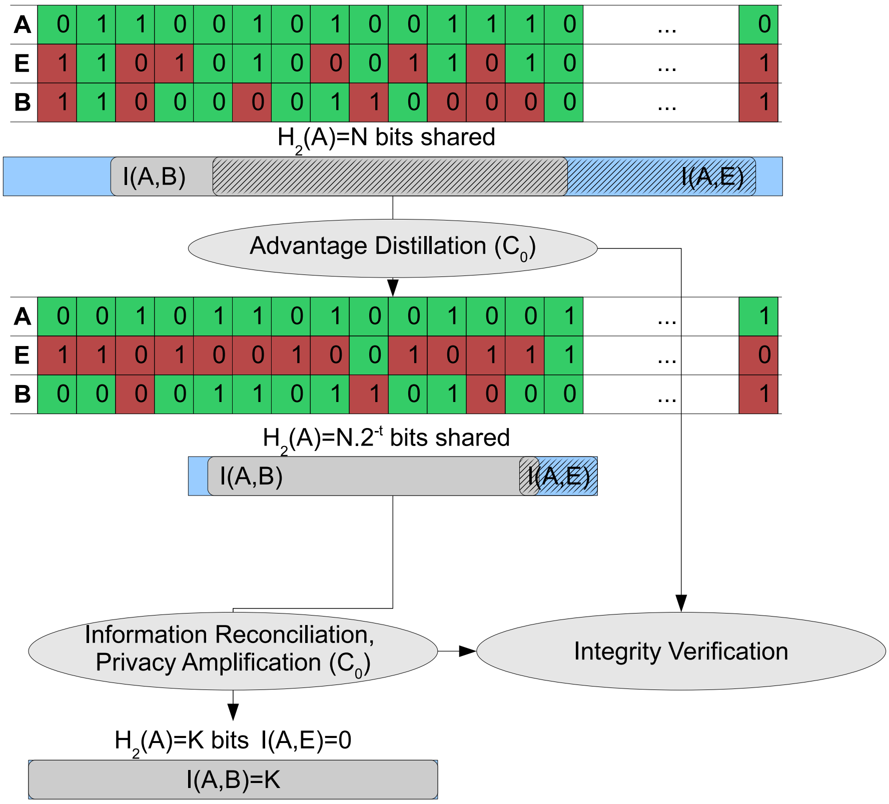
1. **A** chooses the hash function h from a family of hash functions;

2. **B** sends to **A** a bit stream using $C_{p_{AB}}$;

3. **A** and **B** proceed to Advantage Distillation, Information Reconciliation, and Privacy Amplification;

4. **A** sends to **B** the identifier of h;

5. **A** and **B** do the Integrity Verification step: **B** sends to **A** the message $\alpha\left(h(\mathcal{M})\right)$ where $\mathcal{M}$ designates all the messages that were sent over C_0, using On-Off Keying (over C_0).



A	0	1	1	0	0	1	0	1	0	0	1	1	1	0	...	0
E	1	1	0	1	0	1	0	0	0	1	1	0	1	0	...	1
B	1	1	0	0	0	0	0	1	1	0	0	0	0	0	...	1

H$_2$(A)=N bits shared
I(A,B)
I(A,E)
Advantage Distillation (C$_0$)

A	0	0	1	0	1	1	0	1	0	0	1	0	0	1	...	1
E	1	1	0	1	0	0	1	0	0	1	0	1	1	1	...	0
B	0	0	0	0	1	1	0	1	1	0	1	0	0	0	...	1

H$_2$(A)=N.2^{-t} bits shared
I(A,B)
I(A,E)
Information Reconciliation,
Privacy Amplification (C$_0$)
Integrity Verification
H$_2$(A)=K bits I(A,E)=0
I(A,B)=K

Figure D.2: The global scheme, illustrated

The Noiseless Shielded Channel

We here deliver the statement made in Section D.1: with simple tools, to achieve a channel that is noiseless and integrity resistant against the intrusion of an active adversary.

The channel designed so far complies with Definition 41, as this is expressed in the following formalization: Let **A** and **B** be a sender and a receiver; let $n, t_1, t_2 \in \mathbb{N}$ with $n \geq t_1$ and $t_2 \geq t_1$, $h : \{0,1\}^n \to \{0,1\}^{t_1}$ be a hash function, and $\alpha : \{0,1\}^{t_1} \to \{0,1\}^{t_2}$ a source encoding rule following Definition 40.

A emits a message $s \in \{0,1\}^n$ to **B** using On-Off Keying, by sending $S = s||\alpha(h(s))$. At the reception of $S' = s'_1||s'_2$ with $|s'_1| = n$, **B** checks that $s'_2 = \alpha(h(s'_1))$. If this test fails, then **B** emits a standard message expressing failure. If not, **B** uses the now shared key to validate the agreement.

