




Privacy - Preserving Processing of Biometric Templates by Homomorphic Encryption



Pierluigi Failla

Ph.D Thesis in Information Engineering
University of Siena



UNIVERSITÀ DEGLI STUDI DI SIENA

FACOLTÀ DI INGEGNERIA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE



**Privacy–Preserving Processing of
Biometric Templates by
Homomorphic Encryption**

Pierluigi Failla

Ph.D Thesis in Information Engineering

Supervisor

Professor Mauro Barni

Examination Committee

Professor Patrizio Campisi

Professor Fabio Massacci

Associate Professor Sandro Bartolini

Thesis reviewers

Professor Stefan Katzenbeisser

Professor Patrizio Campisi

SIENA, JANUARY 31, 2011

Contents

Glossary	vii
Acknowledgements	xi
1 Introduction	3
1.1 Motivations	3
1.2 Processing Encrypted Signals	7
1.3 Contribution and Outline	10
2 Cryptographic Tools for Privacy Preserving Protocols	17
2.1 Homomorphic Cryptosystems	17
2.2 Security	21
2.2.1 Cryptosystem Security	21
2.2.2 Honest but Curious Model	22
2.3 Complexity	23
2.4 Paillier Cryptosystem	24
2.5 Damgaard – Jurik cryptosystem	29
2.6 Blinding	30
2.7 Application to Processing Encrypted Data	32

3	Biometric Systems	37
3.1	Introduction to Biometric Systems	37
3.2	Biometrics	43
3.2.1	Fingerprint	43
3.2.2	Face	45
3.2.3	Iris	47
3.2.4	Voice	48
3.2.5	DNA	50
3.3	Protecting Biometric Data	51
3.3.1	Generalities	51
3.3.2	Privacy Protection of Biometric Data	55
4	Privacy Preserving FingerCode	59
4.1	Introduction	59
4.2	FingerCode-Based Authentication	60
4.2.1	FingerCode Construction	60
4.2.2	Matching	68
4.3	The Addressed Scenario	69
4.3.1	Security Analysis	72
4.4	Parameters and Model	73
4.5	Basic Building Blocks	74
4.5.1	The sub-protocol BitMin	74
4.6	The FingerCode Matching Protocol	79
4.6.1	Variant for Simple Authentication.	83
4.6.2	Variant for Authentication with Identity Confirmation.	83
4.7	Security	84
4.8	Complexities	86
4.9	Real World Implementation	86
4.10	Summary	92

5 Privacy Preserving Sketch	95
5.1 Introduction	95
5.2 The Fuzzy Commitment Scheme	97
5.3 A Possible Scenario	100
5.3.1 Security Analysis	102
5.4 Basic Building Blocks	103
5.4.1 The sub-protocol XOR	103
5.4.2 The sub protocol eSearch	106
5.5 The Protocol	109
5.6 Security	112
5.7 Complexities	114
5.8 Avoiding the Leakage of Information	114
5.8.1 Complexities	116
5.9 Summary	118
6 Final Remarks	121
6.1 Summary and Contributions	121
6.2 Track for Future Works	123
Bibliography	127
Index	141
Publications List	145
Curriculum Vitae	149

List of Figures

2.1	Blind Computation with Encrypted Data.	31
2.2	EncMul Sub-Protocol.	33
3.1	Pattern Recognition System.	42
3.2	Fingerprint Minutiae.	44
3.3	Face Geometry Measurement.	46
3.4	Eigenface Samples.	47
3.5	IrisCode.	48
3.6	DNA Fingerprint.	51
4.1	FingerCode Authentication System.	61
4.2	Core Detection.	62
4.3	Sector Division.	63
4.4	Fingerprints frequency and orientation.	64
4.5	Gabor filters.	65
4.6	Fingerprint after Gabor filtering.	66
4.7	Various orientations of the Gabor filter.	68
4.8	Finger Scanners.	70
4.9	The Protocol BitMin.	75

4.10	The sub-protocol DGK.	77
4.11	Privacy Preserving FingerCode Matching.	82
4.12	Privacy Preserving FingerCode - Identity Confirmation.	84
4.13	Examples of test images.	87
4.14	Results obtained.	88
4.15	Equal error rate of the different configurations.	90
4.16	ROC Curves.	92
5.1	Sketch Enrollment.	98
5.2	Sketch Match.	99
5.3	Sub protocol XOR with $\llbracket x \rrbracket$ and $\llbracket y \rrbracket$	105
5.4	eSearch.	107
5.5	eSketch – Enrollment.	110
5.6	eSketch – Matching.	112
5.7	eSearch Variant.	115
5.8	eSketch Variant – Enrollment.	116
5.9	eSketch Variant – Matching.	117

List of Tables

2.1	Homomorphic Properties.	20
2.2	NIST Recommendation.	27
3.1	Biometries Comparison - Intrinsic.	40
3.2	Biometries Comparison - Implementation.	41
4.1	Computational Complexities – BitMin sub-protocol.	76
4.2	Computational Complexities – DGK sub-protocol.	79
4.3	Computational Complexities – Privacy Preserving FingerCode.	86
4.4	Computational Complexities – Privacy Preserving FingerCode.	89
4.5	Performance of the proposed method.	91
4.6	Required time for the identification with FingerCode.	91
5.1	Computational Complexities – XOR with $\llbracket x \rrbracket$ and y	104
5.2	Computational Complexities – XOR with $\llbracket x \rrbracket$ and $\llbracket y \rrbracket$	105
5.3	Computational Complexities – eSearch.	109
5.4	Protocol eSketch Complexities.	115
5.5	Variant Protocol eSketch Complexities.	118

Glossary

CEO	Chief Executive Officer., 3
DFT	Discrete Fourier Transform., 9
DGK	Is an homomorphic comparison protocol (see [DGK07] for more details)., 72
EC	Elliptic Curve., 73
ECC	Error Correcting Code., 93
ECG	ElectroCardioGram., 9
EER	Equal Error Rate., 87
FFT	Fast Fourier Transform., 9
FMR	False Match Rate., 84
FNMR	False Non-Match Rate., 84
GCD	Greatest Common Divisor., 23
IND-CPA	INDistinguishability under Chosen – Plain-text Attack or Semantic Security., 19

LBP	Linear Breanching Program., 31
LCM	Least Common Multiple., 23
PIR	Private Information Retrieval., 96
PIS	Private Information Storage., 97
PrK	Private Key., 16
PuK	Public Key., 16
ROC	Receiver Operating Characteristic., 87

Acknowledgements

Acknowledgements

In particular, I wish to express my gratitude to my supervisor, **Professor Mauro Barni** for his continued encouragement and invaluable suggestions during this work. I would also like to include my gratitude to **Professor Nasir Memon** who supervised my stay at Polytechnique Institute of New York University during my visiting in USA.

I am very grateful to examination committee and in particular to **Professor Stefan Ktzenbeisser** and **Professor Patrizio Campisi** for reviewing this thesis and their insightful comments and suggestions.

[REDACTED]

Acknowledgements

[REDACTED]

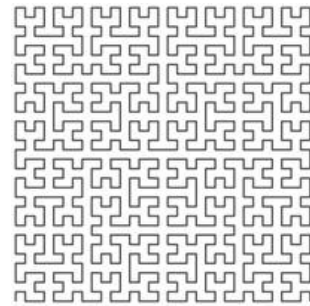
I'd like to thanks all the people who believed in me...

AND I THANK YOU FOR BRINGING ME HERE
FOR SHOWING ME HOME
FOR SINGING THESE TEARS
FINALLY I'VE FOUND THAT I BELONG HERE
Martin Gore - Depeche Mode

Grosseto,
January 31, 2011

Chapter 1

Introduction



*If I were to awaken after having slept for a thousand years,
my first question would be:
Has the Riemann hypothesis been proven?*
(David Hilbert)

1.1 Motivations

Few years ago, few people could predict that the entire digital world would have started to produce data daily at the impressive rate we assist today. Eric Schmidt, the CEO of Google, estimated at 2010 the Internet size at roughly 5 million Terabytes¹ of data and with a constant expansion of 100 terabytes per month. This seems to be crazy especially if we think that also Internet follows the Moore's Law, for instance in [ZZY⁺08] researchers claim

¹5 Exabytes = 5000000 Terabytes = $5 \cdot 10^{18}$ Byte.

that the Internet doubles in size every 5.32 years. Our world is becoming strongly interconnected and by the Internet we are able to share everything. Think about social networks (i.e. Facebook, LinkedIn, MySpace, Twitter) whereby people share thoughts, events, photos and videos with friends. It is clear that behind this massive amount of data there are several issues related to the security of the data itself. Potentially privacy sensitive data such as our age, health, preferences, locations, politics and religious views are being stored in computers that we do not own². Moreover the data is generally transferred to third parties in plain format (think about uploading photos or videos on Facebook): people believe in the good will of third parties to behave and handle their data in accordance to laws but also according to their own privacy policies that very often people do not know or do not care about. A recent example is Facebook [LBW08] that changed its privacy policy and kept data stored even after a user had quit the service. Here is an excerpt from the Facebook contract with users: *You hereby grant Facebook an irrevocable, perpetual, non-exclusive, transferable, fully paid, worldwide license (with the right to sublicense) to (a) use, copy, publish, stream, store, retain, publicly perform or display, transmit, scan, reformat, modify, edit, frame, translate, excerpt, adapt, create derivative works and distribute (through multiple tiers), any User Content you (i) Post on or in connection with the Facebook service or the promotion thereof subject only to your privacy settings or (ii) enable a user to Post, including by offering a Share Link on your website and (b) to use your name, likeness and image for any purpose, including commercial or advertising, each of (a) and (b) on or in connection with the Facebook service or the promotion thereof. You represent and warrant that you have all rights and permissions to grant the foregoing licenses.* It is clear that these new

²In a recent interview to the *Wall Street Journal* (August 19th 2010) Eric Schimdt affirms that to escape their virtually frivolous past, many users of sites such as Facebook will want to change their name because of the potential future employers who will look through their past over the Internet.

platforms and networks are extremely vulnerable to private data disclosures³. This is due to the massively distributed data storage and the resulting increase in system and management complexities. Current ad-hoc security methodologies, combined with the sometimes shocking lack of security, will only lead to more weaknesses as the amount of data and system complexity increase during the coming years. On the other side, laws aiming at protecting private data are continuously emanated. For instance, European privacy law is clear: *“a person’s information can only be used with their prior consent⁴”*. However, legal assurance is only half of the answer. Once our private data such as date of birth, political views, preferences or other sensible information have been compromised, it is very difficult, if not impossible, to “make it private” again (as we see above in the excerpt of Facebook contract with user). For citizens, to take advantage of the ubiquity of forthcoming services, privacy and security of their data as well as its subsequent use has to be guaranteed - a priori.

There are many cases in which the constraints given by privacy and security are even more stringent. A lot of people in everyday life use airplanes to move around the world and as everyone knows following the September 11 attacks, the controls in airports have been increased. New electronic passports have been introduced for improved border controls and now they contain: 1) personal data, 2) face image and 3) the fingerprints. Each time someone takes a flight the above information is available to the staff and to the police for identity check. To be more specific, let us consider the following scenario. There are two parties, say an Intelligence Agency and a remote controller, say the security staff of an Airport. The Agency wants to trace the movements

³For instance consider that: *Facebook allows users to de-associate themselves from unwanted data, but in the case of photographs, the data remains on the server.* More details in [JS05].

⁴Excerpt from: *“Europeans must have the right to control how their personal information is used. European privacy rules are crystal clear: your information can only be used with your prior consent”* by Viviane Reding, Information Society and Media European Commissioner.

of a suspect person. To do so, it exploits some biometric information of the suspect person. In particular it tries to match the biometric sample it owns with the biometric of the people that are going to take a flight. The Agency wants to protect the identity of the suspect person (and hence his biometric) while the Airport wants to protect the privacy of the passengers. From the point of view of the client, the question is: *if I am a good guy, why should I reveal my biometric data to other parties?* At the same time flight safety must be assured, and clearly, from the point of view of the Agency, they are interested in avoiding any risk. More generally, we can affirm that the use of biometric data is becoming a common approach to handle people identities. Consider for instance that at Disney World Resort in Florida customers use the fingerprint scanning for the clients that own a multiple-days ticket to assure the not re-usability [Cam04].

When dealing with biometric data, there is usually a trade-off between the security of the system that is assumed to be protected and the privacy of the users who provide the biometrics. Often government and law enforcement agencies can access personal information to protect public safety and national security: however, abuses of personal information can cause untold harm, wasted resources, and generally lead to the detriment of society. Hence, there is a high demand for technologies that permit the use of biometric data while protecting the privacy of the data owners.

The most obvious and well-known way to secure personal data is to encrypt and store it in a (trusted) database. Such an approach works only when the owner of the data and the party in charge of processing or storing it trust each other, and the goal of the cryptographic module is to protect the data from a third party. This is not the case in many practical situations where the owner of the to-be-protected data and the party that is in charge of storing or processing it do not trust each other. Possible examples include the storage of biometric information in a central database, the processing of personal (e.g. medical) data for statistical analysis, or the analysis of people behaviors

(e.g. log files) for inspection purposes. How is it possible to trade-off between the request for privacy and the need to analyze personal information for a legitimate purpose (possibly in the interest of the data owner itself)?

An effective and elegant way to answer the above question is to process the data while they are encrypted. In the last thirty years⁵ the cryptographic community has hardly worked to build a set of tools that allow to compute with encrypted data. Though this may seem a very difficult task, some solutions have been put forward recently by relying on the use of (i) homomorphic encryption, whereby some algebraic operations are mapped into simple operations to be applied in the encrypted domain, and (ii) multi party computation, where two or more non-trusted parties cooperate to carry out a computation without revealing their own inputs. In this thesis we focus on the use of such techniques for the protection of privacy in biometric systems.

1.2 Processing Encrypted Signals

Though the possibility of processing encrypted data (mainly by means of homomorphic encryption) has been advanced more than thirty years ago [RAD78], processing encrypted signals poses some new problems due to the peculiarities of signals with respect to other classes of data more commonly encountered in the cryptographic literature, e.g. alphanumeric strings or bit sequences. The most straightforward difference is that signals are usually represented by means of real numbers (and processed by means of floating point arithmetic), while all the available cryptosystems work on integer rings. Other important differences include:

- the non-precise nature of signals, that should be contrasted with the bit-precise nature of the data cryptosystems usually deal with;
- the essential role played by the temporal or spatial structure of signals

⁵The first mention is in [RAD78] 1978 by Rivest et al.

(in many cases what really matters is the way the signal varies with time rather than the single values it assumes);

- the large size of many signals such as audio files, still images, and video sequences, that poses very critical constraints on the complexity and storage requirements.

Some recent studies spanning from digital watermarking [AKS03] through secure compression [JIP⁺04] and access to encrypted databases [BDJ04], have shown that the application of signal processing in the encrypted domain is indeed feasible. The cryptographic primitives used to process encrypted signals belong to two main categories: homomorphic encryption [Rap04] and garbled circuits [Yao82].

Homomorphic cryptosystems have the property that some elementary algebraic operations in the plain domain are mapped into elementary operations in the encrypted domain. For instance, in the Pailler cryptosystem [Pai99], an addition in the plain domain corresponds to a multiplication in the encrypted domain. Other examples of homomorphic cryptosystems include RSA [RSA78] that is multiplicatively homomorphic on product, Damgaard-Jurik generalization of Pailler scheme cryptosystem [DJ01] and Bresson et al. cryptosystem [BCP03] (that is additively homomorphic). If a homomorphic cryptosystem is used, it is possible for a party that does not possess the decryption key to perform some simple operations on the encrypted messages. For instance, by relying on the Pailler cryptosystem it is possible to apply any linear operator (with known coefficients) to an encrypted signal.

Despite its elegance and simplicity, the current state of the art in homomorphic encryption does not allow the efficient simultaneous preservation of addition and multiplication even if an innovative result has been presented by Gentry in [Gen09] where it is shown that algebraically homomorphic cryptosystems exist. By this, it is clear that homomorphic cryptosystems do not allow the application of non-linear operators, which, on the other side, are es-

essential ingredients of any non-trivial operation to be applied to the encrypted signals. To avoid the above limitation, the general approach is to use an interactive protocol whereby Alice and Bob collaborate and exchange data to securely compute a given functionality.

In the field of garbled circuits, introduced by Yao in 1982 [Yao82] and later refined in [GMW87], it is known that any function can be computed in a secure manner by implementing a boolean circuit of secure gates. With Yao's circuit approach one can implement circuits using both private-key and public-key primitives. Approaches based on symmetric primitives are several orders of magnitude faster than the asymmetric approaches. The circuit approach can be relatively efficient in different security models even if it requires to transfer from one party to the other a large amount of data which yields an increase in the communication complexity of the protocol.

Homomorphic encryption and interactive protocols provide the basis for processing and analyzing signals in the encrypted domain, however the application of these techniques to real scenarios poses a number of still unsolved challenges. For instance develop a set of elementary signal processing basic primitives. This is by itself a very challenging task, given that several problems need to be faced with including: computational complexity, difficulty of representing real numbers or implementing floating point arithmetic on integer rings, implementation of non-linear operations by minimizing the resort to interaction, etc. Some preliminary work in this direction is described in [BPB08] where the problems one encounters when trying to implement the DFT (Discrete Fourier Transform) or FFT (Fast Fourier Transform) algorithms in the encrypted domain are tackled with.

Using and extending the above examples, researchers developed many complex protocols to be applied in applications where the privacy and the security of the inputs are crucial. The proposed applications range from heuristic search in encrypted graphs [Fai10]; ElectroCardioGram (ECG) classification [BFK⁺09]; data mining [AS00]; face recognition [EFG⁺09, Erk10],

remote diagnosis [BPSW07].

1.3 Contribution and Outline

This thesis focuses on the application of secure computation techniques to biometric signals mostly based on homomorphic encryption. The use of biometric signals for security applications is by itself a very hot research field. As a matter of fact, in the last years a huge amount of research has been carried out concerning measurement and analysis of biometric traits (e.g., fingerprint, iris, face, gait, palm, voice, motion) [WJMM04], [Ash00], [JRP06b]. Particular attention has been given to generating unique identifiers to perform person verification, identification, and recognition for access control, as well as to detect suspicious behaviors for surveillance applications.

Protecting the privacy of biometric traits is also a hot topic. Security and privacy are fundamental especially when biometric data are stored in databases or transmitted in distributed information systems [PP05], [SPG⁺06]. It is commonly known that there is a trade off between the security of the systems based on biometric solutions and the privacy of the biometric data itself. In particular, the technologies behind practical privacy preserving algorithms and protocols belong to several different disciplines including signal processing, cryptography, information theory, each of which with a long standing tradition of theoretical and practical studies. At the same time, only few is known about their joint use, both at a theoretical and a practical level, the separation-paradigm being by far the most popular approach. Furthermore, most of the existing approaches while competitive at the theoretical level, have never been experimented in practice thus making it difficult to judge their viability. Despite many recent advances made in the above fields in the last years, the challenges set forth by the application of secure signal processing tools to complex signals such as biometric signals (face images, fingerprints, iris images, voice samples, etc) appear formidable. This leaves on the ground

a great number of questions about the potentiality and limits offered by the application of secure signal processing tools in biometric systems.

As a possible solution to these problems, biometric encryption, that is the encryption of a personal identifier by means of a biometric trait, has been recently proposed [DRS04]. Such techniques permit to extract a secure key from a biometric trait and allow to store a biometric template, or secure sketch [LSM06]. The problem with secure sketches, is that they leak information about the original biometric trait and can be used for tracking users in databases [STP09] [IW07]. Moreover, practical biometric authentication protocols require the use of trusted third parties to certify the authenticity of the biometric template [CS07]. The main goal of this thesis is to provide privacy preserving solutions to handle biometric samples avoiding the leakage of information that is intrinsic in the existing approaches and guaranteeing the privacy of the users.

In particular we focus on the use of homomorphic cryptosystems to develop privacy preserving protocols for person identification using encrypted biometric samples. Without loss of generality, we can say that this kind of applications can be essentially summarized as a query (the biometric sample) to a database (a set of stored biometric samples). This high level description could be useful to identify three approaches to the problem from the point of view of privacy protection:

encrypted query to a plain database: in this case we want to protect what we are searching;

plain query to an encrypted database: in this case we want to protect the data on which we are searching;

encrypted query to an encrypted database: in this case we want to protect both the query and also the data.

In this thesis we focus on privacy preserving systems that can be cat-

egorized as: (i) encrypted query to a plain database, specifically we detail the privacy preserving version of the FingerCode algorithm (systems capable to protect the client identity and the biometric sample); and (ii) encrypted query to an encrypted database, in particular we exploit the Fuzzy Commitment Scheme (systems capable to protect client identity and biometrics, but also the biometric samples in the database). In the first case we will refer to an application that works with fingerprint biometric samples, but this choice is not a constrains to use the same solution with different biometric samples. In the second case we propose a general solution not related to a specific biometric.

The contribution of this thesis can be summarized as following:

1. we propose a construction that realizes a privacy preserving protocol able to solve the problem of querying a plain database with an encrypted query, to do that:
 - we use an encrypted biometric sample to query a database of plain biometric samples;
 - we focus on fingerprint biometric and the FingerCode algorithm;
 - we provide results related to a real world implementation of the described protocol;
2. we propose a construction for a privacy preserving protocol that solves the problem of querying with an encrypted query an encrypted database. For this case:
 - we use an encrypted biometric sample to query a database of encrypted biometric samples;
 - we focus on the Fuzzy Commitment Scheme and we provide a general solution without focusing on a specific biometric.

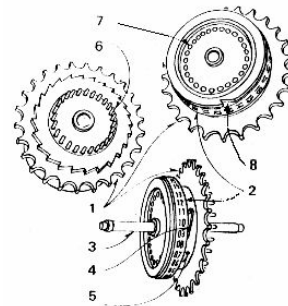
In the above context, the outline of this thesis can be summarized as follows:

- First of all we introduce the basic cryptographic primitives we need (Chapter 2), in particular the Paillier cryptosystem that is the main tool we use in our work. We explore the homomorphic properties of this cryptosystem and give a brief review of Paillier's variants and other homomorphic cryptosystems. The security model and the mathematical background to measure the complexities in bit operations, bandwidth and rounds will be described. At the end, all the cryptographic tools will be available to realize our constructions.
- Chapter 3 is devoted to examine biometrics and biometric systems. We will show that it is possible to design systems that work with different kinds of biometric samples and we will describe different ways to solve the problem of privacy protection in biometric systems. At the end all the background about biometric systems will be pointed out, this will be useful to better understand the two particular systems on which our constructions are based.
- FingerCode is the main topic of Chapter 4. The plain version of this algorithm will be investigated and its privacy preserving version will be detailed. To better understand the goal of the privacy preserving version of FingerCode we will rely on a real life scenario. After a detailed discussion we will evaluate all the complexities involved in the protocol and the related security. Finally some results about a real world implementation will be shown.
- Similar to the FingerCode case, Chapter 5 examines the Fuzzy Commitment Scheme approach to protect biometrics passing from a plain version to an encrypted one. Initially we will introduce the problem and the plain version of the algorithm. Then, we recall a scenario to clarify the main goals and requirement of the privacy preserving ver-

sion. Finally the entire protocol will be described by paying attention to analyze its complexity and security.

- In Chapter 6 we draw the conclusions of this thesis.

Cryptographic Tools for Privacy Preserving Protocols



Probability does not exist.
(Bruno De Finetti)

2.1 Homomorphic Cryptosystems

The problem of computing with encrypted data is a central one in the field of cryptography and goes back to the early days of modern cryptography, about thirty years ago [RAD78]. The problem has a fundamental importance both from a theoretical and a practical perspective. Often and especially in the case of number theoretic cryptosystems, the possibility of computing with encrypted data is a direct consequence of a common property of the cryptosystems: the malleability. More in detail:

Definition 2.1. Malleability. *We say that a cryptosystem is malleable if given an encryption of a plaintext m , it is possible to generate another ciphertext which decrypts to $f(m)$, for a known function f , without necessarily knowing or learning m .*

Although from a security point of view malleability is a weakness of a cryptosystem because it allows to modify the plaintext using just the ciphertext, in our context it is the key that allows to compute on encrypted data and thus permits to a third part to process private data. In fact, the most important application of this property is probably secure function evaluation. In its basic (two party computation) form, secure function evaluation allows two users, namely the Client (Alice) and the Server (Bob), to securely evaluate a known function (sometimes in form of a boolean circuit) using their private inputs. In other words, we require that executing the evaluation protocol does not reveal any knowledge about the inputs beyond what can be deduced merely from the computed output(s). Note that in general, this kind of protocols require interaction between the parties. Starting from the pioneering works of Yao [Yao82] (about the two party computation case) and Goldreich, Micali and Widgerson [GMW87] related to the general multi party computation case, the problem has been extensively studied in a variety of settings and under different assumptions.

In particular, due to the interactive nature of the protocols, great attention has been paid to the issue of reducing as much as possible the number of communication rounds required to realize the computation. In this sense, an intriguing line of research has been focused on encryption schemes (known as homomorphic encryption schemes) that allow one party to perform some basic operations on the encrypted messages by working only with the corresponding ciphertexts. Interestingly, the most known (see [GM84] [NS98] [Pai99] and [DJ01]) public key schemes are based on hard problems in number theory and present some form of homomorphism.

An homomorphic encryption scheme over an algebraic ring, can allow

different kind of homomorphisms, in particular:

Definition 2.2. *Given a public key cryptosystem and the relative public and private keys, respectively PuK and PrK , we indicate with $\llbracket \cdot \rrbracket$ the encryption function and with $\mathcal{D}(\cdot)$ the decryption function. We say that a cryptosystem is **homomorphic** if at least one of the following properties hold:*

Additive Homomorphism: *an operation \circ exists such that:*

$$\mathcal{D}(\llbracket x \rrbracket \circ \llbracket y \rrbracket) = \mathcal{D}(\llbracket x + y \rrbracket) = x + y$$

i.e., \circ maps addition in the encrypted domain;

Multiplicative Homomorphism: *an operation \bullet exists such that:*

$$\mathcal{D}(\llbracket x \rrbracket \bullet \llbracket y \rrbracket) = \mathcal{D}(\llbracket xy \rrbracket) = xy$$

i.e., \bullet maps multiplication in the encrypted domain;

Algebraic Homomorphism: *two operations \circ and \bullet exist, such that, contemporaneously, \circ maps addition and \bullet maps multiplication in the encrypted domain.*

Table 2.1 shows a list of cryptosystems with their homomorphic properties.

For several years the researcher community has believed that fully homomorphic cryptosystems were really difficult if not impossible to realize, but in 2009 in a breakthrough result by Gentry [Gen09] [vDGHV10], the first fully homomorphic encryption scheme was proposed. Gentry's paper shows how to use ideal lattices to construct an encryption scheme that allows to encrypt single bits and that is homomorphic with respect to addition and multiplication. Even though this result is a major theoretical achievement because secure fully homomorphic encryption was previously considered impossible to construct [BL96], the scheme itself and its recent improvements are still too inefficient to be used in practice. Very recently Melchor et al. in [MGH96] and

Table 2.1: Homomorphic Properties and Homomorphic Cryptosystem.

Cryptosystem	Add	Mult	Both
RSA (1978, [RSA78])	NO	YES	NO
Goldwasser-Micali (1982, [GM84])	YES	NO	NO
ElGamal (1985, [ElG85])	NO	YES	NO
Benaloh (1994, [Ben94])	YES	NO	NO
Paillier (1999, [Pai99])	YES	NO	NO
Boneh-Goh-Nissim (2004, [BGN05])	YES	only 1	YES
Gentry (2009, [Gen09])	YES	YES	YES

Gentry et al. in [GHV10], have conceived less general forms of homomorphic encryption schemes based on lattices which are more efficient than existing fully homomorphic schemes but still unsuitable for most applications. Such schemes are less general in the sense that they allow only a limited number of multiplications.

In recent years new solutions have been developed to process data and signals in the encrypted domain ([EFG⁺09], [BFK⁺10] [FB10]). Such solutions employ standard cryptographic tools, such as multi party computation and homomorphic encryption, and although some initial positive results have been obtained, the development of secure signal processing protocols in the real world is still complex and cannot be done with automatic approaches or procedures. This is mainly due to the fact that the proposed solutions are still not efficient enough to be employed in large scale applications and to take advantage of this in the real world. So actually, the best way to obtain the efficiency needed is to design optimized protocols that are able to solve a well established problem in a given scenario, i.e. to find *ad hoc* and high performance solutions in a fixed setting.

2.2 Security

In this section we introduce a few concepts that will be useful in the rest of the thesis. In particular we briefly introduce concepts and notations related to the security of the cryptosystems: hardness of computational and decisional problems and also the security model we used for our privacy preserving protocols: the honest but curious model.

2.2.1 Cryptosystem Security

In this work we mainly focus on cryptosystems that are based on hard problems in number theory. Generally speaking a cryptosystem is said to be secure if the related computational problem is computational infeasible to be solved.

Definition 2.3. *Hardness of Computational Problem:* given $c = \llbracket m \rrbracket$ a ciphertext, it is computational infeasible to find the corresponding plaintext m .

This is equivalent to affirm that the encryption function cannot be easily inverted. Sometimes a stronger security level is required, this is widely known as IND-CPA that means INDistinguishability under Chosen – Plaintext Attack or semantic security. In this case a cryptosystem needs to be probabilistic that is: for each plaintext it is possible to generate a set of valid ciphertexts. The definition of IND-CPA has been introduced in [GM84]. Simply speaking we can say that in a cryptosystem with this kind of security there is no adversary able to distinguish between encryptions of different plaintexts, even when he is allowed to compute encryptions by himself. To see this in detail we recall briefly the *Semantic Security challenge*:

1. A pair of PuK , PrK are generated by the challenger, the PuK is available to the attacker;
2. The attacker is able to perform all the operations he wants (encryption or other operations);

3. The attacker chooses two plains: m_0 and m_1 and sends them to the challenger;
4. The challenger flips a bit b and sends back $\llbracket m_b \rrbracket$;
5. The attacker outputs b ;

if the attacker is not able to understand b the cryptosystem is IND-CPA secure. The best way to prove the IND-CPA is to prove that the decisional problem related with the computational one is hard. In particular:

Definition 2.4. *Hardness of Decisional Problem:* given a ciphertext c and a plaintext m it is infeasible to decide if $c = \llbracket m \rrbracket$.

If for a given cryptosystem the Definition 2.4 is true (or it is assumed to be true) then the cryptosystem is IND-CPA.

2.2.2 Honest but Curious Model

When we run a multi party computation protocol we would like to have the same correctness and reciprocal privacy (assured for instance by a third party) than in the plain domain (trusted domain). In this thesis we concentrate on the *honest but curious* model, where both parties follow the protocol but try to infer additional information from the transcript of messages seen in the protocol. Far from trivial, this model covers many typical practical settings such as protection against insider attacks. Further, designing and evaluating the performance of protocols in the honest but curious model is a first step towards protocols with stronger security guarantees. Indeed, most protocols and implementations of protocols for practical privacy-preserving applications focus on the honest but curious model [LP09].

In the honest but curious model we assume that each party executes the protocol properly, but tries to compute as much additional information as possible with no time limitation. So the parties may deviate from the protocol

only in their internal computation, but the messages are in accordance with the protocol.

Moreover in most cases we compose sub-protocols to obtain more complicated functionalities, in this context it is really important to know that if all sub-protocols are proven secure in the honest but curious model than their sequential composition inherits this security property [Gol04].

2.3 Complexity

In this section we focus on the computational complexities involved in the privacy preserving protocols. We can analyze complexity from three different points of view:

Number of Bit Operations : this is also called computational complexity and indicates the number of basic operations that the protocol needs;

Number of Rounds : the protocols we focus on are client-server protocols, i.e. they require some message exchange to carry out the computation, a measure of the efficiency of a privacy preserving protocol is the *number* of the interactions (the number of message passing among the parties) it requires;

Bandwidth : this is just the amount of bit exchanged during the protocol execution.

To measure the number of bit operations we use the Big- \mathcal{O} notation [Kob94], so assuming that the biggest number involved in the computation has ℓ bits, namely the size of a ciphertext, we have that the cost to compute and addition between two numbers is `add` = $\mathcal{O}(\ell)$; `mult` = $\mathcal{O}(\ell^2)$ to compute a multiplication and finally `exp` = $\mathcal{O}(\ell^3)$ bit operations to compute an exponen-

tiation. We often need to compute exponentiation¹ by -1 (or other negative numbers), this operation is equivalent to compute the multiplicative inverse in the space of the ciphertexts (namely $\mathbb{Z}_{n^2}^*$), this operation can be computed by using the extended GCD and its computational complexity is equivalent to compute an exponentiation, so $\mathcal{O}(\ell^3)$. For the sake of simplicity in the rest of this thesis we will use as measure of computational complexity the number of exponentiations: `exp`.

While the number of rounds is a very simple concept, we spend a few words about the bandwidth. The bandwidth depends of many factors, but probably the most important one is the cryptosystem and so the size of the ciphertext. Sometimes it is possible to implement the cryptosystem over adequate elliptic curves [Ser99] that realize an algebra. This approach permits to save up bandwidth due to the fact that it is possible to use a smaller security parameter. We use this technique in our privacy preserving FingerCode construction (Chapter 4 see also Table 2.2) even if the operations on the elliptic curves could be a little bit less efficient from a computational complexity perspective.

2.4 Paillier Cryptosystem

Before giving a detailed description of the Paillier cryptosystem with its homomorphic properties, we will introduce the mathematical basis needed to prove the security of the scheme itself. Thus we briefly focus on the intractability of the *Composite Residuosity Class Problem* that is a generalization of the *Higher Residuosity Problem* (See Definition 2.5).

We start with a few basic concepts of number theory. Let $n = pq$ where p and q are prime numbers such that n is a secure RSA modulus. Due to Chinese Remainder Theorem [IRR90] we know that \mathbb{Z}_n is isomorph with

¹This is due to the fact that the exponentiation to a plaintext correspond to a multiplication in the Paillier domain, as we will see later in this Chapter.

$\mathbb{Z}_p \times \mathbb{Z}_q$ (the product of rings [Her05]), this isomorphism induces a group isomorphism over the multiplicative group, so, we have $\mathbb{Z}_n^* \simeq \mathbb{Z}_p^* \times \mathbb{Z}_q^*$ where \mathbb{Z}_p^* and \mathbb{Z}_q^* are cyclic groups of order $p-1$ and $q-1$ respectively. Now, if d is such that $d|(p-1)$ (d divides $p-1$) the set composed by the all d -th powers of all elements in \mathbb{Z}_p realizes a subgroup of \mathbb{Z}_p^* , moreover if $GCD(d, q-1) = 1$ it follows that all the elements of \mathbb{Z}_q^* can be expressed as d -th powers and by this we have that the set of all d -th powers generates a subgroup of \mathbb{Z}_n^* . Trivially if $GCD(d, q-1) = h$ than we have a set of $\frac{q-1}{h}$ d -th powers in \mathbb{Z}_q^* . The case $d = 2$ is well-known in number theory, in this case we speak of *quadratic residuals*.

Definition 2.5. *Hardness of the **Higher Residuosity Problem**. Given $n = pq$ with p and q prime numbers, an integer number d such that $d|(p-1)$ and an integer $x \in \mathbb{Z}_n$, it is computationally infeasible to decide if an integer γ exists such that $x = \gamma^d \pmod n$. If $d = 2$ the problem is called **Quadratic Residuosity Problem** (See [NS98] for further details).*

The Higher Residuosity Problem states that it is not possible to efficiently decide if a given integer is or is not a d -th power in the ring \mathbb{Z}_n . A generalization of this problem is the basic infeasible problem for the Paillier cryptosystem (See Definition 2.6).

Definition 2.6. *Hardness of the **Composite Residuosity Problem**. Given $c \in \mathbb{Z}_{n^2}^*$ and $\gamma \in \mathbb{Z}_n^*$ it is computationally infeasible to find $m \in \mathbb{Z}_n$ such that:*

$$c = \gamma^m r^n \pmod{n^2} \quad (2.1)$$

for some $r \in \mathbb{Z}_n^*$.

Definition 2.7 shows a definition for the decisional problem associated with the computational one.

Definition 2.7. ***Decisional Composite Residuosity Problem**. Given $c \in \mathbb{Z}_{n^2}^*$, $\gamma \in \mathbb{Z}_n^*$ and $m \in \mathbb{Z}_n$ it is computationally infeasible to decide if*

$$c = \gamma^m r^n \pmod{n^2} \quad (2.2)$$

for some $r \in \mathbb{Z}_n^*$.

All the above problems are considered intractable and so suitable as basis for the Paillier cryptosystem (detailed discussion can be found in [Pai99]). Given the above definition and hardness assumption it can be proved that the Paillier cryptosystem is a randomized IND-CPA cryptosystem.

To illustrate the way Paillier cryptosystem works, we start by defining the public and private keys, respectively PuK and PrK . Given an RSA modulus $n = pq$, we define $\mathcal{L}(u)$ as the following function:

$$\mathcal{L}(u) = \left\lfloor \frac{u-1}{n} \right\rfloor, \quad (2.3)$$

and we compute the Least Common Multiple $\lambda = LCM(p-1, q-1)$, choosing γ such that:

$$GCD\left(\mathcal{L}\left(\gamma^\lambda \bmod n^2\right), n\right) = 1, \quad (2.4)$$

then let $\mu = \mathcal{L}\left(\gamma^\lambda \bmod n^2\right)^{-1} \bmod n$. Finally we let:

$$\begin{aligned} PuK &= (\gamma, n) \\ PrK &= (\lambda, \mu). \end{aligned} \quad (2.5)$$

Given a plaintext $m \in \mathbb{Z}_n$ and a random $r \in \mathbb{Z}_n^*$ we compute the encryption of m in the following way:

$$c = \llbracket m \rrbracket = \gamma^m r^n \bmod n^2 \quad (2.6)$$

and the decryption as:

$$m = \mathcal{D}(c) = \mathcal{L}\left(c^\lambda \bmod n^2\right) \mu \bmod n. \quad (2.7)$$

In general we indicate with $s = \lceil \log_2 n \rceil$ the Paillier security parameter and we define $\ell = 2s$ the bit size of a ciphertext. The most updated NIST²

²National Institute of Standard and Technology. The mission of the Institute is to: "promote U.S. innovation and industrial competitiveness by advancing measurement science, standards, and technology in ways that enhance economic security and improve quality of life."

Table 2.2: NIST Recommendation for Key Management.

Date	Symmetric	Asymmetric	Elliptic Curve
2010	80	1024	160
2030	112	2048	224
> 2030	128	3072	256
>> 2030	192	7680	384
>>> 2030	256	15360	512

recommendation for security parameters are reported in Table 2.2 (more detail in [BBJ⁺09]).

Following Equation 2.6 and Equation 2.7 we remind that for Paillier cryptosystem the computational complexity is: $\text{enc} \approx \text{dec} = \text{exp}$.

Paillier cryptosystem has several properties. In the following we point out the most important ones with the related proofs. Given $x, y, k, r \in \mathbb{Z}_n$ we have:

Property 2.1. Additive Homomorphism.

- $\mathcal{D}(\llbracket x \rrbracket \llbracket y \rrbracket \bmod n^2) = x + y \bmod n$

Property 2.2. Scalar Homomorphism.

- $\mathcal{D}(\llbracket x \rrbracket^k \bmod n^2) = kx \bmod n$

Property 2.3. Self-Blinding.

- $\mathcal{D}(\llbracket x \rrbracket r^n \bmod n^2) = x \bmod n$

Proof 2.1. Additive and Scalar Homomorphism. Given $\llbracket m_1 \rrbracket, \llbracket m_2 \rrbracket$

and $k, r_1, r_2 \in \mathbb{Z}_n$ we have:

$$\begin{aligned}
c &= \\
&= (\llbracket m_1 \rrbracket \llbracket m_2 \rrbracket)^k = \\
&= (\gamma^{m_1} r_1^n \gamma^{m_2} r_2^n)^k \bmod n^2 = \\
&= (\gamma^{m_1+m_2} (r_1 r_2)^n)^k \bmod n^2 = \\
&= \gamma^{k(m_1+m_2)} (r_1 r_2)^{kn} \bmod n^2
\end{aligned} \tag{2.8}$$

now we define $r = (r_1 r_2)^k$ thus:

$$\gamma^{k(m_1+m_2)} (r_1 r_2)^{kn} \bmod n^2 = \gamma^{k(m_1+m_2)} r^n \bmod n^2 \tag{2.9}$$

finally $\mathcal{D}(\gamma^{k(m_1+m_2)} r^n \bmod n^2) = k(m_1 + m_2)$.

Proof 2.2. Self-Blinding. Given $\llbracket m \rrbracket$ and $r \in \mathbb{Z}_n$ we have:

$$\begin{aligned}
\llbracket m \rrbracket r^n \bmod n^2 &= \\
&= \gamma^m r_m^n r^n \bmod n^2 = \\
&= \gamma^m (r_m r)^n \bmod n^2 = \\
&= \gamma^m \hat{r}^n \bmod n^2 =
\end{aligned} \tag{2.10}$$

setting $\hat{r} = r_m r$, we have that this operation does not change the plaintext, but only the randomization. Similarly in the case:

$$\begin{aligned}
\llbracket m \rrbracket \gamma^{rn} \bmod n^2 &= \\
&= \gamma^m r_m^n \gamma^{rn} \bmod n^2 = \\
&= \gamma^m (r_m \gamma^r)^n \bmod n^2 = \\
&= \gamma^m \hat{r}^n \bmod n^2 =
\end{aligned} \tag{2.11}$$

where $\hat{r} = r_m \gamma^r$.

2.5 Damgaard – Jurik cryptosystem

Two years after the publication of Paillier’s work, Damgaard et al. in [DJ01] proposed a generalization of Paillier cryptosystem. In particular this generalization allows to use moduli of the form n^{s+1} where $s \geq 1$ moreover the security can be proven, similar to Paillier, using the Composite Residuosity Problem (Definition 2.6) and the Decisional Composite Residuosity Problem (Definition 2.7). The possibility to use an $s \geq 1$ affects the cryptosystem *expansion factor*. More in details, cryptosystems like RSA realize a transformation that does not change the size of the data to be encrypted. In fact the plaintext and the ciphertext domain are the same \mathbb{Z}_n . This means that the plains and the ciphers have the same bit size: $\lfloor \log_2 n \rfloor$. With randomized cryptosystems this does not happen, because for the same plain more than one cipher exists. In the Paillier cryptosystem we have plain in \mathbb{Z}_n and ciphers in \mathbb{Z}_{n^2} this implies that the data has an *expansion factor* of:

$$\frac{2\lfloor \log_2 n \rfloor}{\lfloor \log_2 n \rfloor} = 2.$$

The Damgaard – Jurik cryptosystem tries to avoid this problem because it has plains in \mathbb{Z}_{n^s} and ciphers in $\mathbb{Z}_{n^{s+1}}$ this produces an expansion factor equal to:

$$\frac{(s+1)\lfloor \log_2 n \rfloor}{s\lfloor \log_2 n \rfloor} = \frac{s+1}{s} = 1 + \frac{1}{s}$$

that, generally, is smaller than 2.

In DJ cryptosystem, we can define public and private keys in the following way. Given $n = pq$ a secure RSA modulo, we set $\lambda = LCM(p-1, q-1)$ and choose $\gamma \in \mathbb{Z}_{n^{s+1}}^*$ in the following way:

$$\gamma = (n+1)^\alpha \beta \bmod n^{s+1} \tag{2.12}$$

where α is such that $GCD(\alpha, n) = 1$ and $\beta \in \mathbb{Z}_n^*$. Due to the fact that $\alpha = 1$ and $\beta = 1$ satisfy the above condition, DJ cryptosystem is often used in the

simplified version in which $\gamma = n + 1$. Then by using the Chinese Remainder Theorem compute d such $d = 1 \pmod{n^s}$ and $d = 0 \pmod{\lambda}$. Lastly let:

$$\begin{aligned} PuK &= n \\ PrK &= \lambda. \end{aligned} \tag{2.13}$$

Now, given a plaintext $m \in \mathbb{Z}_{n^s}$ and a random $r \in \mathbb{Z}_{n^{s+1}}$ it is possible to compute the encryption of m as:

$$c = \llbracket m \rrbracket = \gamma^m r^{n^s} \pmod{n^{s+1}}, \tag{2.14}$$

and the decryption is computed via a recursive application of the Paillier decryption function on:

$$c^d \pmod{n^{s+1}} = (n + 1)^m \pmod{n^{s+1}} \tag{2.15}$$

to obtain m . DJ cryptosystem inherits the same homomorphic properties of Paillier's scheme.

2.6 Blinding

Very often to realize a privacy preserving protocol non linear functions are needed that cannot be computed by using only homomorphic encryption schemes. In those cases Bob (the server) asks to Alice (the client) some help to carry out a portion of the computation. This means that there is some kind of interaction between the parties (rounds) and during this everything must be kept secret, that is, when Bob sends its data to Alice, he wants to be sure that she is not able to understand anything about the data itself and vice versa [Koc96]. Formally we state Definition 2.8.

Definition 2.8. *Blind Computation with Encrypted Data.* *Bob has some data $\llbracket x \rrbracket$ encrypted with the public key of Alice and needs to compute the functionality f with the help of Alice. Due to the fact that Alice owns the private key she is able to obtain x and Bob does not want to reveal it to*

Alice. So he chooses a suitable r and by homomorphic properties computes $\llbracket x + r \rrbracket$ and sends it to Alice. She is able to decrypt and obtain $x + r$ but she cannot retrieve x , thus she computes: $\llbracket f(x + r) \rrbracket$ and sends it back to Bob that obtains the required computation. Obviously, it is necessary that \tilde{f} exists such that:

$$\tilde{f}(\llbracket f(x + r) \rrbracket, r) = \llbracket f(x) \rrbracket$$

and \tilde{f} can be applied on encrypted data. Figure 2.8 summarizes the flow of actions for blinding.

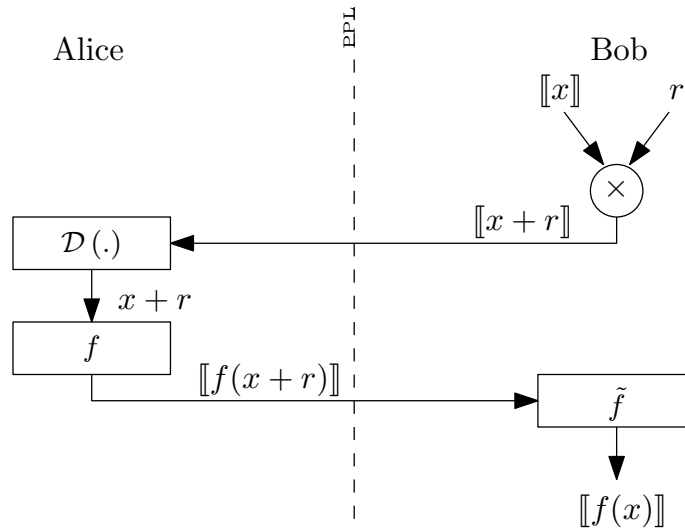


Figure 2.1: Blind Computation with Encrypted Data (PPL indicated the Privacy Preserving Line).

This practise is also powered from the fact that additive blinding is information theoretic secure, so it provides a perfect security on the data allowing at the same time the possibility of computing on encrypted data. The above approach is quite often used, and several sub-protocols have been developed using this approach. To exemplify the blinding procedure outlined above, we now describe the sub-protocol, `EncMul`, that allows to compute the product of two Paillier ciphertexts obtaining $\llbracket xy \rrbracket = \text{EncMul}(\llbracket x \rrbracket, \llbracket y \rrbracket)$. Suppose (See

Figure 2.2) that Bob owns $\llbracket x \rrbracket$ and $\llbracket y \rrbracket$ encrypted with the public key of Alice. He can obfuscate both ciphertexts by adding two random numbers due to homomorphic additive properties and obtain $\llbracket x + r_x \rrbracket$ and $\llbracket y + r_y \rrbracket$. Now he sends these ciphertexts to Alice, she decrypts and multiplies them finding: $w = xy + xr_y + yr_x + r_xr_y$, she encrypts w and sends it back to Bob that computes:

$$\begin{aligned}
\llbracket w \rrbracket \llbracket x \rrbracket^{-r_y} \llbracket y \rrbracket^{-r_x} \llbracket r_xr_y \rrbracket^{-1} &= \\
&= \llbracket w \rrbracket \llbracket -xr_y \rrbracket \llbracket -yr_x \rrbracket \llbracket -r_xr_y \rrbracket = \\
&= \llbracket w - xr_y - yr_x - r_xr_y \rrbracket = \\
&= \left\llbracket \underbrace{xy + xr_y + yr_x + r_xr_y}_w - xr_y - yr_x - r_xr_y \right\rrbracket = \\
&= \llbracket xy \rrbracket \tag{2.16}
\end{aligned}$$

obtaining exactly the product of the two encryptions.

Computing `EncMul` requires 2 rounds (one from Bob to send the obfuscated ciphertexts and one from Alice to send back the result) and a bandwidth of 3ℓ (3 ciphertexts are exchanged) with a computational complexity equal to: 3 `exp` needed to compute $\llbracket x \rrbracket^{-r_y}$, $\llbracket y \rrbracket^{-r_x}$ and $\llbracket r_xr_y \rrbracket^{-1}$; 5 `mult` needed to obfuscate $\llbracket x \rrbracket$, $\llbracket y \rrbracket$ and to compute the additions to $\llbracket w \rrbracket$; 2 `dec` to obtain in plain $x + r_x$ and $y + r_y$ and finally 1 `enc` to encrypt the result, for a total of 6 `exp` operations.