Proposition D.1. *The scheme described in the previous paragraph gives a channel C that is ϵ-resistant against an active adversary, where*

$$\epsilon = \Pr_{x,x'}\left[h(x) = h(x')\right]$$

is the collision probability of h.

> **Proof** Two cases need to be considered : either **E** does not intervene, or **E** tries to alter the communications. In the first case, we obviously have $S = S'$, which also gives $s = s'_1$ which was the desired result.
>
> In the second case, note that, thanks to OOK (see Assumption 1), the only action **E** can do is to change a "0" that was sent into a "1".
>
> - If **E** alters $\alpha(h(s))$ into s'_2, using the unidirectional property of α, the equality $s'_2 = \alpha(h(s'_1))$ is never achieved.
> - If **E** alters s into s', but not $\alpha(h(s))$ then **E** wins only if $h(s) = h(s')$, *i.e.* with probability less than ϵ.
>
> This shows that the alteration of a message by **E** is detected with probability greater than $1 - \epsilon$. Therefore the channel is ϵ-resistant against an active adversary. □

In our application, an active **E** can alter the agreement on the hash function h. If this happens, then **A** owns a function h_A and **B**, h_B. With this kind of advantage, **E** must nonetheless change $\mathcal{M}_A, \mathcal{M}_B$ into $\mathcal{M}'_A, \mathcal{M}'_B$, with the properties $h_A(\mathcal{M}_A) = h_B(\mathcal{M}'_A)$ and $h_A(\mathcal{M}'_B) = h_B(\mathcal{M}_B)$. Moreover, to successfully interfere in the communication, an active **E** must change "on the fly" messages that are sent by **A** and **B** such that the final hashes collide, with no knowledge of the future messages to be sent, and with the constraint $\mathsf{supp}(x) \subset \mathsf{supp}(x')$, *i.e.* **E** can only change "0" into "1". This makes her task even harder.

Remark 55. Our new approach does not resist to an active adversary issuing a low-energy DoS attack to invalidate all key exchanges. As mentioned earlier, our goal is not to prevent DoS attacks.

Conclusion

This appendix describes a method to establish a key with a low cost wireless device. Starting from the classical key agreement methods, we provide the tools to achieve the integrity mechanisms necessary in order to cope with active adversaries. Using integrity (I)-codes - a modulation method that prevents to switch from a “1” to a “0”, combined with unidirectional coding, we add a fourth step that detects intrusion in the communication.

We finally focus on the computation cost so that devices with very few logical gates can instantiate this protocol. Indeed, the device needs only to implement a few functions for the protocol to work:

- A parity evaluator – for the Advantage Distillation and Information Reconciliation steps,
- A universal hash function, for Privacy Amplification,
- A unidirectional coding scheme, for Integrity Verification,
- A binary comparator.

The universal hash function is here the most gate-consuming element, and can be designed in roughly 640 gates following [191]. The universal coding scheme, that uses a Berger code, only requires to compute a binary weight, and a logical negation. For key length of about 64 bits, this can be done in about 320 gates. Finally, the overall complexity of such a device is of the order of 1000 logical gates.

This makes way for the production of large amounts of low-cost tags allowing secure communication.

Résumé

On parle d'identification lorsqu'une personne ou un objet communicant présente un élément qui permet sa reconnaissance automatique. Ce mode s'oppose traditionnellement à l'authentification, dans laquelle on prouve une identité annoncée. Nous nous intéressons ici à l'identification biométrique d'une part, et à l'identification d'objets communicants sans-fil d'autre part. Les questions de la sécurité et du respect de la vie privée sont posées. Il y a sécurité si on peut s'assurer de la certitude que l'identification produit le bon résultat, et la vie privée est respectée si une personne extérieure au système ne peut pas déduire d'information à partir d'éléments publics.

Nous montrons que dans le cas biométrique, le maillon le plus sensible du système se situe au niveau du stockage des données, alors que dans le cas de communications sans-fil, c'est le contenu des messages qui doit être protégé. Nous proposons plusieurs protocoles d'identification biométrique qui respectent la vie privée des utilisateurs; ces protocoles utilisent un certain nombre de primitives cryptographiques.

Nous montrons par ailleurs comment l'utilisation de codes d'identification permet de mettre en oeuvre des protocoles d'interrogation d'objets communicants.

Abstract

The term 'identification' refers to a situation where a person, or a communicating device, provides an element that ensures its automatic recognition. This differs from authentication in which the claimed identity is proved with credentials. We take interest in both the identification of people and devices; the former goes through biometrics, and we study the particular case where devices communicate through electromagnetic waves. These situations raise the issues of security and privacy. Security is a confidence level in the outcome of the identification; privacy ensures that an eavesdropper cannot infer information from public elements.

We show that in order to design private biometric identification protocols, special care must be taken for the storage of the biometric data. We describe several such protocols that are based on cryptographic primitives.

We also show how to use identification codes to design a protocol for private interrogation of low-cost wireless devices, both private and secure.