2.7 Application to Processing Encrypted Data

The techniques introduced in this chapter have been widely used, sometimes in conjunction with other advanced tools like garbled circuits³, to

³Garbled circuits are an efficient method for secure function evaluation of boolean circuits. The general idea of garbled circuits, that has been introduced by Yao [Yao86], is to encrypt (garble) each wire with a symmetric encryption scheme. In contrast to homomor-

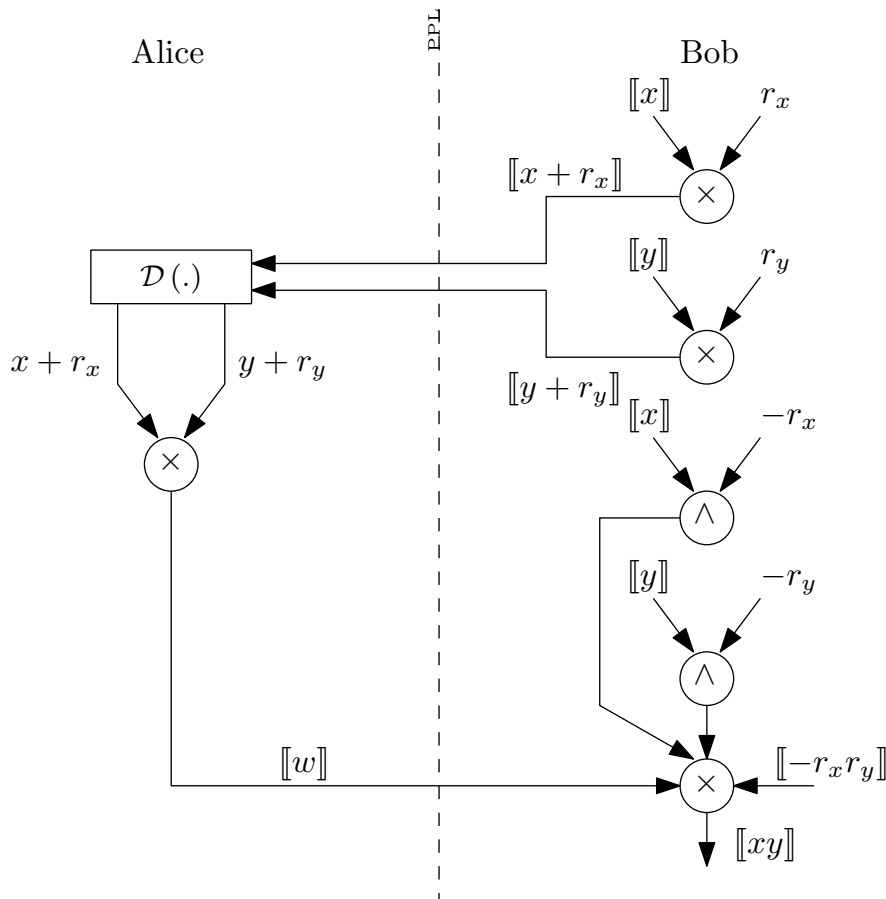


Figure 2.2: *EncMul Sub-Protocol.*

realize systems that are able to solve a variety of problems. We briefly recall some of them to give an idea of the possibilities allowed by this kind of approach. The number of possible applications is virtually endless. Among the most interesting scenarios investigated so far we mention: private database access [AS00], in which the client accesses a server by means of an encrypted

phic encryption, the encryptions/garblings cannot be operated directly, but requires helper information which is generated and exchanged in a setup phase in form of a garbled table for each gate.

query; private data mining [LP08], in which two or more parties wish to extract aggregate information from a dataset formed by the union of their private data; watermarking of encrypted signals [KLC⁺08], for digital rights management within buyer-seller protocols; recommender systems [ABF⁺08], in which users data is analyzed without disclosing it.

In [BPB08] a system to compute the Fast Fourier Transform in the encrypted domain is described. In [BFK⁺09] a privacy-preserving system has been described where Bob classifies an ElectroCardioGram (ECG) signal without learning any information about the ECG signal and Alice is prevented from gaining knowledge about the classification algorithm used by the Server. The system relies on the concept of Linear Branching Programs (LBP) and a related cryptographic protocol for secure evaluation of private LBPs [BFK⁺10] based on homomorphic encryption and garbled circuits. The paper faces with the study of the trade-off between signal representation accuracy and system complexity both from practical and theoretical perspectives. As a result, the inputs to the system are represented with the minimum number of bits ensuring the same classification accuracy of a plain implementation.

In [Fai10], a novel technique has been proposed to compute the well-known A^* algorithm, on the encrypted weights of a graph. A^* is a *best first* graph search algorithm that uses an heuristic function helping to choose the best candidates during the traversing of common graphs [HNR68]. Graphs are data structures widely used to represent: social networks; computer networks; geographic maps; game moves; possible paths in a given environment and many more. In the considered setting two parties are interested to compute the shortest path between two nodes in a context where: part of the graph topology (only the numbers of nodes) is publicly known; Alice knows the weights for each edges and Bob owns the heuristic to use for searching in. Moreover, Alice wants to keep secret her weights and Bob the heuristic used.

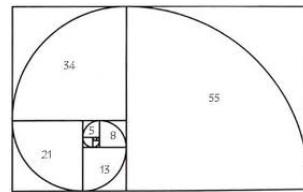
In [FB10], a scenario in which two parties are interested in computing a given functionality in a privacy preserving way has been considered, but

this functionality needs a sub-protocol that computes the *Gram – Schmidt Orthogonalization* on encrypted vectors. The main goal of the paper is a detailed description comprehensive of security proof and complexity evaluation. There are a lot of applications in which this kind of sub-protocol could be embedded as a basic privacy preserving primitive, including: QR decomposition [GVL96]; linear least squares problems [Bjo67]; face recognition [ZZZ04]; improving performances of neural networks [Orf90]; wavelets computation [CQ92]; principal component analysis [SP07] and image compression [MQQ⁺06].

In [EFG⁺09] a privacy-enhanced face recognition system is proposed. In particular the construction allows to efficiently hide both the biometric using an encrypted version of the widely known Eigenfaces algorithm and it is able to keep secret the result from the server that performs the matching operation. Similarly in [OPJM10] is an ad hoc system for face recognition in the privacy preserving framework is proposed, specifically designed for usage in secure computation.

Chapter 3

Biometric Systems



11:15, restate my assumptions:

1. Mathematics is the language of nature.
2. Everything around us can be understood through numbers.
3. If you graph these numbers, patterns emerge.

Therefore: There are patterns everywhere in nature.

(“Pi” 1998)

3.1 Introduction to Biometric Systems

Generally speaking *biometrics* is the science of measuring an individual’s physiological and/or behavioral properties (or features) in an automatic way. More in details the term *biometric recognition* is used to indicate the use of distinctive physiological (e.g., fingerprints, face, retina, iris) and behavioral (e.g., gait, signature) characteristics, called *biometric identifiers* (or simply biometrics) for automatically recognizing individuals.

Physiological biometric is based on data derived from a direct measurement of a part of the human body, while *behavioral* biometric is based on

measurements and data derived from a human action, but it is possible to affirm that all biometric identifiers are a combination of physiological and behavioral characteristics and they should not be exclusively classified into either physiological or behavioral characteristics. For example, fingerprints may be physiological in nature, but the usage of the input device (e.g., how a user presents a finger to the fingerprint scanner) depends on the person's behavior. Thus, the data produced is a combination of physiological and behavioral characteristics. Often, a similarity can be noticed among relatives, children, and siblings in their voice, gait, and even signature. The same argument applies to face: faces of identical twins may be extremely similar at birth but afterwards, the faces change based on the person's behavior and history (e.g., lifestyle differences leading to a difference in bodyweight, etc.).

Advances in automation and the development of new technological systems, such as cellular phones and the Internet, have led users to more frequent use of technical devices that require some kind of authentication. Personal identification has taken the form of token-based or knowledge-based methods, such as secret passwords and *PINs* (Personal Identification Numbers), ID cards, keys or passes. Everyday-life examples include ATMs, cellular phones or Internet access on a personal computer. A password should never be guessed, so it should be as long as possible, it should not appear in a dictionary, and should include special symbols such as +, -, %, or #. Moreover, for security purposes, a password should never be written down, never be given to another person, and should be changed at least every three months [Bis91]. Considering that people use many passwords (laptop PCs, log on LANs, POS cards, cellular phones) and that the expense and annoyance of a forgotten password is enormous, it is clear that users are forced to sacrifice security due to memory limitations. While passwords are very machine-friendly, they are still far from being user-friendly.

One of the possible solutions to the above problems can be found in the use of biometry. Because biometric identifiers cannot be easily misplaced, forged,

or shared, they are considered more reliable for person recognition than traditional methods. Biometric recognition may provide user convenience (e.g., money withdrawal without ATM card or PIN), better security (e.g., difficult to forge access), and higher efficiency (e.g., lower overhead for computer password maintenance).

Biometric technologies are becoming the foundation of an extensive set of highly secure identification and personal verification solutions. As the level of security breaches and transaction fraud increases, the need for highly secure identification and personal verification technologies is becoming apparent.

The problem of associating an identity to an individual can be split in two distinct types of problems: *verification* that is confirming or denying a person's identity¹ (one to one comparison); and *recognition* that is establishing a subject's identity in a set of possible candidates (one to many comparison). It is obvious that a lot of situations exist that require the identification of a subject in the real world, for instance consider the case of video-surveillance in public locations like: subways, airports or train stations.

Focusing on the engineering point of view, the problem of authenticating a person can be reduced to the problem of authenticating a concrete entity related to the person, by this, we can identify two principal cases: authentication based on **something that you possess** like the ATM card or **something that you know** like a username and a password. A completely different approach is to identify a person by using the physical characteristics of the person himself. As mentioned before this method is called biometric recognition and can be categorized, with the above notation, as **something that you are**. A good biometry should have the following four properties [Cla94]:

universality: every person owns the biometry. This means that is a common peculiarity among people;

¹Sometimes also the term *identification* is used, in this case it refers to a binary matching.

Table 3.1: Comparison Among the most Common Biometrics - Intrinsic Properties.

Biometric	Universality	Uniqueness	Permanence	Collectability
Fingerprint	Medium	High	High	Medium
Face	High	Low	Medium	High
Hand	Medium	Medium	Medium	High
Iris	High	High	High	Medium
Signature	Low	Low	Low	High
Voice	Medium	Low	Low	Medium
DNA	High	High	High	Low

uniqueness: two persons do not own two identical biometrics or the probability of this event is negligible, for instance the iris is really distinctive;

permanence: the characteristic is time invariant, for instance the hand geometry change during the growth while DNA does not;

collectability: the characteristic can be measured quantitatively and easily with a measurement tool.

Table 3.1 shows the above properties for a set of common biometry.

Other three properties have to be added to those above, that are strictly related to the practical implementation of the biometric systems:

performance: the resources required to achieve a good identification accuracy, for instance biometric systems based on DNA are really expensive and require many days to perform an identification task;

acceptability: the confidence on the system by people. The measurement system has to be non invasive and user-friendly;

circumvention: robustness under fraudulent techniques, this means that it should be really difficult (or impossible) to replicate a biometric trait of

Table 3.2: Comparison Among the most Common Biometrics - Implementation Properties.

Biometric	Performance	Acceptability	Circumvention
Fingerprint	High	Medium	Medium
Face	Low	High	Low
Hand	Medium	Medium	Medium
Iris	High	Low	High
Signature	Low	High	Low
Voice	Low	High	Low
DNA	High	Low	Low

a third person.

Table 3.2 summarizes the above additional properties for some common biometrics.

A biometric-based authentication system is generally composed by two main phases: the enrollment and the matching. During the enrollment phase a new biometric sample is used to generate a template or another compact representation of the data that is stored in a database. In the matching phase, the client provides a new sample of his biometry that is matched against the enrolled one to check the identity. Figure 3.1 summarizes the flow of the matching phase: a) the biometric sample is acquired; b) a compact representation is computed (features extraction); c) a matching is looked for using a specified metric defined in the features space.

A biometric system is essentially a pattern recognition system for person identification relying on the authenticity of a specific physiological or behavioral characteristic possessed by the user. An important issue in designing a practical system is to determine how an individual is identified. Depending on the context, a biometric system can be either a verification (authentication)

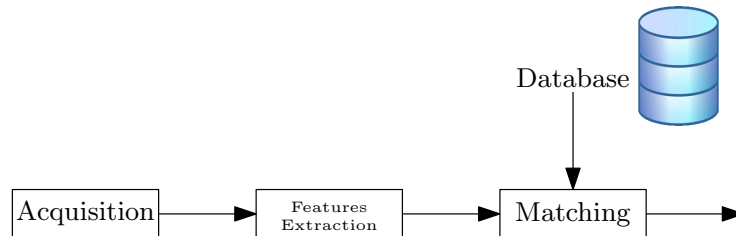


Figure 3.1: Architecture of a Pattern Recognition system.

system or an identification system.

For the sake of simplicity we can say that **verification** answers to the question "Am I whom I claim I am?", this clearly involves confirming or denying a person's claimed identity. Similarly in the **identification** case, one has to establish a person's identity answering to "Who am I?". Each of these problems have their own complexities and could probably be solved best by a certain biometric system.

Verification (**1:1** one-to-one matching) is the process of establishing the validity of a claimed identity by comparing a verification template to an enrolled template. Verification requires that an identity be claimed, after which the individual's enrollment template is located and compared with the verification template. Some verification systems perform very limited searches against multiple enrolled records. For example, a user with three enrolled fingerprint templates may be able to place any of the three fingers to be verified, and the system performs three 1:1 matches against the user's enrolled templates until a match is found.

Recognition (**1:N**, one-to-many recognition) is the process of determining a person's identity by performing matches against multiple biometric templates. Identification systems are designed to determine identity based solely on biometric information. There are two types of identification systems: positive identification and negative identification. *Positive identification* answers the "Who am I?" question, although the response is not necessarily a name

– it could be an employee ID or another unique identifier. A typical positive identification system would be a prison release program where users do not enter an ID number or use a card, but simply look at an iris captured by a device and then try to find a match in a database. *Negative identification* systems search databases in the same fashion, comparing one template against many, but are designed to ensure that a person is not present in a database. This prevents people from enrolling twice in a system, and is often used in large-scale public benefits programs in which users could try to be enrolled multiple times to gain benefits using different names.

3.2 Biometrics

In this section we give some details about the most common and widely used biometrics, in particular we focus on: fingerprint, face, iris, signature, voice and DNA. Let us now detail the characteristics of each of the above biometric traits.

3.2.1 Fingerprint

Fingerprints are the graphical ridges present on human fingers. These biometric traits are believed to be unique for each person, in general they are different also from finger to finger and have been intensively used in forensics, for many years, for criminal investigations. Generally speaking a fingerprint image can be captured in one of the two following ways: a) scanning an inked impression of a finger (*inked fingerprint*) or b) using a fingerprint scanner device (*live-scan fingerprint*). Fingerprint images can be represented as: a picture, a sequence of finger ridges or a set of features, said minutiae, extracted from the conformation of ridges. The prevalent approaches to fingerprint identification can be summarized as follows:

1. invariant properties of the gray-scale image;

2. global ridge pattern (fingerprint classes);
3. local ridge patterns;
4. minutiae.

The lines that flow across fingerprints are called ridges and the spaces between ridges are called valleys. One of the best known approaches to fingerprint feature extraction is called *minutiae matching*. There are several types of minutiae, but the most important are: ridge ending and ridge bifurcation. Those minutiae are stored with some additional information like: location in the image or direction. Sometimes, in this approach, two other features are used: the core, that is the center of the fingerprint and the delta that is a singular point from which three patterns deviate (See Figure 3.2).



Figure 3.2: *Fingerprint Minutiae.*

Commonly a digital fingerprint image is stored in gray scale: 8 bits for intensities for a range going from 0 to 255. The image size is a square from 1.27 cm to 2.54 cm with a resolution of 500 dpi (200 dpc). Image processing tools are applied to remove noise and enhance fingerprint images, for instance: adaptive matched filter and adaptive threshold filter. The first one is used to

highlight ridges and the second one to obtain the binarization of the image (from 8 bit representation to 1 bit representation). After these operations, the edges are thinned to squeeze ridges in lines of 1 single pixel. Extracting minutiae requires to find ridge endings and ridge bifurcations, this procedure could introduce extraneous features that can be removed by using empirical thresholds. The minutiae are stored in a vector with at least two additional information the (x, y) coordinate location, the direction and the type of minutiae; this vector is called minutiae template, its size ranges from 400 bytes to 1024 bytes. To implement the matching technique a similarity function is used that gives the similarity of two feature vectors. Note that in the space of the minutiae it is not possible to define a metric, so generally speaking, the similarity functions are complex decision rules. Generally the feature vector is compared with all the other vectors stored in the database, so, to speed up the process, sometimes an order (importance) is given to the single minutiae stored in the vector.

Other techniques are based on correlation matching. In this case the algorithm works directly on two images and the main operation is the computation of the difference between of the two images. This approach could be very difficult because the images have to be aligned, rotated, zoomed and shrunk in the same way. Yet another approach consists in the use of the so-called FingerCode [JPHP00]. The FingerCode is a widely used algorithm in biometric systems, we will speak extensively about this approach later in the thesis.

3.2.2 Face

Face is one of the most accepted biometric traits because it is the standard method of identification used in human interactions, moreover acquiring faces is non-intrusive and can be done with low cost devices. Two principal methods are used to identify people by faces:

1. transform-based approach (eg. eigenfaces) [TP91]

2. attribute-based approach (requiring geometric properties extraction see Figure 3.3) [AGR96].

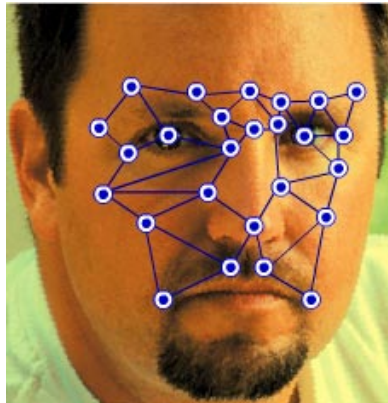


Figure 3.3: *Face Geometry Measurement.*

Algorithms for face recognition can be really complex because they must be tolerant to: aging effect, facial expressions, variation of poses and camera position. Generally speaking the features can be extracted manually or automatically: in the first case a human operator defines which are the features. The second case is the most common and the most interesting one from the point of view of practical implementations. Neural networks are often used to efficiently extract features (see for example [WAH97]). Other methods are related to statistical informations. A face can be represented as a linear combination of a principal component vector, that is computed from a dataset of images [TP91]: this approach is called eigenfaces (See Figure 3.4). A common problem of the eigenface technique is that it is not invariant to face position or size and so generally the images are required to be acquired in canonical form².

²With *canonical* form we refer to the fact that each face has to be in the same position and all the images need to have the same shrink factor and size. For instance the photos used in the passport are canonical photos.

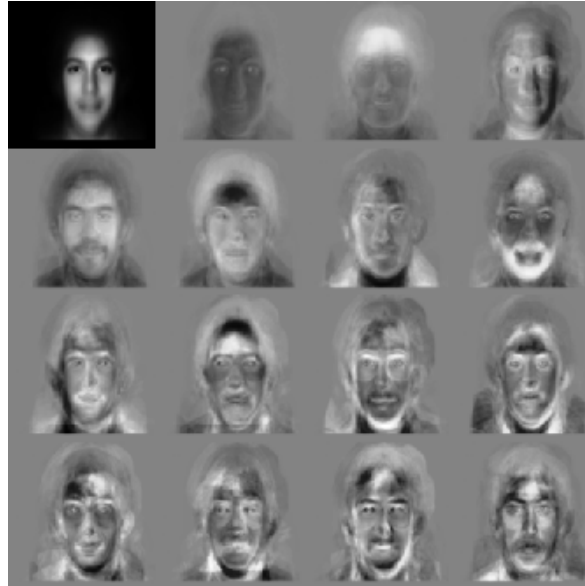


Figure 3.4: *Eigenface Samples.*

3.2.3 Iris

The visual texture of the iris is commonly believed to be unique for each person and for each eye [Dau93] so it is strongly suitable for biometric applications. Given the high number of degrees of freedom intrinsic in the iris (circa 249 [Dau03]), it is possible to identify a person with a very high confidence, moreover it is possible to affirm that quite all the variables that generate the iris are independent [Dau93] and so this biometry is really distinctive. The process of acquiring an iris image employs a non-contact device that generally uses a CCD with a resolution of 512 dpi. Iris is quite difficult to measure because there are several factors that are involved in this process. Probably the most important one is that to record an iris image a great cooperation between the operator and the subject is needed; the iris image in fact must be registered with a high accuracy of the focal distance, for this reason users can feel the iris more intrusive than other biometrics. By manipulating the iris

image it is possible to extract a constant length binary vector, called *IrisCode* [Dau06]. This byte vector permits an extremely fast method of recognition. The information in the iris pattern (See Figure 3.5) can be extracted using different approaches for instance a demodulation with complex wavelets (see [DD95]).

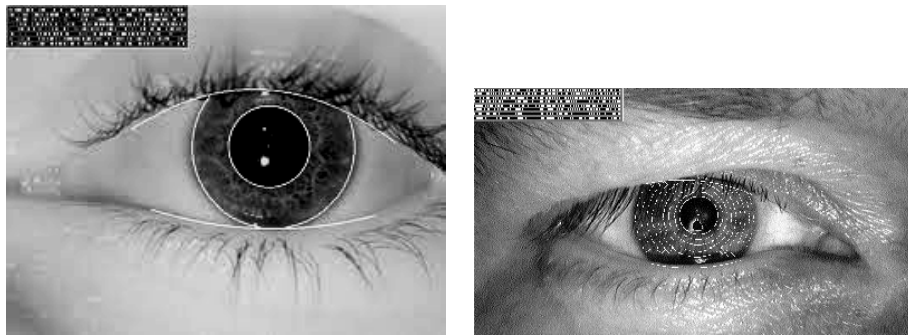


Figure 3.5: *IrisCode*.

After the localization of the iris in the image, a double dimensionless coordinate system is applied to define a mapping of the tissue in a way that is invariant to pupillary constriction or zoom factor. As already said the iris is detailed using a wavelet transform and the pattern is encoded into 256 bytes: the *IrisCode*. Due to the fact the *IrisCode* is really distinctive, a simple Hamming Distance can be used to identify a subject. The Hamming Distance is implemented by using the bitwise XOR, so the complexity to execute Iris Recognition is very low and so it is very attractive especially in systems that use very large databases (i.e. airport, subway) [Dau05].

3.2.4 Voice

The voice is an individual characteristic [Fur97], in general it is considered non sufficiently unique (low uniqueness) to guarantee an exact match in large databases. This because the voice signal is degraded in quality by microphone,

digitalization and communication channel. Moreover voice is a behavioral biometry and it is strongly affected by person's health, stress and emotions. There are two different approaches to voice recognition:

1. text-dependent: speaker verification is carried out on a set of predetermined phrases, this is the most common approach;
2. text-independent: speaker verification is carried out on a generic phrase. This is a very difficult task to achieve.

From the point of view of the circumvention, the voice is not so good due to the fact that some people are very skilled in mimicking voice. The most commonly used feature is the cepstral feature, which is the logarithm of the Fourier Transform coefficients in several bands (this feature is widely used in music classification [JLZ⁺02]). The feature vector is compared with the speaker model using pattern recognition techniques like: Dynamic Time Warping [WG97], Hidden Markov Model [LHW98], Neural Networks [FMA94] or Vector Quantization [CG90]. The result of this matching is a measure of similarity of the given features with the speaker model. Initially the data acquired is filtered by using an antialiasing filter and then converted into a digital format (12 or 16 bit at 8000 to 20000 samples per second). As said, one of the most used features is the cepstrum, but several others features can be extracted from the voice, in particular an AutoRegressive model (AR) can be applied to obtain a model for the speaker (often a 8-degree AR model is used) and in this case the features are computed from the coefficients of the model by changing their domain. In this way AR coefficients are transformed using: Reflection Coefficients [RSR78], LogArea Ratio [Ita75] or LP cepstrum [RSR78]. Pattern matching uses a distance measure to evaluate the similarity between the feature vector and the templates stored in the database, in this scenario some common solutions are: Vector Quantization Source Modeling [EW06] or Nearest Neighbors [HBP02].

3.2.5 DNA

The DNA is a one dimensional code that identifies each person uniquely, except for mono-zygote twins. Although people think that DNA is the best way to identify a person with no error, this principally comes from television fiction; in real world applications, there are three fundamental limits to the use of DNA for identity recognition:

contamination and sensitivity : it is quite simple to steal a piece of DNA, so everyone is able to obtain a third person DNA (i.e. hair);

real-time identification : with the actual technology it is still quite difficult the manipulation of the DNA to obtain relevant information (3 to 5 days are needed to extract a DNA profile);

privacy : certain diseases can be detected by examining DNA, so abuse of genetic code could be discriminant for instance in hiring practices.

DNA samples are used to generate a DNA fingerprint (or DNA profiling); to do so the portion of DNA usable is really short, in fact to compute a DNA fingerprint just 10% of the DNA extracted in the portion of non-coding short tandem repeat³ is used. More specifically the lengths of variable sections of repetitive DNA, such as short tandem repeats and minisatellites⁴, are compared between people [CM94]. Figure 3.6 shows samples of DNA fingerprints.

From the point of view of the uniqueness, it can be shown that the probability of two people having the same profile is about 10^{-9} , but it is also known that it is impossible to distinguish between mono-zygote twins. This kind of fingerprint extracted from DNA cannot be altered by surgery or treatment and for this reason is widely used for paternity test and forensics (i.e. CODIS⁵

³A tandem repeat in DNA occurs when a pattern of two or more nucleotides are repeated and the repeated sequences are directly adjacent to each other.

⁴A minisatellite is a section of DNA that consists of a short series of bases.

⁵Combined DNA Index System is the software used by the FBI laboratories to store and search among DNA profiles.

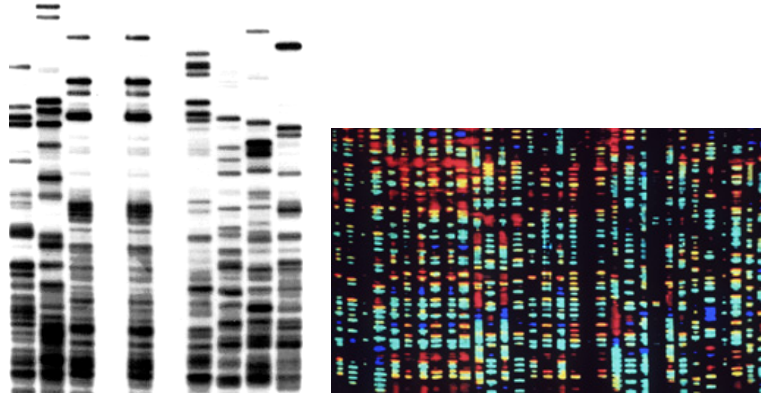


Figure 3.6: *DNA Fingerprint.*

database).

3.3 Protecting Biometric Data

3.3.1 Generalities

Biometric templates are uniquely associated with each person and thus represent the strongest form of personal identifiable information. If from one hand, such fact strengthens the authentication process, on the other hand the possibility that a biometric template could be stolen or exchanged raises concerns on its possible uses and abuses. A first concern seems to be the possibility that a government agency or a company which maintains personal data monitors and tracks the actions and the behavior of each individual. This may augment the enormous amount of information which already both public and private organizations can collect by tracking, for instance, credit cards or mobile phones.

Another basic concern regards the loss of anonymity when biometrics is used in a pervasive way. The control on the release of personal information

should always be kept with its owner, so that he could maintain the capability to avoid that other parties know who he is and avoid *Big Brother* scenarios and identity misuses.

The users commonly perceive biometric authentication and identification techniques as a threat to their privacy rights. In particular there are some aspects that reinforce this perception [JRP06a]. The first one is related to the fact that the acquisition of the biometric traits is considered as an exact and permanent filing of the user's activities and behaviors. For example, it is very common think that most biometric systems have 100% identification accuracy and that the biometric samples and templates are necessarily stored and/or sent over a network, exposing them to further risks of being exposed. Actually, the latter is a well-funded concern. In fact, while it should be granted to the user that the biometric information collected is not used for any other activities in addition to those expressly declared, in some cases it is hard to grant this aspect, especially if the biometric samples themselves are sent over an open network. The second issue is related to the possibility to track the user activities associated to the biometric acquisition, even in the far future. This raises in the users the perception of the possibility that their movements are tracked along with buying and life style. Commonly this issue is associated to a sort of phobia, in which a superior entity is capable of observing and acquiring knowledge on each activity of the user.

In a negligible part of the population, the usage of a biometric system is also perceived as uncomfortable or dangerous. For example, the fingerprint sensor – when previously used by other people and not properly cleaned – can be considered as unpleasant or disgusting. Face and iris acquisition systems might induce apprehension to have the eyes damaged by lasers and/or IR sources. Very interestingly, users often overlook other privacy related problems arising when biometrics are involved.

The first point concerns the possible usage of biometric information for operating *Proscription Lists*. For example, a user can be classified from a

previous behavior or activity in a specific class, and then – as a consequence of this classification – some services and accesses can be denied. Important examples of this situation are the black lists maintained by call centers and service providers especially designed to identify and to manage users considered as offending or no-collaborative. Other examples are the bad-credit lists filled by investors and mutual funds companies. Indeed, proscription lists can be employed also without the adoption of biometric systems (and actually they are), but the usage of biometric technologies can make the situation by far more dramatic.

The second point concerns the fact that many biometric features can be used to *obtain personal information* about the users, such as medical information of past illnesses or the current (and future) clinical trends. For example, the retinal pattern acquired by a biometric system can provide valuable information about the presence of hypertension, diabetes and others illnesses [JJW⁺93]. Much more personal information can be extracted from DNA samples [IR97].

The real risk of privacy invasiveness can be analyzed in more detail with respect to both the final application the biometric system is dedicated to and the biometric trait which is involved. Biometric *covert applications* (such as surveillance systems without explicit authorization from the users) are considered to be more privacy invasive. On the other hand, the biometric systems for identification or verification that are *optional* are considered to be more privacy compliant. In this case, users can decide to not be checked by the biometric system, and they can adopt a different identification/verification strategy.

Privacy is considered to be exposed to a greater risk when the biometric system performs an *identification* instead of a simple verification, in the last case the identity is claimed, so there is no worry about revealing the identity. That is related to the fact that the identification process encompasses a 1-to-many comparison, which, in most cases, is not carried out in the same

place of the acquisition (typically, the biometric data is sent through a network to a database for the comparison). Also the *duration* of the retention of the biometric data impacts the privacy risk. If retention expires in a fixed period of time, the privacy risk is reduced. Best practice notions require that for every project which encompasses biometric data retention should always explicitly state its duration.

Different risks are present with respect to the application scenario: the biometric setups in the *public sector* are considered to be more susceptible to privacy invasiveness than the same installations in the *private sector*.

Also the *role* of the individuals that use the biometric system has a great impact on the privacy. The most relevant privacy invasion is related to the association of the fundamental rights of the individual to a biometric identity test. The privacy risks are lower in applications where the individuals retain usage rights over the biometric data.

Another useful taxonomy concerns the different approaches for biometric data collection and storing. The IBG⁶ classifies four different classes concerning privacy protection: Protective, Invasive, Neutral, Sympathetic [IBG03]. A *privacy-protective* system is designed to protect or limit the access to personal information, providing a means for an individual to establish a trusted identity. In this case, the biometric systems use biometric data to protect personal information which might otherwise be copied, stolen or misused. A *privacy-sympathetic* system limits access/usage to personal data. A *privacy-sympathetic* approach encompasses the specific design of elements able to protect biometric data from unauthorized access and usage. Also the storage and the transmission of biometric data must be informed, if not driven, by privacy concerns. In a *privacy-neutral* system, privacy aspects are not important or the potential privacy impact is light. Privacy-neutral systems are designed to be difficult to misuse with regards to privacy issues, but they do not have the capability to protect personal privacy. A *privacy-invasive*

⁶International Biometric Group <http://www.biometricgroup.com/>.

system facilitates or enables the usage of personal data in a fashion which is contrary to privacy principles. In privacy-invasive systems, personal data are used for purposes broader than what originally intended. Systems which facilitate the linkage of personal data without an individual's consent, and those in which personal data are loosely protected belong to this class.

3.3.2 Privacy Protection of Biometric Data

As already said a typical biometric authentication system consists of two phases. During the enrollment phase, Alice provides her biometric data, from which features are extracted and a template is created and stored, either in a central database, or on a mobile device. During the authentication phase, a client who claims to be Alice would give her biometric data again, and the same feature extraction algorithm is applied. The result is then compared with the stored template. If they are sufficiently similar according to some similarity measure, the client is authenticated.

There has been intensive study on how to secure the biometric templates in recent years and a comprehensive coverage of many proposed solutions can also be found in [JNN08]. These techniques can be roughly categorized into three types:

1. approaches based on non-invertible transformations where similarity of biometric samples would be preserved through the transformation, but for which it is difficult to find the original template from a transformed one (e.g., [ASNM05] [RCCB07])
2. methods based on helper-data, where a recently proposed cryptographic primitive, the secure sketch, (or a variant of it) is employed, such that given a noisy biometric sample, the original biometric data can be recovered with the help of some additional information (i.e., a sketch), which makes it possible to use biometric data in the same way passwords are used. These techniques include [JW99] [JS06] [SLM07a]

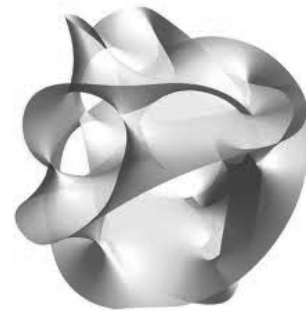
3. methods based on micro and macro data like the concepts of k -anonymity where portion of biometric data is released but requiring that it can be indistinctly matched to at least k respondents [SS98] or [Sam01].

The use of homomorphic encryption techniques to add a privacy protection layer in biometric application is a natural extension of the potentiality of this kind of cryptographic tools to achieve the goal of protecting the biometric traits. In particular processing biometric traits requires tools that are really common in the pattern recognition field. It is clear that the application of these tools to *encrypted* data, is an open problem, because computation in the encrypted domain is difficult and strictly related to specific cases and applications (see [OPB07], [LLM06]). The approach to privacy preserving can be seen as a Privacy Preserving Database Access, specifically: Alice wants to access a database owned or managed by Bob. In this setting we can identify three specific settings:

- a) **encrypted query to plain database:** in this case we want to protect what we are searching (this is the approach we will use in Chapter 4);
- b) **plain query to encrypted database:** in this case we want to protect the data on which we are searching;
- c) **encrypted query to encrypted database:** in this case we want to protect the query and also the data, in this class includes the technique proposed in Chapter 5.

In this explanation, the application of privacy preserving protocols achieves the goal of protecting biometrics (owned by Alice) against the party in charge of checking the identity (Bob), and vice versa it protects the information stored in the database from external users (an example is presented in [EFG⁺09]).

Privacy Preserving FingerCode



*Archimedes will be remembered when Aeschylus is forgotten,
because languages die and mathematical ideas do not.*

(Godfrey Harold Hardy)

In the present chapter we will introduce our construction to realize a privacy preserving version of the FingerCode algorithm.

4.1 Introduction

Several approaches to automatic fingerprint matching have been proposed in the literature. Probably the most popular ones are based on the minutiae pattern of the fingerprint and in general are called minutiae-based approaches. A large part of these methods require extensive preprocessing operations (i.e. extraction of ridge orientation and direction, flow estimation, ridge segmentation, ridge thinning, minutiae detection) in order to reliably

extract the minutia features [JHPB97]. For this reason, methods based on minutiae do not seem suitable for an implementation in a secure two party computation framework. Another class of fingerprint matching approaches do not use the minutiae features of the fingerprint, but try to match directly the fingerprint images [WWP00], or match features extracted from the image by means of certain filtering or transform operations. In particular the algorithm described in [Lee99] is based on a specific representation of the fingerprints which yields a relatively compact and fixed length code, called FingerCode [JPHP00] that it is suitable for matching as well as storage on a smartcard. Using the FingerCode, the matching is really fast and the representation is amenable to be indexed. This technique utilizes both the global flow of ridges and valleys and the local ridge characteristics, to generate a short fixed length code representing the fingerprints while maintaining a high recognition accuracy. Our construction is based on an adaption of the FingerCode algorithm that works in the encrypted domain.

4.2 FingerCode–Based Authentication

In the following we examine the two main blocks that are the basis of biometric systems based on the FingerCode template representation: feature extraction and matching.

4.2.1 FingerCode Construction

The feature extraction algorithm can be split in four main steps:

- determine a reference point in the fingerprint image;
- tessellate the region of interest around the reference point;
- apply a filter in the region of interest using eight¹ different directions of a bank of Gabor filters;

¹Eight directions are required to completely capture the local ridge characteristics in

- compute the average absolute deviation from the mean of gray values in individual sectors of the filtered image to define the feature vector or the FingerCode.

These steps are summarized in Figure 4.1.

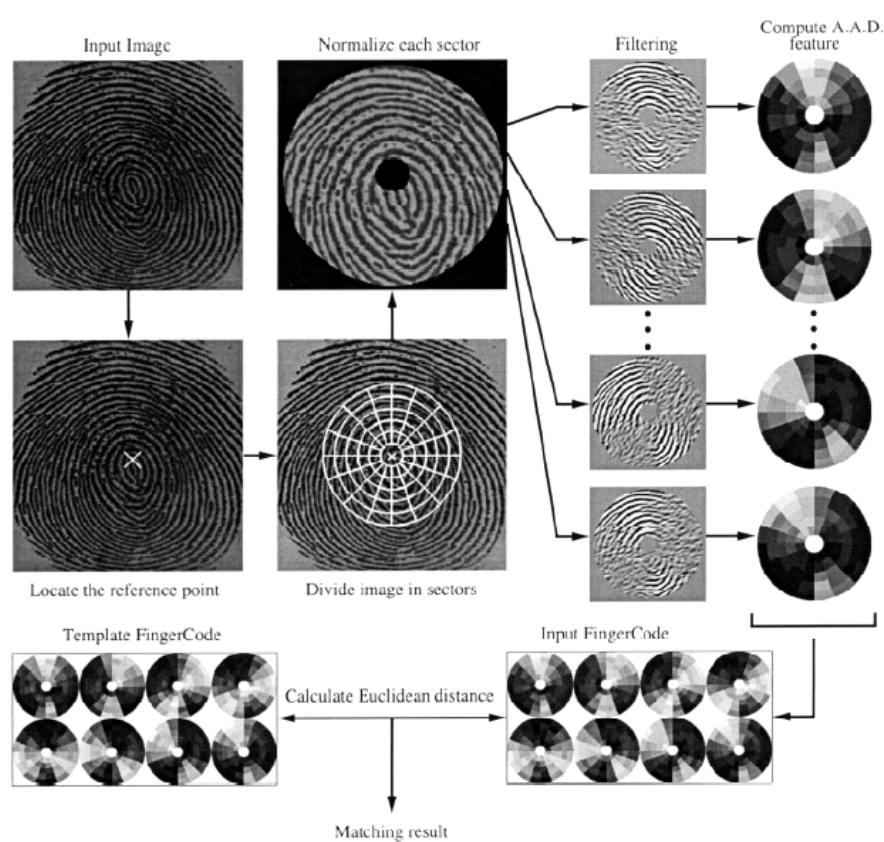


Figure 4.1: *FingerCode Authentication System.*

a fingerprint while only four directions are required to capture the global configuration [JPH99].

Identification of the Reference Point

Any point that can be consistently detected in a fingerprint image can be used as a registration point even if in general a point positioned almost at the center of the image is preferable because with high probability it is quite near to the center of the fingerprint. Moreover, in a fingerprint, the core point is a good point because of its robustness. In the FingerCode algorithm the core point is defined as the central point (x_c, y_c) of the fingerprint. A core point detection algorithm is needed to perform this measurement. In the following description we refer to [HWJ98]. The first step is an estimation of the orientation field that can be done using the least square orientation estimation algorithm proposed in [JHB97]. This algorithm uses techniques related to the orientation field and the Poincaré index² to identify the reference point (the core point) in a fingerprint. Figure 4.2 shows two examples of the estimated center positions.



Figure 4.2: *Core Detection.*

When the core is found the image has to be split in sectors. The number of sectors depends on the application and the accuracy we want to obtain. Figure 4.3 shows an example of the division in sectors.

²”The index of a vector field with finitely many zeros on a compact, oriented manifold is the same as the Euler characteristic of the manifold.” [Wei].



Figure 4.3: The reference point x , the region of interest and 80 sectors superimposed on a fingerprint.

Normalization and Filtering of Sectors

Fingerprint images present a strong orientation tendency and have a well-defined spatial frequency in each local neighborhood that does not contain singular point(s) (See Figure 4.4). Gabor filters are band-pass filters which have both *orientation-selective* and *frequency-selective* properties and have optimal joint resolution in both spatial and frequency domains [Dau85]. By applying properly tuned Gabor filters to a fingerprint image, the true ridge and furrow structures can be greatly accentuated. These reinforced ridges and furrow structures, constitute an efficient representation of a fingerprint image. An even symmetric Gabor filter has the following general form in the spatial domain:

$$G(x, y; f, \theta) = \exp \left\{ -\frac{1}{2} \left[\frac{x'^2}{\delta_x^2} + \frac{y'^2}{\delta_y^2} \right] \right\} \cos(2\pi f x'), \quad (4.1)$$

where: $x' = x \sin \theta + y \cos \theta$, $y' = x \cos \theta - y \sin \theta$, f is the frequency of the sinusoidal plane wave along the direction q from the x -axis, and δ_x , δ_y specify the Gaussian envelope along x and y axes, respectively, which determine the bandwidth of the Gabor filter. In general, the filter frequency f is set

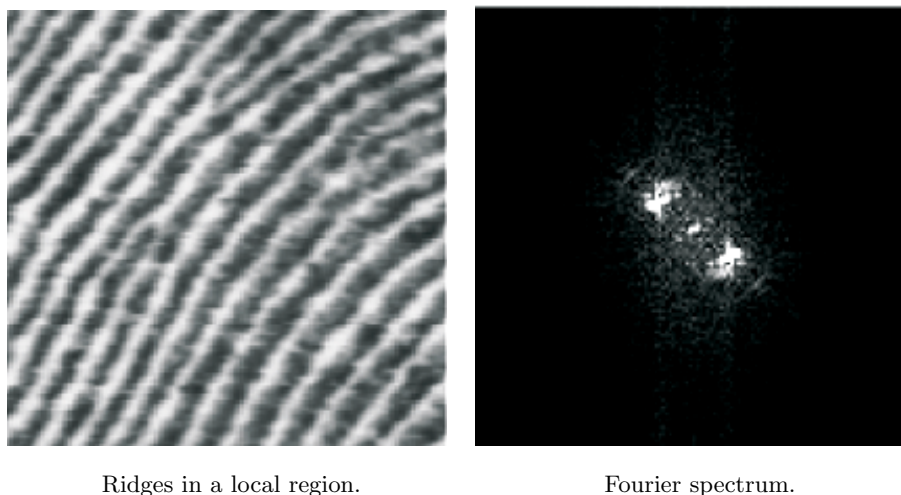


Figure 4.4: *Fingerprints have well-defined local frequency and orientation.*

to the average ridge frequency ($1/K$), where K is the inter-ridge distance. Commonly in a 500 dpi fingerprint image the average inter-ridge distance is approximately 10 pixels. If f is too large, spurious ridges may be created in the filtered image, whereas if f is too small, nearby ridges may be merged into one. The bandwidth of the Gabor filters is determined by δ_x and δ_y . The selection of the values of δ_x and δ_y is based on the following trade-off. If they are too large, the filter is more robust to noise, but is more likely to smooth the image, ridge and furrow details in the fingerprint are lost. On the other hand, if they are too small, the filter is not effective in removing noise, a good choice is $\delta_x = 4.0$ and $\delta_y = 4.0$.

A fingerprint image is decomposed into four component images corresponding to four different values of θ (0° , 45° , 90° , and 135°) with respect to the x -axis (Figure 4.5). A fingerprint image is convolved with each of the four Gabor filters to produce the four component images. Convolution with a 0° -oriented filter accentuates ridges parallel to the x -axis and smoothes ridges which are not parallel to the x -axis. Similarly filters tuned to other directions enhance different directions.

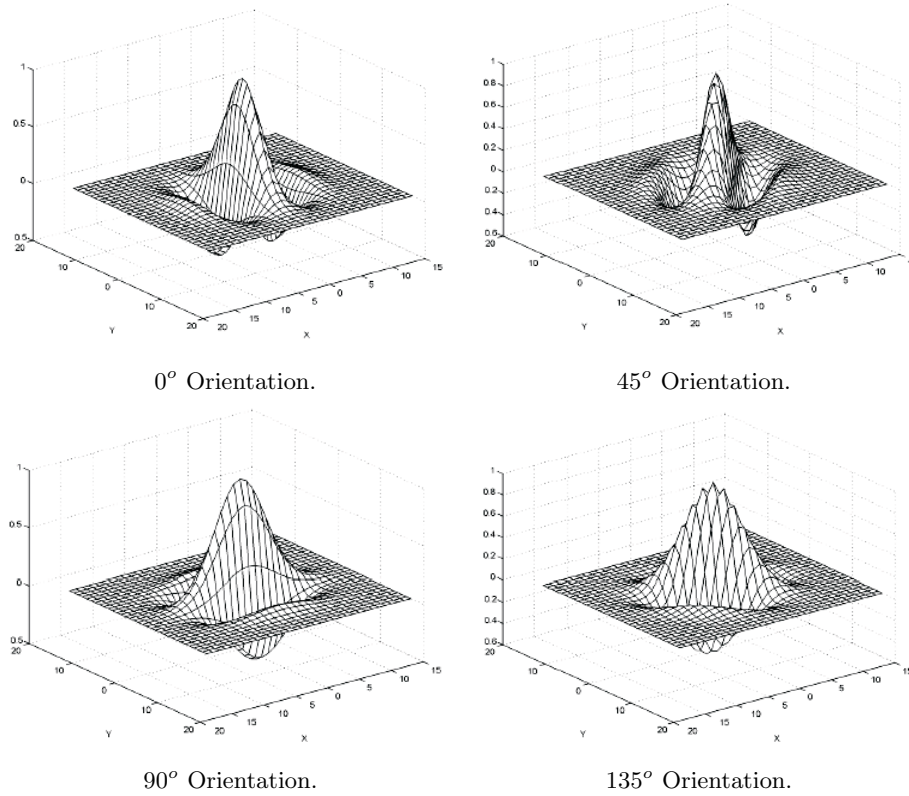


Figure 4.5: Gabor filters (size = 33×33 , $f = 0.1$, $\delta_x = 4.0$, $\delta_y = 4.0$).

The final image is similar to the original one but the ridges have been reinforced in a given direction (Fig. 4.6). Before decomposing the fingerprint image, each sector in the region of interest has to be normalized to obtain a constant mean and variance. The normalization is necessary to remove the effects of sensor noise and differences in finger pressure during measurement. Let $I(x, y)$ denote the gray value at pixel (x, y) , M_i and V_i , the estimated mean and variance of sector S_i , respectively, and $N_i(x, y)$ the normalized gray-level value at pixel (x, y) . For all the pixels in sector S_i , the normalized

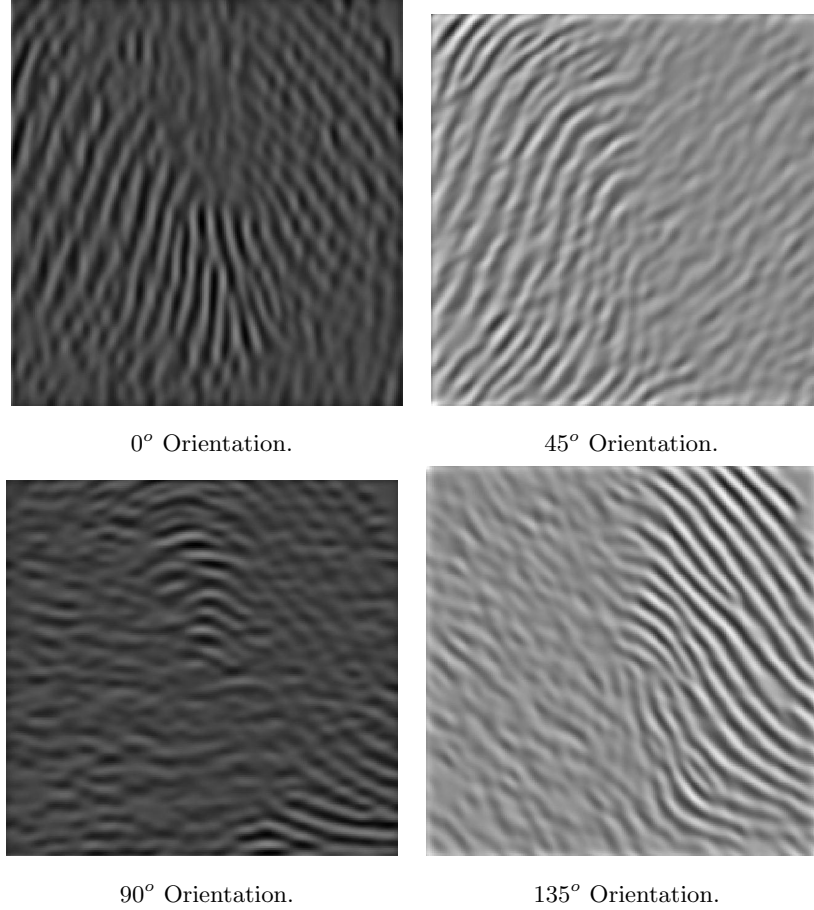


Figure 4.6: Fingerprint after Gabor filtering (size = 33×33 , $f = 0.1$, $\delta_x = 4.0$, $\delta_y = 4.0$).

image is defined as:

$$N_i(x, y) = \begin{cases} M_0 + \sqrt{\frac{(V_0) \times [I(x, y) - M_i]^2}{V_i}}, & \text{if } I(x, y) > M_i \\ M_0 - \sqrt{\frac{(V_0) \times [I(x, y) - M_i]^2}{V_i}}, & \text{otherwise.} \end{cases} \quad (4.2)$$

where M_0 and V_0 are the desired mean and variance values, respectively. Normalization is a pixel-wise operation which does not change the clarity of the ridge and furrow structures. If normalization is carried out on the entire image, then it cannot compensate the intensity variations in the different parts

of the same finger that are due to finger pressure differences, a problem that is solved by normalizing each sector separately.

Generation of the Template

In each component image, a local neighborhood with ridges and furrows that are parallel to the corresponding filter direction exhibits a higher variation, whereas a local neighborhood with ridges and furrows that are not parallel to the corresponding filter tends to be diminished by the filter which results in a lower variation. The spatial distribution of the variations in local neighborhoods of the component images thus constitutes a characterization of the global ridge structures and is captured well by the standard deviation of gray scale values. The standard deviation within the sectors defines the feature vector.

Let $C_{i\theta}(x, y)$ be the component image corresponding to θ for sector S_i . For $\forall i, i = 0, 1, \dots, 47$ and $\theta \in [0^\circ, 45^\circ, 90^\circ, 135^\circ]$, a feature is the standard deviation $F_{i\theta}$ defined as:

$$F_{i\theta} = \sqrt{\sum_{(x,y) \in S_i} (C_{i\theta}(x, y) - M_{i\theta})^2}, \quad (4.3)$$

where $M_{i\theta}$ is the mean of the pixel values in $C_{i\theta}(x, y)$. The resulting 192-dimensional feature vector, also called FingerCode for typical fingerprint images from different classes, is shown as gray level images with four disks, each disk corresponding to one filtered image (see Figure 4.7). The gray level in each sector of a disk represents the feature value for that sector in the corresponding filtered image. Generally speaking we can individuate five kind of fingerprint each defined by the peculiarity of loops, whorls and arches: whorl, right loop, left loop, arch and tented arch. One can see that visually this representation appears to discriminate the five fingerprint classes reasonably well. Six concentric bands around the center point are used. Each band is 20-pixels wide ($b = 20$), and segmented into eight sectors ($k = 8$). The innermost band is not used for feature extraction because the sectors in the region

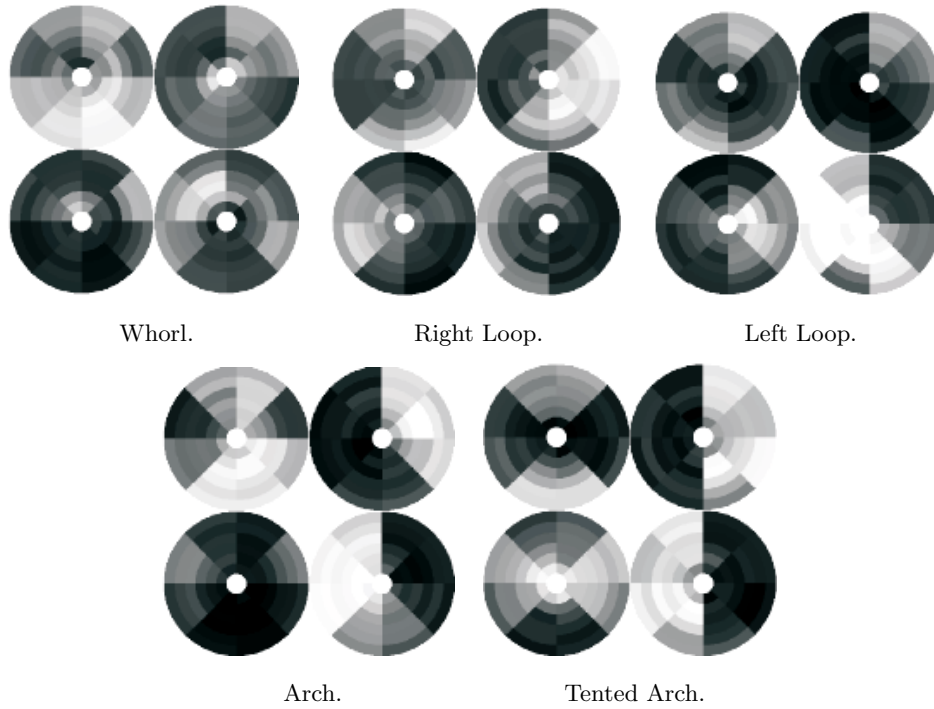


Figure 4.7: For each kind of fingerprint there are four disks that represent various orientations of the Gabor filter (0° , 45° , 90° and 135°). Each disk corresponds to one particular filter and there are 48 features (shown as gray values) in each disk ($8 \times 6 = 48$ sectors) for a total of 192 (48×4) features.

near the center contains very few pixels. Thus, a total of $8 \times 6 = 48$ sectors (S_0 through S_{47}) are used for matching.

4.2.2 Matching

FingerCode matching is based on the Euclidean distance between FingerCodes. Consistency with regard to translation is due by the use of the reference point even if the features are not rotationally invariant. An approximate rotation invariance is achieved by cyclically rotating the features in the FingerCode itself. A one step cyclic rotation of the features in the FingerCode

as described by Equation (4.4) corresponds to the feature vector that would be obtained if the image was rotated by 22.5° . A rotation by R steps corresponds to a $R \times 22.5^\circ$ rotation of the original image. A positive rotation implies clockwise rotation while a negative rotation implies counterclockwise rotation. The FingerCode obtained after R rotation steps is given by:

$$\begin{aligned} V_{i\theta}^R &= V_{i'\theta'} \\ i' &= (i + k' + R) \bmod k' + \left\lfloor \frac{i}{k} \right\rfloor k' \\ \theta' &= (\theta + 180^\circ + 22.5^\circ R) \bmod 180^\circ \end{aligned} \quad (4.4)$$

where $k' = 8$ is the number of sectors, $i \in [0, 1, \dots, 79]$ and the rotations are $\theta \in [0^\circ, 22.5^\circ, 45^\circ, 67.5^\circ, 112.5^\circ, 135^\circ, 157.5^\circ]$. Due to this, for each fingerprint in the database, it is necessary to store five templates corresponding to the following five rotations of the corresponding FingerCode: $V_{i\theta}^{-2}$, $V_{i\theta}^{-1}$, $V_{i\theta}^0$, $V_{i\theta}^1$, and $V_{i\theta}^2$. The input FingerCode is compared with the five templates stored in the database to obtain five different matching scores. The minimum matching score corresponds to the best alignment of the input fingerprint with the database fingerprint.

Some implementations use 10 templates generated with a rotation of the fingerprint equal to 15° . This solution guarantees more robustness against image rotation, but it requires more storage space and computational time.

4.3 The Addressed Scenario

We consider the following scenario: a client, Alice is equipped with a specific-biometric device (a fingerprint reader) and is interested to learn if the acquired fingerprint belongs to the database of authorized entities that is managed by a server, namely Bob. For instance we may consider the following real-world scenario. To access a building (i.e. a bank *cavoux*) the user must provide his biometric, this could happen either touching a device

near the access door or using a portable device that can be connected to an apposite plug near the door (see Fig. 4.8). Another example could be a Police



(a) A door with fingerprint reader. (b) A USB fingerprint reader.
Figure 4.8: Real world devices for fingerprint scanning.

department that given a biometric sample is interest to extract from database managed by a third party all the entries matching with a given threshold.

We require that Alice trusts Bob to correctly perform the matching algorithm for the fingerprint recognition. In addition she should not learn anything about the database managed by Bob, beyond the outcome of the matching process. On the other hand Bob should not get any information about the biometric trait provided. To be more specific, Bob that owns a list of enrolled users allowed to access the service, should only verify if the fingerprint template provided by Alice corresponds to one of the enrolled users, without knowing which particular user is accessing the system.

Our approach to this scenario is somewhat different from those considered in [EFG⁺09] [SSW09], where at the output of the computation the identity associated to the matching face is revealed to the client, and where the best matching face is identified. On the contrary, in the case considered here, the need to look for the best match is avoided thus permitting a significant simplification of the protocol. Our construction allows also to select and report the identifiers of *all* (if more than one are present) the enrolled identities

whose distance to the user’s FingerCode is lower than a given threshold. Thus we propose the following two variants. The first one considers applications where the client is interested only into knowing if the users’s fingerprint is in the database or not (without an identifier). The second one handles the case of a client who wishes to verify if a given claimed identity is in the database and if the just acquired fingerprint matches with such identity.

The developed protocols are entirely based on the use of homomorphic cryptosystems and permit a notable bandwidth saving (about 25 – 39%), if compared with the best previous work [SSW09]. The computational complexity is still quite low and suitable for practical applications. Moreover, even if our protocols are presented in the context of a fingerprint-based system, they can be generalized to any biometric system that shares the same matching methodology, namely distance computation and thresholding.

In our scenario Alice owns a pair of (matching) keys PuK and PrK for a public-key cryptosystem. We assume that Bob has a certified copy of PuK ³.

As in [EFG⁺09, SSW09], our solutions adopt the following three basic steps:

Vector Extraction: on a first stage the target biometric (i.e. the information acquired by the biometric device) is *converted* to a quantized characteristic feature vector \bar{x} ; in our specific case, the fingerprint image is processed as described in Section 4.2 in order to extract the FingerCode vector; similar processes are available in literature for other biometric systems (i.e. the iriscodes [Dau06]);

Distance Computation: the distances (according to some appropriate metric) between the target vector \bar{x} and the vectors corresponding to each ID in the database are computed; in our case, we are going to use the Euclidean distance as required by the FingerCode system;

³Jumping ahead, the protocols require the use of two different encryption schemes (Paillier and EC-ElGamal, see below): to simplify the presentation we are assuming the PuK contains the different public-keys of the required cryptosystems.

Selection of the Matching Identities: one (or more) IDs matching the target ID are selected.

In particular regarding the last step, we slightly change the original semantic of the problem: instead of querying about the nearest matching enrolled identity in the database as mentioned before, we are interested in getting all the matching enrolled identities. In other words: the required outcome for Alice is the list of all the identities in the database whose characteristic feature vectors are *near enough* to be considered a successful match (i.e., the distance is lower than a general threshold τ). With some biometric systems, if we assume well-chosen parameters (like the threshold τ), one may assume that a measure of a specific biometric sample matches with just one person: the owner of that biometric trait⁴. If, for some application-related reasons, the same person is enrolled in the database more than once, it should be fine to return all these identities to the client. However, for specific biometric systems or applications, it could not be equivalent and/or desirable.

4.3.1 Security Analysis

In this Section we show the main differences between the standard scenario and the privacy preserving one. For the sake of simplicity we recall the scenario described in the previous Section as a sequence of actions focusing on the most important ones.

In the standard scenario Bob is the trusted authority, so during the enrollment Alice asks Bob to be enrolled in the system, if everything is fine Bob stores in his database the FingerCode computed using the fingerprint provided by Alice. In this case Alice must trust Bob because he can access the biometric trait and also the template, additionally Bob is in charge to protect the database from third parties attacks. In the matching phase, Alice uses a

⁴Often to improve the performances of the algorithms, for every user several rotations of the same fingerprint are stored in the database. This is done to achieve a better accuracy during the matching phase.

finger-scanner to compute a plain version of her FingerCode. Note that the finger-scanner owns to Bob and so Alice must be confident with the device and Bob must assure that the matching procedure is protected against third parties and correctly computed. Bob manages the database and the fingerprint. The database contains plain data: the enrolled templates. The fresh fingerprint (used in the matching phase) is not encrypted.

In the privacy preserving scenario the enrollment phase is identical to the standard case. Bob is required only to protect the database, but Alice is not forced to trust Bob during the matching phase. Alice generates her keys PuK and PrK and she uses a finger-scanner that requires also the PuK to produce the encrypted template. Bob and Alice together apply a privacy preserving protocol to decide if Alice can be allowed to the system. Bob manages the database with the plain templates. During the matching phase each user manages his cryptographic keys, the fingerprints (the enrolled the fresh one) and the encrypted templates.

4.4 Parameters and Model

We will denote the symmetric security parameter by t and the asymmetric one (i.e., bit-length of RSA moduli) as usual by s . Recommended parameters for short-term security are $t = 80$ and $s = 1024$, whereas for long-term security $t = 128$ and $s = 3072$ are recommended [BBJ⁺09].

In all the scenarios we consider a server Bob with a database of n enrolled entities, where each of them is represented by a characteristic FingerCode of k components each of λ -bits integers. We will denote with τ the biometric-threshold that, given a specific metric, allows to say if two biometric measures match or not. In order to support the specific matching logic on the FingerCode, we will assume that for each enrolled identity $m = 5$ different vectors⁵

⁵Additional templates related with rotated fingerprint (Section 4.2).

are stored in the database as well as an eventual identity-specific thresholds⁶ τ^i .

The values of the k , λ and m parameters must be tuned according to the current fingerprint dataset. For example, working with a dataset of $n = 900$ fingerprints captured with a standard fingerprint sensor, a proper configuration is the following: 2 – 5 concentric bands, 4 – 16 sectors, 2 – 8 Gabor filters, quantized with 4 – 8 bits and stored with five different orientations ($k = 16 - 640$, $\lambda = 4 - 8$ and $m = 5$). Typical bit lengths of the FingerCode range from 64 to 5120 bits.

Finally, we work in the *honest but curious* model as common in privacy preserving applications, where parties are assumed to follow the protocol but may try to learn additional information from the protocol trace beyond what can be derived from the inputs and outputs of the algorithm when used as a black-box (Section 2.2.2).

4.5 Basic Building Blocks

In addition to the Paillier cryptosystem we will use a couple of sub-protocols.

4.5.1 The sub-protocol BitMin

In this section we introduce the sub-protocol BitMin used in the following Section 4.6, which is a variant of the protocol proposed in [EFG⁺09]. As usual: we consider a client Alice and a server Bob. The latter has got the encryption of two λ -bit integers $\llbracket X \rrbracket$ and $\llbracket Y \rrbracket$ ⁷.

⁶Sometimes and for specific application it is possible to personalize the sensitivity of the system by setting several user-defined threshold. In general, this approach improves the performances of the system even if the choice of the thresholds could not be an easy task.

⁷We assume that encrypted input and output values of this protocol are computed using the Paillier public-key of Alice. Furthermore we note that, in the context of our protocol, BitMin is usually applied on inputs with a bit-length of $\lambda' = 2\lambda + \lceil \log_2 k \rceil + 1$. In this section

The protocol **BitMin** allows Bob to compute the encrypted bit $\llbracket b \rrbracket$ such that:

$$b = \begin{cases} 0 & \text{if } X < Y \\ 1 & \text{if } X \geq Y. \end{cases} \quad (4.5)$$

and uses as building block a variant of the comparison protocol proposed in [DGK07] and recalled later in this Section. The protocol **BitMin** is given in Figure 4.9 and works as follows.

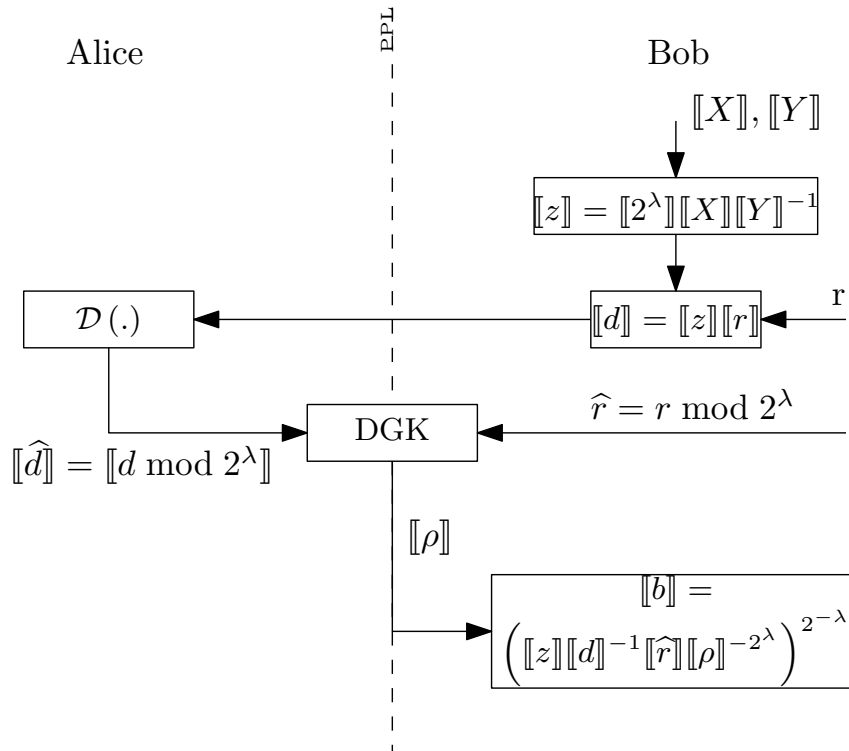


Figure 4.9: *The Protocol BitMin.*

As a first step the server homomorphically computes $\llbracket z \rrbracket = \llbracket 2^\lambda + X - Y \rrbracket$. Since X and Y are λ -bit long, z is an $\lambda + 1$ -bit integer. Moreover one can interestingly see that the most significant bit of z (which we denote z_λ) is 0

we assume λ -bit integers in order to simplify the protocol description.

if and only if $X < Y$. Thus in order to learn if $X < Y$ it suffices to compute z_λ . This can be done as follows.

Bob additively blinds z with a suitable random value r , obtaining $\llbracket d \rrbracket$, he sends $\llbracket d \rrbracket$ to Alice, then they run the sub-protocol DGK after which Bob will learn $\llbracket \rho \rrbracket$ such that $\rho = 0 \Leftrightarrow \hat{d} < \hat{r}$ (where \hat{d} and \hat{r} are, respectively, $d \bmod 2^\lambda$ and $r \bmod 2^\lambda$). We notice that the information about $\hat{d} < \hat{r}$ is useful to compute z_λ . In fact observe that:

$$b = z_\lambda = 2^{-\lambda}(z - \hat{z}) = 2^{-\lambda}(z - ((d - r) \bmod 2^\lambda)) \quad (4.6)$$

where it is possible to compute $(d - r) \bmod 2^\lambda = (d \bmod 2^\lambda) - (r \bmod 2^\lambda) + \rho \cdot 2^\lambda$. Since $\rho = 0 \Leftrightarrow \hat{d} < \hat{r}$ it is easy to see the correctness of z_λ .

In the rest of the paragraph we use some results about the DGK protocol that will be detailed later in this Section. The **BitMin** is a simple protocol that requires a number of rounds equal to 4: 1 to exchange the result and 3 due to DGK protocol. Only 1 ciphertext is sent from Bob to Alice, so the bandwidth is a Paillier ciphertext plus the bandwidth of DGK, thus: $\ell + \frac{2\lambda s}{3} + 1$, we recall that $\ell = 2s$, so: $\ell \left(1 + \frac{\lambda}{3}\right) + 1$. Finally the number of bit operations is the sum of: 3 **mult** + 1 **exp** to compute $\llbracket d \rrbracket$, 1 **dec** + 1 **enc** to obtain $\llbracket \hat{d} \rrbracket$ and 4 **mult** + 3 **exp** to reach $\llbracket b \rrbracket$; that is 7 **mult** + 6 **exp**. Considering that the exponentiations are the most complex operations and recalling that DGK exponentiations are 4λ we have: $(6 + 4\lambda)$ **exp**.

Table 4.1: Computational Complexities – *BitMin* sub-protocol.

#exp	Bandwidth	Rounds
$6 + 4\lambda$	$\ell \left(1 + \frac{\lambda}{3}\right) + 1$	4

The sub-protocol DGK

The DGK comparison protocol of [DGK07] allows both parties (i.e. the client Alice and the server Bob) to learn the bit ρ of the predicate $d < r$ where d and r are two λ -bit integers owned by Alice and Bob respectively. The original DGK protocol is given in Figure 4.10 and works as follows.

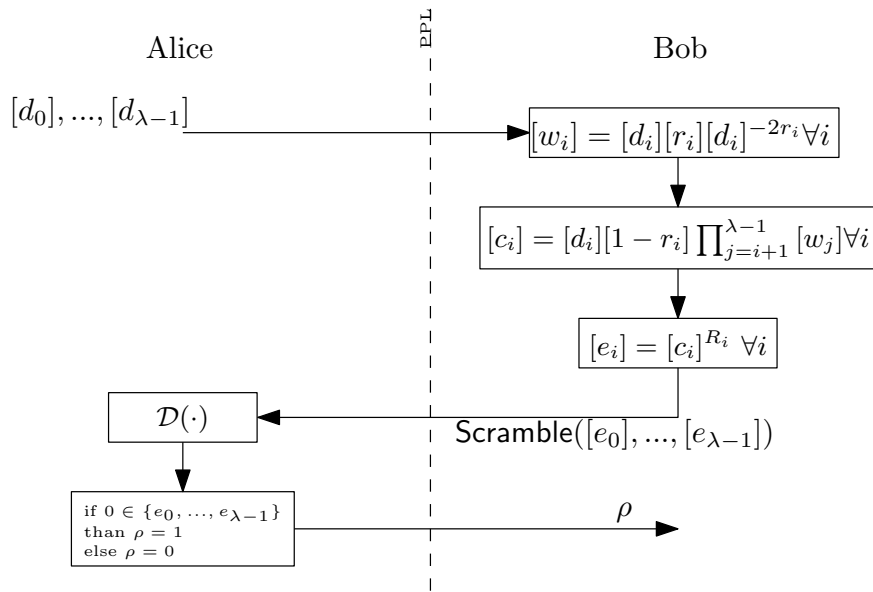


Figure 4.10: *The sub-protocol DGK.*

As in the other protocols, Alice has a pair of keys PuK and PrK for an additively homomorphic cryptosystem: the original protocol uses the DGK [DGK07] cryptosystem, we will use a different scheme as stated later. We are going to use another notation for such ciphertexts: $[x]$. In particular, the chosen cryptosystem is a known variant of the well-known ElGamal cryptosystem [ElG85]. Such a scheme differs from the original in two points: it is additively homomorphic and all the computation is carried over a suitably chosen EC. The use of EC allows to obtain a great bandwidth saving, indeed, exploiting the point compression [CCMR06], the ciphertext can be transmitted using $2 \cdot (2t + 1)$

bits. For example, for a security parameter $t = 80$, the ciphertext sizes for the considered cryptosystems would be: Paillier 2048 bits, DGK [DJ01] 1024 bits and EC-ElGamal 322 bits. The inputs for the parties are, respectively, an λ -bits integer d for the client and another λ -bit integer r for the server.

After the original protocol has been run, the server (as well as the client) will learn the decision bit ρ related with the predicate $d < r$ (i.e. $\lambda = 0 \Leftrightarrow d < r$) while d and r will remain hidden to Bob and Alice respectively. In our `BitMIn` we use a slightly different version of this protocol where Alice sends $\llbracket \rho \rrbracket$ (encrypted with his Paillier public-key) instead of ρ : in this way the value of the decision bit remains hidden to the server.

The protocol consists of three rounds during which 2λ ciphertexts are exchanged. More in detail, Bob computes the values $[w_i] = [d_i \oplus r_i]$ and $[c_i] = [d_i - r_i + 1 + \sum_{j=i+1}^{\lambda-1} w_j]$. The values c_i carry the information whether or not $d < r$, in particular we have that one of the c_i 's will be 0 if and only if $d < r$. To see the correctness of this, consider all possible cases. If $d = r$, then we clearly have $c_i = 1$ for all $i = 0, \dots, \lambda - 1$. If $d \neq r$, assume that the m -th bit (starting from the most significant) is the first one where they differ. Then $c_{\lambda-1}, \dots, c_{m+1}$ are equal to 1 while $c_m = d_m - r_m + 1$ (as $\sum_{j=m+1}^{\lambda-1} w_j = 0$). Moreover since $w_m = 1$, we have $\sum_{j=i+1}^{\lambda-1} w_j \geq 1$ and $c_i \geq 1 \forall i \in \{0, \dots, m-1\}$. Thus c_m depends only on d_m and r_m and it will be 0 only if $d_m < r_m$.

Finally, since c_i 's might contain information about d and r , they are randomized (creating e_i) so that when Alice decrypts e_i she will obtain either 0 (if $c_i = 0$) or a random value⁸. Therefore Alice will set $\rho = 0$ if one of the decrypted e_i 's is equal to 0.

We now briefly give some details about the complexities involved in this sub-protocol. The DGK construction require 3 rounds in which 2λ cipher-

⁸It is sufficient to check if the plaintext is equal to 0: the DGK cryptosystem, as well as the one that we adopt, has a decryption procedure that is based on an exhaustive search in the plaintext space.

texts⁹ and 1 plaintext that correspond to 1 bit are exchanged. From the point of view of computational complexity (Chapter 2), the protocol requires to compute $\lambda \text{ enc} + \lambda \text{ dec}$, $2\lambda \text{ mult} + \lambda \text{ exp}$ to compute all $[w_i]$, $\left(2\lambda + \frac{\lambda(\lambda+1)}{2}\right) \text{ mult}$ to compute $[c_i]$ and $\lambda \text{ exp}$ to compute $[e_i]$. Thus we have a total of: $(4.5\lambda + 0.5\lambda^2) \text{ mult} + 4\lambda \text{ exp}$. Keeping in mind that the most expensive bit operation is the exponentiation we see that the computational complexity is equal to $4\lambda \text{ exp}$. The above results are shown in Table 4.2.

Table 4.2: Computational Complexities – DGK sub-protocol.

#exp	Bandwidth	Rounds
4λ	$\frac{2\lambda s}{3} + 1$	3

4.6 The FingerCode Matching Protocol

During a preliminary phase the acquired fingerprint image is converted into a FingerCode vector. We assume that this phase is done in clear (i.e. not in the encrypted domain) by Alice. Notice that this is not an issue in our honest but curious setting where the client (i.e. the biometric device) *already* has the fingerprint data. Moreover, given our current state of knowledge, such an assumption seems to be necessary for our protocol to be practical. Indeed, for many biometric systems (e.g. fingerprint, iris,...) the analysis of the biometric measures, and their corresponding quantization process, are too complex to be carried out efficiently in the encrypted domain.

As stated above, we assume that Alice has already processed the fingerprint image to get a characteristic feature vector (FingerCode) \bar{x} . On the other side, Bob manages a database of n pairs (id^i, \bar{y}^i) , where id^i is a unique numeric identifier associated to the specific enrolled identity and \bar{y}^i is the

⁹Note the a size of EC-ElGamal is $\frac{s}{3}$ bit.

related precomputed FingerCode. Our solution requires the use of specific values for these identifiers: $id^i = 2^i$ (powers of 2). In this phase we deliberately ignore some technical details that are FingerCode-specific. In particular we do not consider here the presence of m different FingerCodes for each identity and the use of identity-specific thresholds τ^i . In this way we get a more general protocol that could be used for other biometric systems as well. Later we discuss those specific aspects.

Alice sends element-by-element encryption of the integer vector \bar{x} to Bob: more specifically, k Paillier encryptions $\llbracket x_0 \rrbracket, \dots, \llbracket x_{k-1} \rrbracket$ jointly with a further encryption $\llbracket \sum_{j=0}^{k-1} x_j^2 \rrbracket$, this value will be used to complete the computation of the distances in the ciphertexts domain as described later.

The FingerCode system, as well as other biometric systems, uses the Euclidean distance as underlying metric. In particular we consider squared distance to reduce the complexity of the protocol¹⁰. Denoting with D^i the square of the Euclidean distance between \bar{x} and the stored vector \bar{y}^i , the server can non-interactively compute $\llbracket D^i \rrbracket$ by exploiting the homomorphic properties of the Paillier cryptosystem, its knowledge¹¹ of \bar{y}^i and the ciphertexts received by Alice as follows:

$$\begin{aligned}
\llbracket D^i \rrbracket &= \left\llbracket \sum_{j=0}^{k-1} (x_j - y_j^i)^2 \right\rrbracket = \\
&= \left\llbracket \sum_{j=0}^{k-1} x_j^2 \right\rrbracket \cdot \left\llbracket -2 \sum_{j=0}^{k-1} x_j y_j^i \right\rrbracket \cdot \left\llbracket \sum_{j=0}^{k-1} (y_j^i)^2 \right\rrbracket = \\
&= \left\llbracket \sum_{j=0}^{k-1} x_j^2 \right\rrbracket \cdot \prod_{j=0}^{k-1} \llbracket x_j \rrbracket^{-2y_j^i} \cdot \left\llbracket \sum_{j=0}^{k-1} (y_j^i)^2 \right\rrbracket \quad \forall i \in \{1, \dots, n\} \quad (4.7)
\end{aligned}$$

¹⁰We are of course using the fact that the square function is monotonically increasing function on positive inputs. We implicitly assume that the threshold values τ, τ^i are properly adapted to accommodate this.

¹¹Bob knows the plain version of the enrolled biometric templates.

To realize the identity selection we use the sub-protocol **BitMin** introduced in Section 4.5.1. We briefly recall that this sub-protocol allows to obliviously compute an encryption of the binary predicate $X < Y$: i.e. $\llbracket b \rrbracket = \text{BitMin}(\llbracket X \rrbracket, \llbracket Y \rrbracket)$ where:

$$b = \begin{cases} 0 & \text{if } X < Y \\ 1 & \text{if } X \geq Y. \end{cases} \quad (4.8)$$

Once all the distances have been computed, Bob gets the distances vector: $\llbracket D^1 \rrbracket, \dots, \llbracket D^n \rrbracket$. Let's consider the following set of n pairs identity - distance: $(id^1, \llbracket D^1 \rrbracket), \dots, (id^n, \llbracket D^n \rrbracket)$. He randomly permutes¹² these pairs and obtains: $(id^{j_1}, \llbracket D^{j_1} \rrbracket), \dots, (id^{j_n}, \llbracket D^{j_n} \rrbracket)$ and then he computes, using parallel executions of **BitMin**, the values $\llbracket b^{j_i} \rrbracket = \text{bit-MIN}(\llbracket \tau \rrbracket, \llbracket D^{j_i} \rrbracket)$ for $i \in \{1, \dots, n\}$.

Finally, Bob computes and returns to Alice the following encrypted value:

$$\llbracket R \rrbracket = \left[\sum_{i=1}^n b^i \cdot id^i \right] = \prod_{i=1}^n \llbracket b^i \rrbracket^{id^i}. \quad (4.9)$$

Due to the fact that: $b^i = 1 \Leftrightarrow D^i < \tau$ for $i \in \{1, \dots, n\}$, it is easy to check that the final value R will consist of the sum of the numeric identifiers associated to the enrolled identities that match the target biometric. In other words: the bit at position i in R is set to 1 if and only if the i -th identity matches. Now, Alice can easily extract R and reconstruct the list of matching identities. The complete protocol flow is shown in Figure 4.11.

The construction in Figure 4.11 strictly requires the use of powers of 2 as identifiers: in real application scenarios with a large number of enrolled people this fact could limit its scalability. Indeed, the maximum number of different identifiers is equal to the bit-length of the Paillier plaintext (T). For a security level of $t = 128$, we can handle at most $s = 3072$ different identifiers. This can be handled clustering the identifiers for which $n > s$ and using multiple

¹²The randomization is only used to hide to Alice the relation among the positions of the identities in the DB and the order of querying. It is strictly required a fresh permutation at each new session.

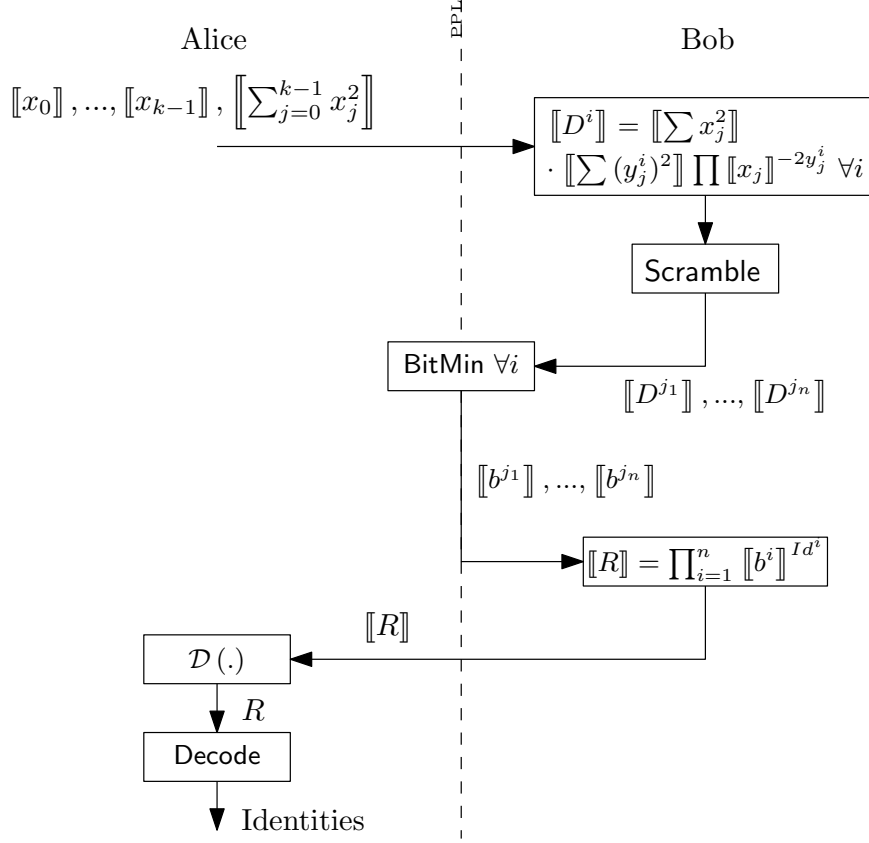


Figure 4.11: Privacy Preserving FingerCode Matching.

outcomes R_j ; more specifically, for $j = 0, \dots, \lceil \frac{n}{s} \rceil - 1$ the server computes $\llbracket R_j \rrbracket = \prod_{i=1}^{n_j} \llbracket b^{js+i} \rrbracket^{2^i}$, where $n_0 = s, n_1 = s, \dots, n_{\lceil \frac{n}{s} \rceil - 1} = n \bmod s$ are the cluster cardinalities. In this way the i -th bit in R_j is associated to the identity id^{js+i} . The ciphertexts $\llbracket R_j \rrbracket$ are sent to Alice in the last round of the protocol. These changes on the protocol do not imply any further leakage of information.

4.6.1 Variant for Simple Authentication.

As stated at the beginning of Section 4.3 in the scenario we are interested in the client needs a simple boolean outcome, *like authenticated/rejected*.

In order to do so, it is sufficient to change the way the value $\llbracket R \rrbracket$ is computed as follows:

$$\llbracket R \rrbracket = r \cdot \left[\sum_{i=1}^n b^i \right] = \left(\prod_{i=1}^n \llbracket b^i \rrbracket \right)^r \quad (4.10)$$

where r is a fresh random integer. Alice will obtain as output *rejected* if $R = 0$, *authenticated* otherwise.

4.6.2 Variant for Authentication with Identity Confirmation.

Let's think about the following high-security authentication scenario where we want to confirm the identity without revealing the biometric sample: the person who is going to be authenticated is double checked through some kind of hardware token (or a simple card with a bar-code) and some specific biometric (e.g., FingerCode). In this case Alice (the biometric reader) is able to send to Bob an alleged identity \hat{id} read from the hardware token. The final boolean outcome will be positive (*authenticated*) if and only if the submitted biometric matches one of the enrolled identities as well as the alleged identity \hat{id} .

This variant of the protocol is shown in Figure 4.12.

After the computation of the encrypted distances $\llbracket D^i \rrbracket$, the server will compute the auxiliary values:

$$\llbracket m^i \rrbracket = \llbracket r^i \cdot (\hat{id} - id^i) \rrbracket = \left(\llbracket \hat{id} \rrbracket \cdot \llbracket id^i \rrbracket^{-1} \right)^{r^i} \quad (4.11)$$

where r^i are fresh random integers. All the values m^i will be different from zero except for the alleged identity \hat{id} .

The values $\llbracket m^i \rrbracket$ will be sent to Alice during the executions of the sub-protocol BitMin: Bob will return the exact outcome $\llbracket b^i \rrbracket$ of BitMin only if the

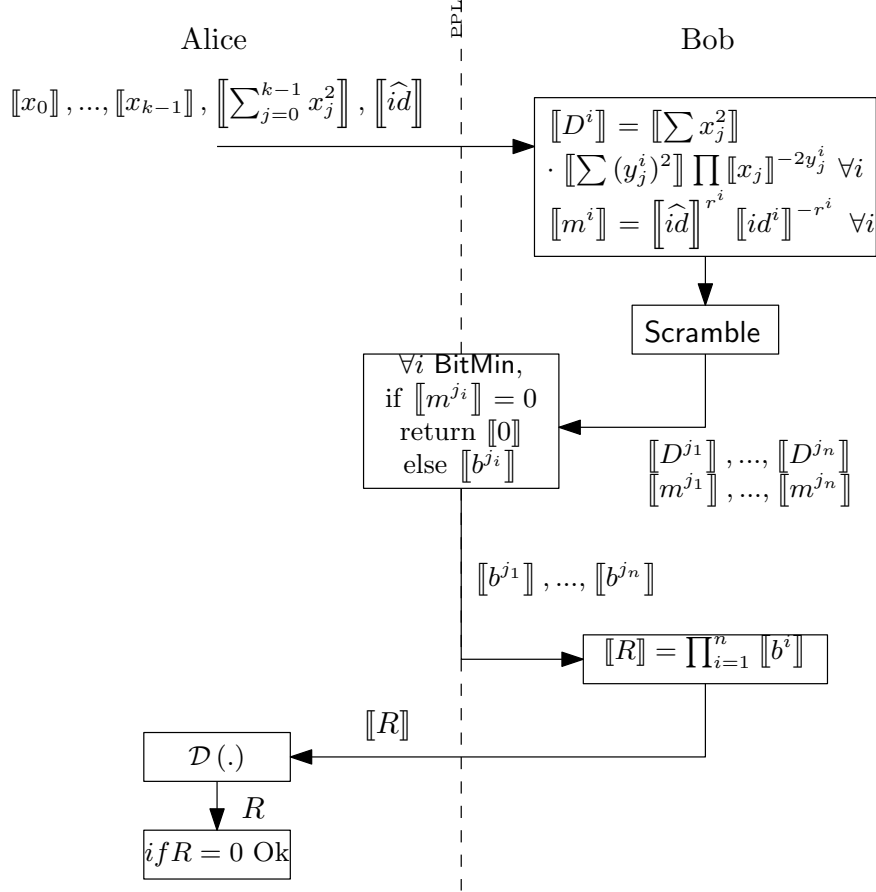


Figure 4.12: Privacy Preserving FingerCode - Identity Confirmation.

corresponding m^i is not null, otherwise a dummy outcome $\llbracket 0 \rrbracket$ is sent. In this way only a single bit b^i can be non-null and only if it matches the identity \hat{id} .

4.7 Security

We now sketch a security argument for our protocol. In particular we want to argue that, in the honest but curious setting, no party should be able to get any information about the other party's input. In other words,

this means that the client Alice should not be able to get anything about the database held by Bob (beyond what revealed by the functionality implemented by the protocol) whereas Bob should not get anything about the FingerCode and outcome of the authentication process.

We discuss each phase of the protocol separately. The vector extraction phase is done entirely by Alice so no information is leaked to Bob. Security of the distance computation phase can be proved easily following the same approach used in [EFG⁺09] (recall that our distance computation protocol is the same). It remains to discuss the selection of the matching identities phase.

Intuitively it is clear that the protocol is private for the server as all the messages it receives are encrypted with respect to Alice's public key (using a semantically secure cryptosystem). Things are a bit trickier for Alice as she knows the private key corresponding to the public key with respect to which the ciphertexts are created. Still, we argue that this does not allow Alice to get more information than what prescribed by the protocol. This is because, whenever Alice receives a ciphertext, the encrypted message is altered by Bob via an information theoretic secure mask. For instance, in the `BitMin` protocol Alice receives an encryption of d which is statistically indistinguishable from a uniformly distributed $101 + \ell$ random integer. As a final note we point out that even though Alice gets b^{13} in the clear, at the end of the protocol `BitMin` this is not an hazard (in an honest but curious setting) because of the fact that all the couples are randomly permuted by Bob before executing the `BitMin` protocol.

¹³More specifically, Alice does not directly get b but instead a related information: the bit λ .

4.8 Complexities

We consider a scenario where Alice has an already-computed vector \bar{x} with k components of λ -bit integers and Bob manages a DB with n identities. Moreover we refer to the protocol shown in Figure 4.11.

We start considering the number of rounds required to compute the privacy preserving FingerCode. Initially Alice sends $k + 1$ encryptions to Bob, than there are n calls to `BitMin` and then Bob sends the result to Alice (only 1 encryption). Thus we have 2 rounds plus $4n$ due to `BitMin`. Similarly for the bandwidth we have $k + 1$ encryptions sent by Alice, n encryptions in input to `BitMin`, n times the bandwidth required by `BitMin` and 1 encryption for the final result; with a total bandwidth of; $(k + n + 2)\ell + n\ell(1 + \frac{\lambda}{3}) + n$. As last step we consider the number of operations, in particular the number of exponentiations. The protocol requires $(k + 1)$ `enc` and 1 `dec`, additionally to compute all $\llbracket D^i \rrbracket$ nk `exp` are required and n `exp` to compute $\llbracket R \rrbracket$. Finally we need to add up the `BitMin` number of exponentiations that are $6 + 4\lambda$ times n calls. Summarizing we have: $(k + 2 + nk + n + n(6 + 4\lambda))$ `exp`. Table 4.3 summarizes the computational complexities involved.

Table 4.3: Computational Complexities – Privacy Preserving FingerCode.

#exp	Bandwidth	Rounds
$n(k + 4\lambda + 7) + k + 2$	$k\ell + 2\ell + n\ell(2 + \frac{\lambda}{3}) + n$	$4n + 2$

4.9 Real World Implementation

We tested the privacy preserving FingerCode by using a well known public fingerprint dataset composed by 408 grayscale fingerprint images acquired by a CrossMatch Verifier 300 sensor [cro] [pDa]. The dataset contains 8 images for each subject with a resolution equal to 500 dpi and the

dimension of 512×480 pixels. Figure 4.13 shows two examples of images of the test database.



Figure 4.13: Examples of test images.

The application of the individual threshold method cited in Section 4.4 on the testing dataset resulted in a relevant enhancement of the accuracy. Figure 4.14 plots the individual thresholds and the results obtained by using the *human selection*¹⁴ method and *Poincare* method for the estimation of the reference point with a set of 640 fingerprints. Results show that the accuracy is effectively improved. For example, the Equal Error Rate (EER) of Poincare method has been reduced by 4% with respect to the initial value. This method can typically produce relevant enhancement in overall accuracy when the samples belonging to dataset do not have the same quality level. This is the case of the proposed test dataset.

As a second step, in order to test the effect of the number of features on the FingerCode template, we generated a total of eight different configurations, corresponding to eight sets of FingerCode vectors with length ranging from 640 features (the original configuration) to 8 features. This is one of the

¹⁴This is human based procedure in which an expert is asked to identify manually the core point in a fingerprint.

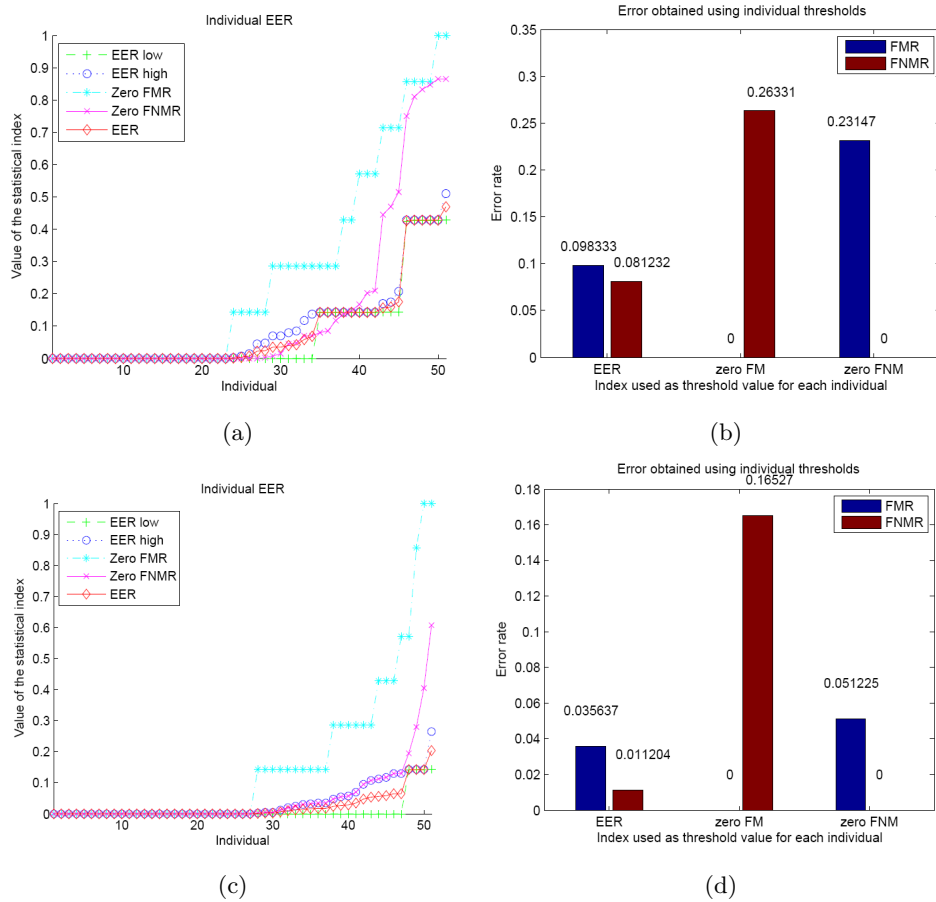


Figure 4.14: Results obtained: (a) and (c) the individual thresholds processed for both methods human selection and Poincare respectively; (b) and (d) the final False Match Rate (FMR) and False Non-Match Rate (FNMR) obtained.

most important approaches to reduce the complexity of the protocol. With this study it was possible to identify the most convenient configuration from the point of view of accuracy and complexity (specifically the bandwidth) of the system. The parameters of the reduced tessellations for each configuration are detailed in Table 4.4.

For investigating the effects of data quantization, each configuration has

Table 4.4: Computational Complexities – Privacy Preserving FingerCode.

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

been normalized and quantized using a different number of bits, ranging from eight bits to a single bit, producing a total of $5 \times 8 = 40$ quantized configurations. The behavior of the Equal Error Rate¹⁵ for the testing dataset is shown in Figure 4.15. It is evident that the performance of the system are practically unaffected by the feature size reduction when the number of features is above 96 and the number of bits is above 2. This suggested to consider for further testing only the configurations C and D, both quantized with 4 and 2 bits.

To evaluate the performance in terms of bandwidth and computational complexity we implemented a client-server prototype version of our protocol written in C++, using the GMP Library [Gra] and the PBC Library [Lyn]. The experimental results were obtained on a PC with 2.4 GHz Intel processor and 4 GB of RAM.

The results show that the proposed method based on FingerCode templates and homomorphic encryption is recommendable in the cases when the

¹⁵A biometric security system predetermines the threshold values for its false acceptance rate and its false rejection rate, and when the rates are equal, the common value is referred to as the equal error rate.

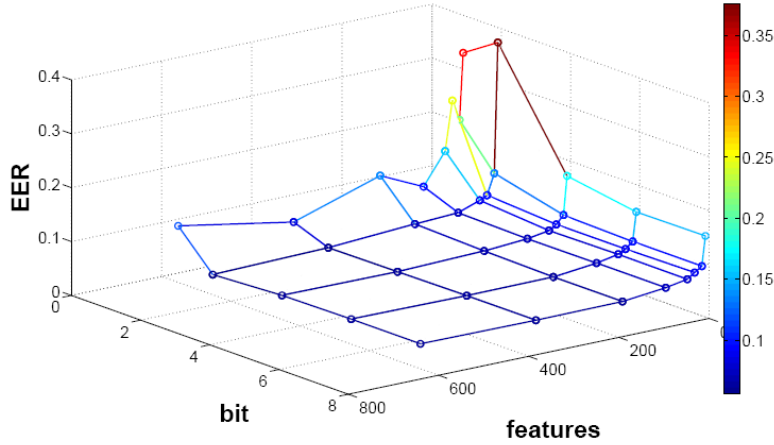


Figure 4.15: Equal error rate of the different configurations.

privacy of the data is crucial and high accuracy of the system is required, in fact the obtained performances on accuracy measured as Equal Error Rate are comparable to the original method. Table 4.5 shows the obtained accuracy and the bandwidth required by the configurations C and D described in Table 4.4, each quantized with 2 and 4 bits.

We estimated the time required for the identification in the encrypted domain by using a dataset composed by 100 enrolled individuals using 80 bits security key. Table 4.6 reports the obtained results measured in seconds. As expected the time complexity of the underlying protocol is linear in the number of enrolled identities.

As shown in Table 4.5 and Table 4.6 different performance can be obtained by varying the number of features of the template and the number of bits used for representing each value. On the other hand, the best computational performance are obtained with a small number of features and bits.

Figure 4.16 plots the ROC curves of the configurations that we consider to be a good trade off. The performance of the different configurations are very close each other, the effects of both feature reduction and quantization

Table 4.5: Performance of the proposed method with a database of 408 entries (3672 feature vectors).

Parameters				
Configuration	Quantization	Security	EER	#bit
C	2	80	0.07577	6568792
		112		10824021
		128		14374232
C	4	80	0.07321	7802584
		112		12527832
		128		16313048
D	2	80	0.071465	6902008
		112		11299320
		128		14932856
D	4	80	0.067324	8135800
		112		13003128
		128		16871672

Table 4.6: Required time for the identification in the encrypted domain using a dataset composed by 100 enrolled entries using 80 bits security key.

Configuration	Quantization	Time (s)
C	2	44.43
	4	53.66
D	2	37.43
	4	45.58

being very limited on the accuracy of the system. It is worth noting that the original configuration, i.e., 640 features with floating point implementation, reported an EER of 0.065333 on the testing dataset, which is comparable

with the performance of the tested configurations. All the reported results are obtained using the human selection method for the estimation of the reference point in the fingerprint images and using a unique threshold for all the users τ . The final EER of the system is only slightly worse than the EER of the original FingerCode technique applied on the same dataset, proving that the privacy protection implementation we proposed is feasible.

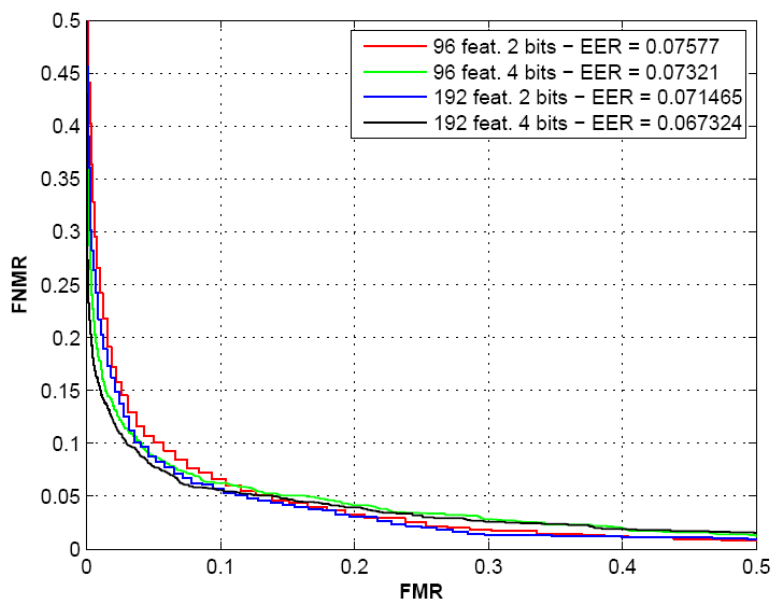


Figure 4.16: ROC curves of the configurations of the proposed method that we consider as the best suitable in real applicative conditions.

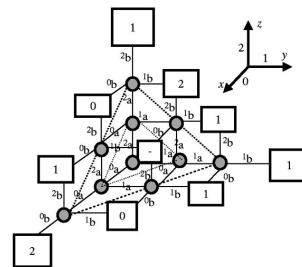
4.10 Summary

In this Chapter we introduced the privacy preserving version of the FingerCode algorithm. We considered a scenario where Alice is equipped with a fingerprint reader and is interested in learning if the acquired fingerprint belongs to the database of authorized entities managed by a server. For security, it is required that Alice does not learn anything on the database and Bob

does not get any information about the requested biometric. The proposed protocol follows a multi party computation approach and makes extensive use of homomorphic encryption as underlying cryptographic primitive.

Chapter 5

Privacy Preserving Sketch



*God exists since mathematics is consistent,
and the Devil exists since we cannot prove it.*
(Andre Weil)

This chapter is devoted to design and define the encrypted version of the Fuzzy Commitment Scheme. Respect to the FingerCode described in Chapter 4 this protocol achieves the goal of protecting also the database, in fact this is a case of encrypted query on encrypted database. Additionally this construction allows the revocability of the enrolled biometric sample.

5.1 Introduction

As already said, a biometric system may serve one of two basic purposes: *authentication/verification* or *identification*. *Authentication* (or *verification*) is the process of positively verifying the identity of a client. *Identification*, on the other hand, is the process of distinguishing an individual from

a larger set of individual records by comparing the presented biometric data with all the entries in the database [O’G03].

Biometric features of individuals are tightly bound with their identities. Moreover, they cannot be easily forgotten or lost. Therefore they provide significant potentials in applications where both security and client convenience are needed. However, achieving the desirable level of security and usability is not trivial. The key challenges, from a security point of view, are the difficulty to protect the biometric templates, ensuring revocability and allowing easy matching.

In recent years, there has been intensive study on how to secure the biometric templates and a comprehensive coverage of many proposed solutions can be found in [JNN08]. These techniques can be roughly categorized into two types: (1) approached base on non – invertible transformations – where similarity of biometric samples is preserved through the transformation, yet it is difficult to find the original template from a transformed one (e.g., [ASNM05, RCCB07, FBJR07]) and (2) methods based on helper-data, where a recently proposed cryptographic primitive, the *secure sketch*, (or a variant of it) is employed, such that given a noisy biometric sample, the original biometric data can be recovered with the help of some additional information (i.e., a sketch), which makes it possible to use biometric data in the same way passwords are used. These techniques include [JW99, JS06, SLM07b, DRS04, LSM06].

The secure sketch framework does not only allow a more rigorous security analysis (in an information theoretic sense) compared to many other approaches, but also helps generalizing much of the prior works based on helper-data. Most importantly, a sketch allows the exact recovery of the biometric template. Therefore, a *strong extractor* (such as pair-wise independent hash functions) can be further applied on the template to obtain a key that is robust, in the sense that it can be consistently reproduced given a noisy measurement that is similar to the template. However, although it has been

shown that there are a few difficulties in extending these techniques to biometric templates in practice, the most important problem is the fact that the information leakage on the biometric sample is unavoidable when using these schemes [IW07] [IW10]. In the next Sections we address two separate problems with the classical fuzzy sketch approach: the first one gives to the Fuzzy Commitment Scheme a privacy preserving layer of security and the second one, that is a variant of the first, achieves also the goal of avoiding the leakage of information.

5.2 The Fuzzy Commitment Scheme

The Fuzzy Commitment Scheme has been introduced to solve the problem of identity theft and it can be used with several biometrics: ear [TVI⁺04], face [VDVKS⁺06] or signature [MC09]. Roughly speaking a biometric system works making a comparison between a new biometric sample and a set of samples stored in a database, during this process all the sensible information are available to Bob (the server). It is possible that a third party Eve (the attacker) tries to steal information from the database acting like an enrolled user, but providing fake data just to extract some biometric sample. In literature this kind of problem is called *template theft* and it is a central point in the handling of biometric data because the template (biometric) is something that is intrinsic to the user, so a template or biometric sample theft is equivalent to an identity theft.

The Fuzzy Commitment Scheme as proposed in [JW99] is a technique that combines well-known approaches in the areas of Error Correcting Codes (ECC) and cryptography to reach the goal of an efficient commitment scheme. Formally speaking, an ECC is a set of codewords $\mathcal{C} \subseteq \{0, 1\}^n$ for some integer n selected for mapping the information. Therefore, for a message space of size 2^k we need at least $n = k$, but to achieve redundancy, in general, we require that $n > k$. Given the message space $\mathcal{M} = \{0, 1\}^k$, we define $g : \mathcal{M} \rightarrow \mathcal{C}$ as

the *translation function* (sometimes called coding function), thus g is a map from \mathcal{M} to \mathcal{C} . Conversely g^{-1} is the inverse map from \mathcal{C} to \mathcal{M} . The function f is the *decoding function* $f : \{0, 1\}^n \rightarrow \mathcal{C}$ that maps arbitrary n -bit strings to the nearest codeword in \mathcal{C} . We say that f has a correction capability of t if it can correct up to t bit errors.

In the Fuzzy Commitment Scheme, biometric data is treated as a corrupted codeword. Therefore, we use only the decoding function to reconstruct the right associated codeword and we do not need to care about g and g^{-1} . A Fuzzy Commitment Scheme F works on codewords c and binary vectors x where both are strings of length n . In particular for any given x and codeword c , we can express x uniquely by means of the codeword c and an offset δ ($x = c \oplus \delta$) where \oplus is the binary XOR. It is simple to show that the information of x contained in δ depends on the cardinality of \mathcal{C} ¹ in fact, for a given δ there is a number equal to the cardinality of \mathcal{C} of possible x .

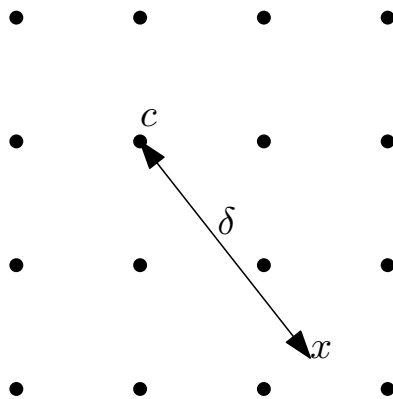


Figure 5.1: Enrollment – The Server chooses c and stores the pair $(\delta, \text{Hash}(c))$.

The original Fuzzy Commitment Scheme in [JW99] works as follows. During the enrollment phase (see Figure 5.1 Enrollment), the client presents a biometric data x and the server chooses a codeword c . At this point the server

¹Note that binarization techniques are often used to manipulate biometric data.

stores, for that client, the pair $(\delta, Hash(c))$ where: $\delta = x \oplus c$ and $Hash(c)$ is the hash of the codeword c .

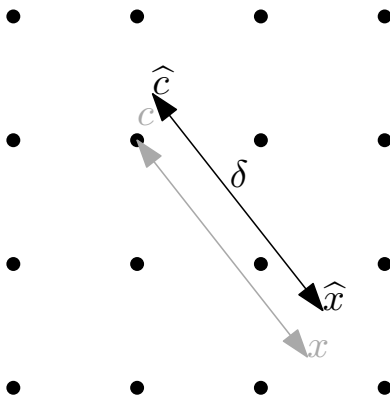


Figure 5.2: Match – The Server checks if $Hash(c) = Hash(f(\hat{c}))$.

During the matching phase (see Figure 5.2 Matching) a new noisy biometric data \hat{x} is presented by a client who claimed his identity, the server computes $\hat{c} = \hat{x} \oplus \delta$ and also $Hash(f(\hat{c}))$ where f is the decoding function. If $Hash(c) = Hash(f(\hat{c}))$ then the client is authenticated. In case of identification, the basic scheme outlined above is repeated for all registered clients, resulting in a 1 to M matching request (M is the total number of enrolled clients).

The Fuzzy Commitment Scheme is not privacy preserving because it does not preserve the identity of the user, in the sense that it is possible to associate the sketch to the client, moreover it produces a leakage of information. As shown in [IW07] and [IW10] the simple sketch approach described above suffers from a leakage of information that cannot be avoided with standard algorithms, in fact as stated in Theorem 2 (pag. 122 of [IW07]): *in the secure Fuzzy Commitment Scheme, information leakage on x is unavoidable*. Therefore, in case of non-trusted parties, the protocol should be secured, in the sense that, after running the protocol, neither the server nor the client should obtain any information beside the output of the protocol. The en-

encrypted sketch (**eSketch**) scheme described in the following sections prevents the information leakage and provides an efficient solution to the user privacy along with template security.

5.3 A Possible Scenario

As an example we may consider the following scenario: an online service (think of a remote medical service) that is accessible by using the fingerprint reader of a standard notebook. As a registered client, Alice wants to access the service, but does not want to reveal her identity, because, for example, she is requesting some particular medical diagnosis and she does not want that anybody knows that she needs a specific diagnosis. The server should be able to verify whether Alice's fingerprint corresponds to a registered client, without knowing which particular client is asking to access the service. This request can be summarized in a *motto* sounding like: *everybody is allowed to know that you are registered to a particular service, but no one is able to know when you use it and for which purpose, moreover none is able to distinguish you among the other clients*. Note that in the above scenario privacy protection is not needed during the enrollment phase. In fact we may assume that when Alice is enrolled she gives a plain version of her biometric. In this phase we can assume that the server is trusted since, for instance, the client is physically present during the enrollment phase.

We propose a scheme based on fuzzy commitment [JW99] which makes possible to perform all the operations in the encrypted domain. In addition to ensuring the security of the biometric data that is always managed in encrypted format, and the revocability of the biometric template ensured by the fuzzy sketch approach, the proposed scheme is capable of protecting the privacy of the client that is going to be authenticated. The proposed scheme addresses the above scenario wherein a client entitled to access a given service is asked to provide her biometric data for accessing the service. The proposed

protocol permits to verify whether a client is included in a list of registered clients without that the server is able to track which client accessed the system and when.

The application of privacy preserving techniques to the biometric verification problem² has been proposed in [BC08] where the biometric data stay encrypted during all the computations thanks to the integration of secure sketches into homomorphic cryptosystems. Moreover, confidentiality of requests made to the database is also obtained thanks to a Private Information Retrieval (PIR) protocol. In particular [BC08] uses the Fuzzy Commitment Scheme described in [JW99], and solves the correcting code problem by using a linear correcting code implementable using Goldwasser-Micali cryptosystem [GM84]. The solution described in [BC08] is more performing than ours in large databases, but for small sets the Lipmaa protocol for PIR (Private Information Retrieval [Lip05]) could be inadequate. Moreover our construction allows parallel computation due to the possibility of exploiting the composite representation introduced in [BPB10] to pack together several encrypted values.

As another proposal, Upmanyu et al. in [UNSJ09] has developed an efficient protocol for biometric verification based on asymmetric cryptosystem (RSA). More specifically, in order to achieve a secure and efficient verification, a linear classifier is used. However, it is highly probable that the same solution using the Paillier cryptosystem would be much more efficient. Moreover, RSA is not semantically secure and due to the structure of the scheme, the client identity is disclosed.

With respect to the above solutions our construction is not a composition of general purpose techniques like PIR or PIS, but it is an ad hoc solution. Moreover delegating the decoding function to the client it is possible to use more complex decoding functions. Finally we used a semantic secure (IND-CPA) cryptosystem like Paillier that provides a more suitable security for the

²Some authors refer this as *claimed identity* authentication, that is a 1:1 matching.

privacy preserving applications.

5.3.1 Security Analysis

In this Section we show the main differences between the standard scenario and the privacy preserving one. As already done for the FingerCode (Section 4.3.1), for the sake of simplicity we recall the scenario described in the previous Section as a sequence of the most relevant actions performed by Alice and Bob.

In the standard scenario Bob, that is the service provider, is also the trusted authority. During the enrollment phase Alice provides a plain version of her biometric trait and Bob stores, in his database, the sketch for that given biometric sample. During this phase Alice must trust Bob. Bob is in charge to protect the database containing the enrolled sketches. In the matching phase Alice provide a new biometric sample to Bob, due to this, Bob is able to associate a given sketch with the correspondent identity. Moreover, during the entire process Bob has to assure the protection of the data by third parties attacks. Bob manages the database and the sketches. The database contains the sketches in plain. The fresh biometric sample (used in the matching phase) is not encrypted.

In the privacy preserving scenario the enrollment phase is identical to the standard case, even if the database contains encrypted sketches and Alice enrolls a biometric samples that is encrypted using her PuK . Due to this Bob is not required to protect the database which contains encrypted sketches. During the matching phase Alice generates an encrypted version of her biometric trait and together with the help of Bob applies a privacy preserving protocol to decide if she can be allowed to the system. In this case Bob is not able to associate any entries in his database to any other identity, so the service provider is not able to identify which user is using the service. Bob manages the database containing the encrypted sketches of all users. Each user manages his cryptosystem keys, his biometric trait and the encrypted

sketch.

5.4 Basic Building Blocks

We recall briefly the notation we will use through the next Sections:

- $x \in \{0, 1\}^n$ is the biometric data consisting of a binary string of length n . We indicate with x_i the i -th bit of the string;
- with \bar{a} we refer to the bit-wise representation of a ;
- c is a codeword in the set \mathcal{C} ;
- with $\llbracket a \rrbracket$ we indicate the Paillier [Pai99] encryption of a ; with $\llbracket \bar{a} \rrbracket$ the bitwise encryption of a . Sometimes we indicate with $\llbracket a \rrbracket_i$ the encryption of a with the key of the client i ;
- PuK and PrK are respectively the public key and the private key of the cryptosystem adopted in the protocol;
- s is the cryptosystem security parameter and ℓ is the bit size of a cryptogram ($\ell = 2s$ for Paillier cryptosystem).

5.4.1 The sub-protocol XOR

By assuming that x and y are binary values (bit), computing the XOR function is equivalent to the following:

$$x \oplus y = x + y - 2xy \tag{5.1}$$

which can be used to compute the XOR function in the encrypted domain.

In the following we consider two main cases:

1. x is encrypted and y is not;
2. both x and y are encrypted.

Computing $\llbracket x \oplus y \rrbracket$ from $\llbracket x \rrbracket$ and y Due to the additive homomorphic properties of Pailler’s cryptosystem, it is possible to rewrite Equation (5.1) as follows:

$$\llbracket x \oplus y \rrbracket = \llbracket x + y - 2xy \rrbracket = \llbracket x \rrbracket \llbracket y \rrbracket \llbracket x \rrbracket^{-2y}. \quad (5.2)$$

The computational complexity is mainly the cost to compute the modular inversion (i.e. $x^{-1} \bmod n$) that requires the same complexity of an exponentiation, so it is $2 \text{ mult} + 1 \text{ exp} \simeq 1 \text{ exp}$.

In this case the bandwidth requirement is 0 since there is no interaction between the parties. Thus, the round complexity is also 0.

Table 5.1: Computational Complexities – XOR with $\llbracket x \rrbracket$ and y .

#exp	Bandwidth	Rounds
1	0	0

Computing $\llbracket x \oplus y \rrbracket$ from $\llbracket x \rrbracket$ and $\llbracket y \rrbracket$ We now suppose that both bit values are available in encrypted format, i.e. Bob knows $\llbracket x \rrbracket$ and $\llbracket y \rrbracket$, where encryption is carried out by using Alice’s *PuK*. The server does not want to reveal neither x nor y to the client, so it chooses two additional random bits r_x and r_y and uses equation (5.2) to compute $\llbracket x \oplus r_x \rrbracket$ and $\llbracket y \oplus r_y \rrbracket$ then sends these values to the client. Note that x and y are perfectly obfuscated by the xor-ing with r_x and r_y , so the client can decrypt them, compute the encryption of $\llbracket (x \oplus r_x) \oplus (y \oplus r_y) \rrbracket$ and send the result back to the server. At this point the server using again equation (5.2) can remove r_x and r_y from the result and obtain $\llbracket x \oplus y \rrbracket$. The entire protocol is shown in Figure 5.3.

Since the server needs to compute the XOR function by using equation (5.2) four times, the client computes two decryptions and one encryption, the complexity is $4 \text{ exp} + 2 \text{ dec} + 1 \text{ enc} \simeq 7 \text{ exp}$.

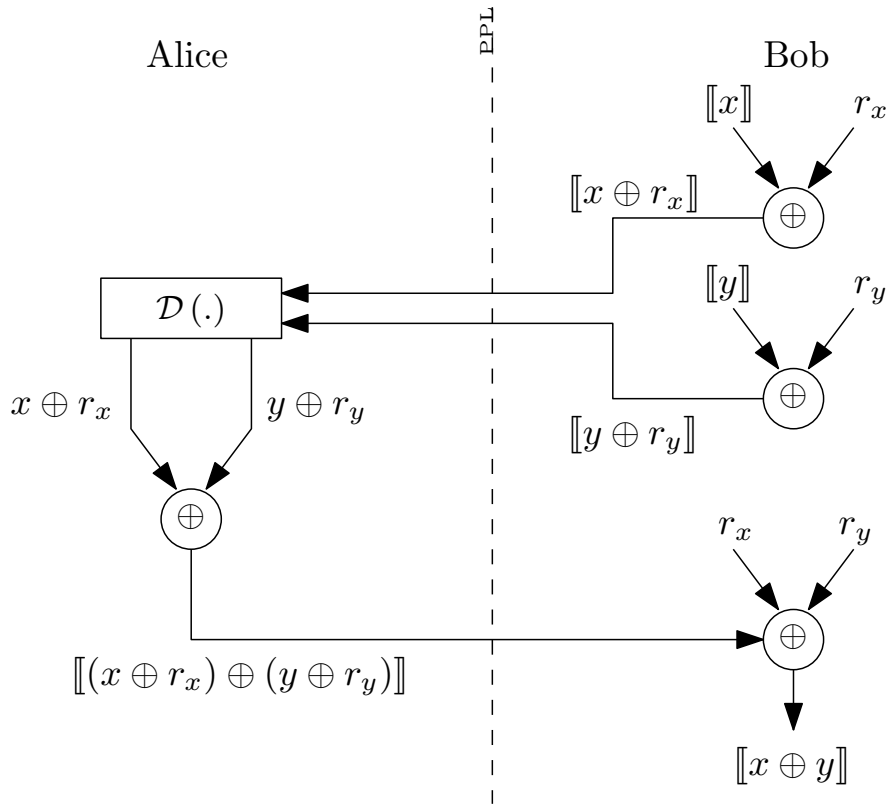


Figure 5.3: Sub protocol XOR with $\llbracket x \rrbracket$ and $\llbracket y \rrbracket$.

This sub-protocol requires a bandwidth of 3ℓ because the server sends two cryptograms to the client that responds with one cryptogram. The round complexity is 2.

Table 5.2: Computational Complexities – XOR with $\llbracket x \rrbracket$ and $\llbracket y \rrbracket$.

#exp	Bandwidth	Rounds
7	3ℓ	2

5.4.2 The sub protocol **eSearch**

A key step in the Fuzzy Commitment Scheme is the search for the codeword in \mathcal{C} that is closest to \hat{c} , i.e. the computation of $f(\hat{c})$. In this subsection we present a protocol to compute such a function when \hat{c} is available in encrypted form to the client. We will refer to such a protocol as **eSearch** functionality. The approach that we will follow is to delegate the computation of f to the client in a such way that the client is not able to understand which are the input and the output of the computation. The details of the ECC code are supposed to be public.

To describe the **eSearch** protocol we start by assuming that the space \mathcal{C} of all the codewords is a linear subspace that is closed under bitwise XOR operation³. If the above holds also the following property holds.

Property 5.1. *We have $f(\hat{c} \oplus d) \oplus d = f(\hat{c})$, $\forall d = c_j \in \mathcal{C}$.*

Proof. Let $c_i = f(\hat{c})$. We surely have $\hat{c} = c_i \oplus \varepsilon$ for some ε . We have:

$$\begin{aligned}
 f(c_i \oplus \varepsilon \oplus d) \oplus d &= \\
 &= f(c_i \oplus \varepsilon \oplus c_j) \oplus c_j = \\
 &= c_i \oplus c_j \oplus c_j = \\
 &= c_i
 \end{aligned} \tag{5.3}$$

where we have exploited the fact that the decoding function is able to correct the error ε whatever codeword c_i is added to, and where due to the linearity assumption the addition of two codewords always results in a valid codeword. \square

Thank to the above result, a very simple **eSearch** protocol can be obtained: Bob blinds \hat{c} by adding to it a random codeword d , then it asks Alice to decode with f the blinded message. The client evaluates f in the plain

³This is always the case with the most common ECC.

domain, re-encrypts the result and sends it back to Bob, that can obtain the encrypted version of the decoded codeword by XOR-ing back the result with d . A more detailed description of the `eSearch` protocol outlined so far is given in Figure 5.4. Note that all codewords are encrypted sample-wise so to allow the application of the first of the two secure XOR protocols described in the previous section.

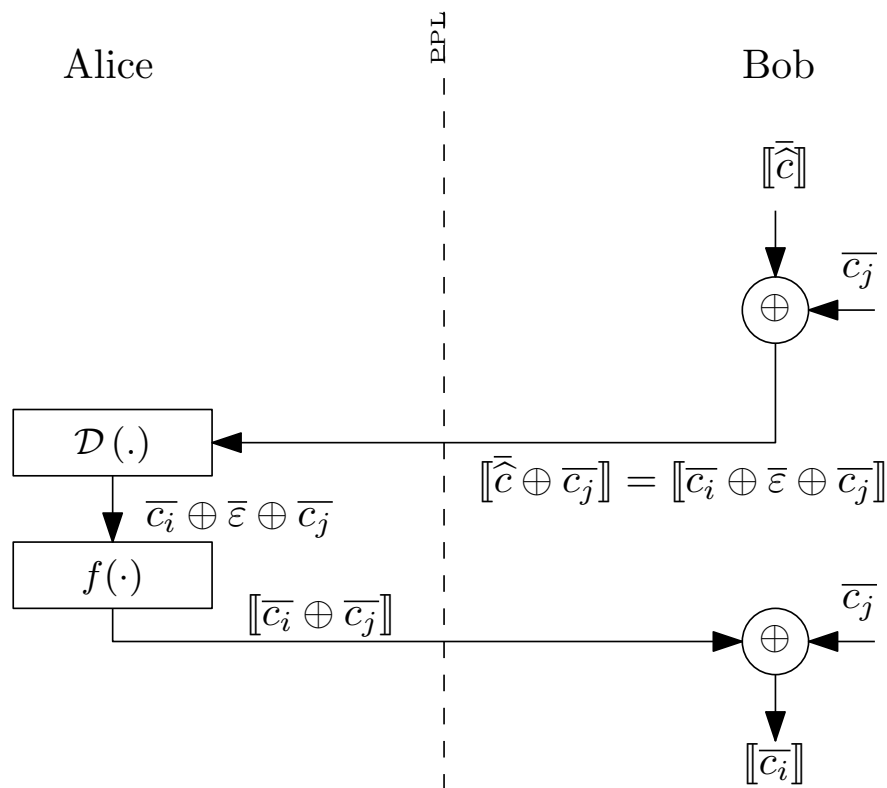


Figure 5.4: `eSearch`.

Security discussion. In the following we argue that, under the assumption that the client is allowed to know ε , `eSearch` is secure in the *honest but curious* model (Section 2.2.2). This is a reasonable assumption since ε reveals only the error between the enrolled biometric data and the new one, and the

error between two biometric measurements can be assumed to be uncorrelated to the biometric value itself. Furthermore this information is revealed only to the client that owns the biometric data⁴. So, while ε can be seen as a leakage of information, this leakage is seen by the client only. This can not be considered to be a problem since the sensible information that needs to be protected is the codeword that the server does not want to reveal. More specifically, the `eSearch` protocol achieves both client and server privacy, in fact during the whole protocol the server sees only encrypted data, from which it can not get any information due to IND-CPA security of the underlying cryptosystem. Considering the server privacy note that an eavesdropper can not get any information from the encrypted values due to the IND-CPA security of Pailler's cryptosystem. As to the client, he is only able to know ε and the blinded codeword message $\widehat{c} + c_j$. We already discussed why disclosing ε is not a problem. As to the blinded message it corresponds to $c_i + \varepsilon + c_j$. Since the client knows ε this is equivalent to knowing $c_i + c_j$. If the server chooses c_j randomly and uniformly over all possible codewords in \mathcal{C} , then it is easy to show that the mutual information $I(c_i; c_i \oplus c_j)$ is equal to zero, hence proving the server privacy of the protocol.

To achieve the security it is necessary to require also that the clients refresh their cryptographic keys periodically. This requirement came from the fact that looking at the modulus used by each client, the server could be able to associate to the modulus the correspondent user. For the sake of simplicity consider the case with only two users with their modulus n_1 and n_2 ($n_1 < n_2$), it clear that in the interval $(n_1, n_2]$ only the second client is able to produce ciphertexts. Due to this the server could be able to understand with probability 1 the second client if he can see a ciphertext in the interval $(n_1, n_2]$. The above situation can be remains true also in the case of M clients, for this reason it is necessary refresh the cryptographic key periodically.

⁴This is really important point because when ε is revealed is just to the owner of the biometric itself, so practically nothing is unveiled.

Complexity. The most expensive operation in the protocol is computing XOR: initially to obfuscate $\llbracket \widehat{c} \rrbracket$ and later to remove the obfuscation. The computational complexity is then dominated by: $2n$ **exp** needed to compute the obfuscations, n decryptions and n encryptions (we recall that the code-words are encrypted bitwise), so: $2n$ **exp** + n **enc** + n **dec** $\simeq 4n$ **exp**. The bandwidth is exactly $2n\ell$ because 2 blocks of n cryptograms are transmitted. Finally, 2 rounds are needed to run **eSearch**.

Table 5.3: *Computational Complexities – eSearch.*

#exp	Bandwidth	Rounds
$4n$	$2n\ell$	2

5.5 The Protocol

We are now ready to describe the overall **eSketch** protocol for privacy preserving authentication. In the rest of this section we suppose that there are M registered clients, moreover we consider that all the values involved in the protocol are bitwise encrypted so for the sake of simplicity we omit the notation $\llbracket \bar{x} \rrbracket$ and we will use just $\llbracket x \rrbracket$.

Enrollment. Let us start by considering the enrollment phase for a generic client j . The j -th Client sends the plain version of his biometric data x_j to be enrolled in the system, moreover he sends also an encrypted obfuscated version $\llbracket x_j \oplus R_j \rrbracket_j$, where R_j is a random blinding factor chosen by the Client. The Server chooses a codeword c , computes $\delta_j = x_j \oplus c$ and stores the pair δ_j and $\llbracket x_j \oplus R_j \rrbracket_j$. Figure 5.5 shows these steps. As we already said, in this phase we assume that the client trusts the server.

Upon presentation of a new noisy biometric data \widehat{x}_j from the client, Bob must check whether this biometric data corresponds to one of the M enrolled

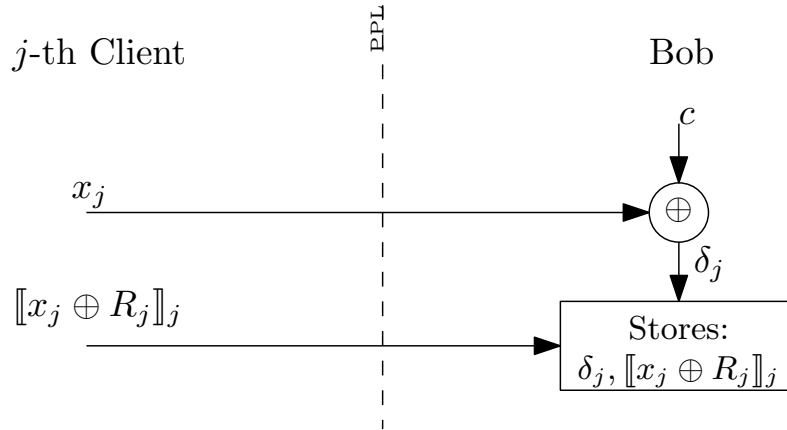


Figure 5.5: *eSketch* – Enrollment.

users. In this phase, the client wants that nothing is disclosed about the noisy biometric \hat{x}_j . At the same time Bob wants that nothing is revealed about the biometric data of the other users and must avoid that a non-registered user results in a positive match. At the end of the protocol, Bob will know only if the user trying to access the system is a registered user, but will not know which user is accessing the system. The above goals are obtained by means of the following protocol.

Matching. In our description we refer to Figure 5.6. Let us assume that the j -th client wants to access the system. He sends $\llbracket \hat{x}_j \rrbracket_j$ and his PuK_j to Bob. Note that in our framework revealing PuK_j does not reveal the identity of the client. The reason for this is that in our set up PuK_j and PrK_j are generated directly by the client during the enrollment with no intervention of a certification authority, so there is nobody that would be able to associate a given PuK_j to the particular j -th user and the PuK is never given to Bob in this phase. Actually the server could be able to trace the behavior of the clients by keeping trace of the usage of the M PuK 's of the clients. This could be a problem for small values of M since it could be possible to trace back to the identity of the client from his behavior. However, for large values

of M as those typically encountered in on-line services, this is unlikely to be a problem. On the contrary, the possibility of tracking users's behavior collectively without that a particular behavior is associated to a given user could be seen as an advantage of the `eSketch` protocol. In any case, to prevent this kind of attack, all the clients may be asked to re-enroll with a new *PuK* regularly, depending on the application.

Since the user did not claim his identity, Bob cannot index the database for a given client. For this reason, for each entry in the database, the server computes $\llbracket \hat{c}_i \rrbracket_j = \llbracket \hat{x}_j \oplus \delta_i \rrbracket_j$ (he can do that by exploiting the homomorphic property of the cryptosystem as in Equation (5.2) obtaining M noisy codewords each one encrypted with the j -th client's *PuK*). At this point the server and the client run the `eSearch` protocol M times to obtain M denoised codewords ($\llbracket c'_i \rrbracket_j$), then Bob XOR's each of them with δ_i . In this way he obtains a set of M enrolled encrypted *candidate*-biometrics: $\llbracket x'_i \rrbracket_j$. For each entry in the database, the server has also stored $\llbracket x_i \oplus R_i \rrbracket_i$ so he can compute $\llbracket W_i \rrbracket_j = \llbracket x_i \oplus R_i \oplus x'_i \oplus R \rrbracket_j$, where R is an additional random number chosen by Bob and used to avoid the possibility that the client is cheating. Note that only if $i = j$ the homomorphic property makes sense (this is due to the standard properties of IND-CPA cryptosystems⁵), in all the other cases the result of this operation is simply a random string of bits. In addition only if there is one $x_i = x'_i$ the j -th Client can be authenticated. To do so the server sends all the $\llbracket W_i \rrbracket_j$ values ($i = 1, M$) to the client. Alice decrypts them and subtracts to each the value R_j she used in the enrollment phase. Then Alice scrambles over i (to obfuscate the matching position to Bob) and sends the results back to the Server. Bob removes the blinding factor R homomorphically and checks if in the list he obtained there is a 0's vector. If this is the case, access is granted.

⁵Note in fact that decrypting a ciphertext with the wrong key does not produce a plaintext correlated with the original one.

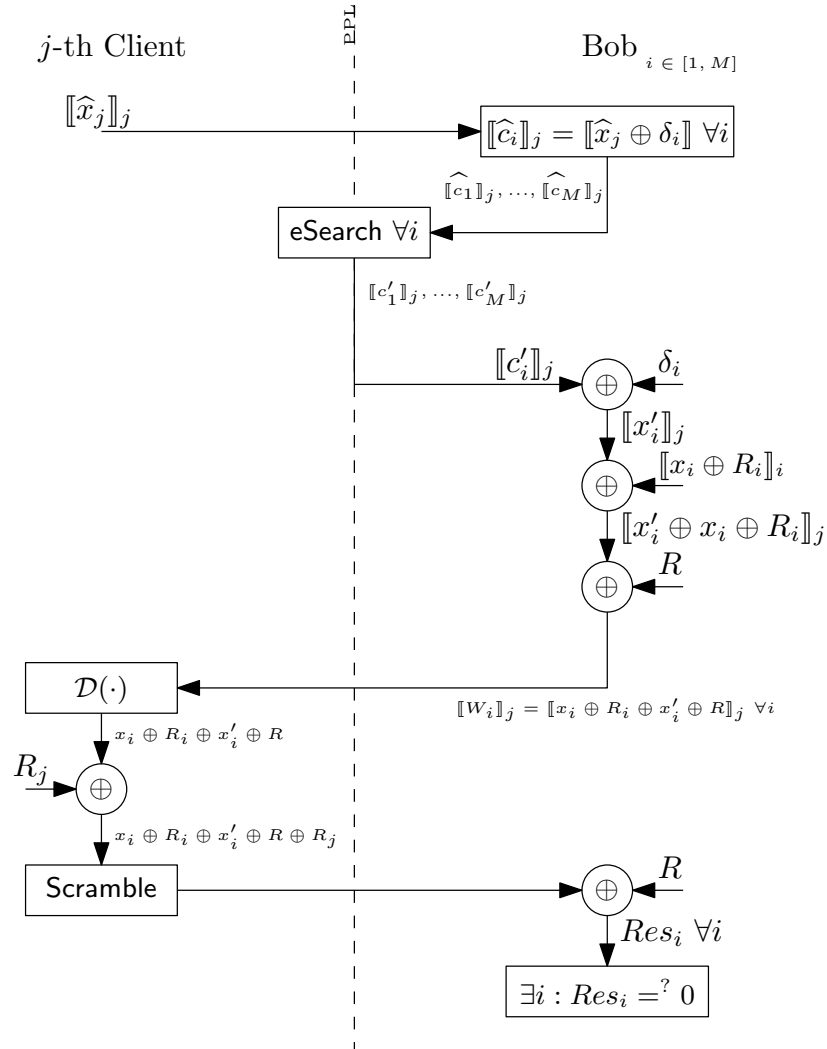


Figure 5.6: *eSketch - Matching*.

5.6 Security

To discuss the security of the *eSketch* protocol we observe that with respect to the XOR and *eSearch* protocols (that we already proved to be secure) the only additional steps in which we may have some leakage of

information are in the last two rounds. To see that no leakage of information occurs during this phase let us consider the two cases: $i = j$ and $i \neq j$. In the first case Alice sees: $x_j \oplus R_j \oplus x'_j \oplus R \oplus R_j$ if the biometric is not too noisy⁶ we have that $x_j = x'_j$ and so the above computation returns only R that is a random value chosen by Bob, and so no leakage of information occurs here. When $i \neq j$ Alice receives: $\llbracket W_i \rrbracket_j = \llbracket x_i \oplus R_i \oplus x'_i \oplus R \rrbracket_j$, when she applies the decryption function \mathcal{D}_j using her PrK , she obtains something that is completely random, since part of the cryptograms is encrypted with a different PuK and so the decryption is completely meaningless.

After that, Alice subtracts R_j and sends back to server $x_i \oplus R_i \oplus x'_i \oplus R \oplus R_j$. Bob removes R and obtains $x_i \oplus R_i \oplus x'_i \oplus R_j$ that is a completely random number. The server, then, sees a string composed by random numbers and, possibly, a zero in a random position, hence no leakage of information occurs on his side well. Finally we observe that if someone tries to access the system without knowing the correct keys, he only sees random string values due to the security of the underlying cryptosystem. Moreover a result in [Gol04] states that the composition of sub-protocols secure in the honest but curious model inherits this security property.

Due to the fact that every client uses his own personal PuK we face with the problem that the server could infer some information by looking at the modulus used by each client. In fact, if we consider the ordered list of the modulus it is clear that there are intervals in which only one client is able to produce ciphers⁷ and the server could be able to understand which is the modulus of a user just comparing the ciphertext with modulus. A possibility to avoid this issue is to ask the users to produce ciphertexts in a smaller interval, this is possible due to self-randomization. Alternatively we could

⁶This is an assumption that must hold if we want that the whole fuzzy sketch approach works.

⁷Consider for instance, the two biggest modulus $M_b < M_a$, only the owner of M_a could produce a ciphertext in the interval $(M_b, M_a]$.

require to refresh the keys periodically.

5.7 Complexities

We now briefly discuss the complexity of the `eSketch` protocol. In doing so, as always, we focus on the most expensive operations. During the *enrollment* phase the computational complexity is:

$$n \text{ enc} + 1 \text{ add} \simeq n \text{ exp}$$

with just 1 round.

The *matching* phase is much more complex and requires:

$$n \text{ enc} + 3n \underbrace{(1 \text{ exp})}_{xor} + \underbrace{M(4n \text{ exp})}_{M \text{ eSearch}} + (Mn) \text{ dec} + \underbrace{(7n \text{ exp})}_{XOR} \quad (5.4)$$

that is dominated by $(5Mn + 11n) \text{ exp}$. Moreover in the matching phase 3 rounds are needed plus those needed to compute $M \text{ eSearch}$ and 1 XOR, for a total of $4 + 2M$ rounds.

Bandwidth. The enrollment phase requires a transfer of 1 plain (we recall that the plaintext size is, at the very most, $\frac{\ell}{2}$ bits), n encrypted values and the *PuK* so: $\frac{\ell}{2} + n\ell + 2\ell = (n + 2.5)\ell$ bits while the matching phase requires:

$$\underbrace{2\ell}_{PuK} + n\ell + \underbrace{2nM\ell}_{M \text{ eSearch}} + nM\ell + nM0.5\ell + \underbrace{6n\ell}_{XOR} = (3.5M + 4)n\ell + 2\ell \quad (5.5)$$

bits.

Table 5.4 shows a summary of the complexities involved in the protocol.

5.8 Avoiding the Leakage of Information

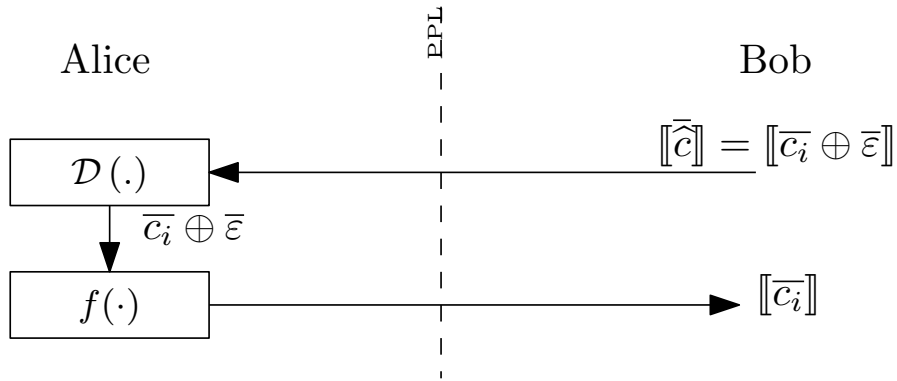
In this section we propose a variant of the above protocol that avoids the leakage of information inherent in letting the server know δ . In the rest

Table 5.4: Protocol *eSketch* Complexities

	#exp	Bandwidth	Round
Enrollment	n	$(n + 2.5)\ell$	1
Matching	$5Mn + 11n$	$(3.5Mn + 4n + 2)\ell$	$4 + 2M$

of this section we consider that the codebook \mathcal{C} is public available and the security model is the honest but curious model.

First of all, note that allowing the codebook to be public, we can simplify the *eSearch* algorithm as report in Figure 5.7 removing the obfuscation values.

**Figure 5.7:** *eSearch Variant*.

Note that we cannot allow Alice to send back the codeword in plain because Bob could be able to identify Alice if M is small or if the protocol is run several times. We point out that this variant of the protocol holds only in the honest but curious model because in the malicious model Alice could inject false values in place of c_i to try to get authenticated.

Enrollment. During the enrollment phase (see Figure 5.8) the j -th client is able to produce a random c due to the fact that we consider \mathcal{C} to be known. The client computes $\delta_j = c \oplus x_j$ using a fresh sample of his biometric then he

encrypts it ($\llbracket \delta_j \rrbracket_j$) and sends this encryption with an obfuscated and encrypted version of his new noisy biometric: $\llbracket x_j \oplus R_j \rrbracket_j$ where R_j is a random blinding.

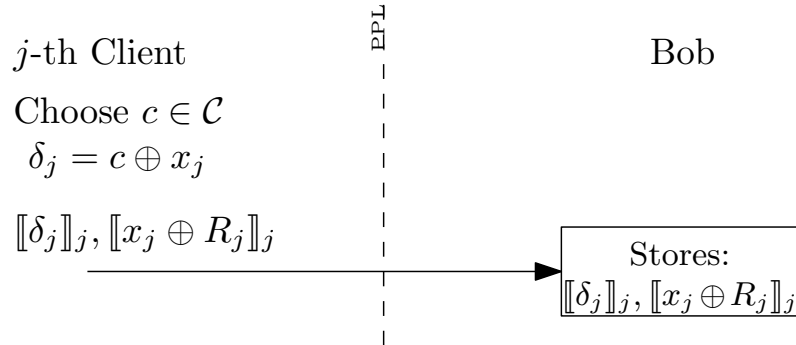


Figure 5.8: *eSketch Variant – Enrollment.*

Matching. The matching phase is shown in Figure 5.9 and is identical to the original one, so we omit the details; we just note that this solution is a little more complex due to the fact that it requires 3 times the computation of the XOR protocol instead of only one in the original version.

The leakage is avoided due to the fact that now the entire algorithm runs on encrypted data. In the previous version δ was stored by the server in plain form, while now all the information related with the biometric data is encrypted.

5.8.1 Complexities

We now study the complexity of the new version of **eSearch**. It is simple to show that only $2n$ ciphertexts are exchanged in 2 rounds between Bob and Alice thus we have a bandwidth complexity of $2nl$. The computational complexity is only $n \exp$ due to the decryption computed by Alice. During the **enrollment** phase Alice computes mostly $2n$ encryption ($2n \exp$) and sends to Bob $2n$ ciphertexts in 1 round. The **matching** phase is the most complex. We start focusing on the number of rounds that are 3 plus M calls

bits. We now consider the computational complexity and we obtain:

$$n \text{ enc} + 3n \underbrace{(7) \text{ exp}}_{\text{XOR}} + n \underbrace{(1) \text{ exp}}_{\text{xor}} + M \underbrace{(4n) \text{ exp}}_{\text{eSearch}} + (Mn) \text{ dec} \quad (5.7)$$

for a total of $5Mn + 23n$ exponentiations. Table 5.5 summarizes the results obtained.

Table 5.5: Variant Protocol *eSketch* Complexities.

	#exp	Bandwidth	Round
eSearch	n	$2n\ell$	2
Enrollment	$2n$	$2n\ell$	1
Matching	$5Mn + 23n$	$(3.5Mn + 10n + 2)\ell$	$9 + 2M$

Finally, comparing the complexities of this construction with the previous one, it is possible to note that they are really close and differ only for constants, so we can affirm that the asymptotic complexities are the same.

5.9 Summary

In this Chapter we introduced a privacy preserving version of the Fuzzy Commitment Scheme. The fuzzy commitment approach has gained popularity as a way to protect biometric data used for identity verification of authentication. Moreover in force of its generality it can be used with different biometrics. The *eSketch* has been proposed in this chapter as a possible solution for the above problems by resorting to tools from multi party computation relying on the additively homomorphic property of the underlying cryptosystem. In particular, the complexity and the security of the proposed protocol has been discussed. Finally in the last section we proposed a variant of our construction that is able to avoid the leakage of information typical of fuzzy commitment schemes.

support of a third party as warranter, these constructions are able to provide security by construction and in force of the security models they use.

In this dissertation we examined privacy preserving solutions in biometric applications as special cases of encrypted query to a plain database (the FingerCode) and encrypted query to an encrypted database (the Fuzzy Commitment Scheme).

The FingerCode is an approach to protect the privacy of the biometric data (in particular fingerprint) in distributed biometric systems. In our construction, the biometric data of the client is captured and afterwards, an encrypted representation of the computed template is produced. Then our protocol allows the computation of the FingerCode algorithm on the encrypted data. We have shown that our protocol is secure in the honest but curious model and we detailed all the complexities involved. Additionally we exploited that by reducing the size of the template it is possible to obtain a smaller representation of the data and consequently a reduced encrypted template thus producing a more performing practical implementation.

The good results obtained suggest that it is possible to stress a little more our construction in the perspective of optimizing the run time of the application. Table 4.6 shows clearly that the privacy preserving FingerCode is not ripe to real world applications however it is a proof that could be useful as a starting point for further improvements.

The privacy preserving version of the Fuzzy Commitment Scheme successfully addresses the problem of keeping secret the user identity and the biometric data itself. This privacy preserving capability makes the proposed scheme suitable for protecting the users privacy by allowing them to be authenticated anonymously. We provided an outline of the security proof in the honest but curious model by assuming that the error correction code satisfies certain, rather common, properties. The computational complexity of the protocol is linear in both the number of entries in the database (M) and the length of template representation (n). Moreover, also the bandwidth required

to compute the protocol depends linearly on the number of entries and the template size. Finally the number of rounds depends just on the number of enrolled clients.

The proposed technique is just a prototype however it clearly shows that a generic (in the sense of biometric-independent) protocols can be developed. A real world implementation could be a future work, it is reasonable, expecting that an ad hoc modeling of the `eSketch` could lead to good performances, but probably, still infeasible in a real world scenario.

Finally it is clear that our results and findings are applicable to a broad variety of different settings not only using different biometric samples but also in different applications. We should not forget that these constructions are just possible solutions of the biggest problem: querying a database in a privacy preserving fashion and so they are suitable in many different scenarios.

6.2 Track for Future Works

This section discusses some of the extensions that may be pursued to improve the work of this thesis and in general this field of research. Future works could be oriented to the application of the results we obtained to the development of privacy preserving systems with high accuracy and with high performance. This goal can be split into different parts.

- **Efficient real world applications.** We have shown a real world implementation of the FingerCode algorithm giving a good estimation of the performance in a case study close to reality. A future work could certainly be the realization of a similar demonstrator for the privacy preserving sketch.
- **Improvements of the security model.** The honest but curious model is suitable in a number of applications, but sometimes stronger security is required. Sometimes Alice (or Bob) could be malicious or

act in a malicious way trying to steal sensible information to the other party. Studies on privacy preserving protocols in the malicious model would solve this limitation.

- **Extension of the basic cryptographic tools.** Biometric systems mostly consist of operations such as computing distances, and thresholding (querying a database). Advanced applications can be investigated to identify more complex and specific building blocks. Thus, taking advantage from the composability of the honest but curious model, construct the new generation of privacy preserving algorithms and protocols.
- **Algebraically homomorphic cryptosystems.** Gentry in [Gen09] proposed a cryptosystem that is algebraically homomorphic, but it is still too complex to be used in practice. An efficient cryptosystem able to preserve additions and multiplications could speed up the multi party computation protocols with significant benefits for applications.
- **Formalization of the signal processing in encrypted domain discipline.** Signal or data processing in the encrypted domain is a very young discipline so there are still no general methodologies to approach problems and develop solutions. For instance there are a lot of works in this field that do not examine in detail the three aspects of complexity of the protocols: number of rounds; number of bits transmitted; number of basic operations. Moreover, industrial designers and developers should be informed about the effectiveness of these techniques in the everyday life.

Privacy is important and today we can improve the technologies to realize the next generation of applications able to protect sensible data and indirectly the owners of such informations.

Did you know that every 3 seconds someone's identity is stolen? More than 10.5 million of identities per year ([Wil07]). In this thesis we have shown that

this new generation of systems based on privacy preserving techniques can be developed and these algorithms and protocols could be the way for a safer digital world.

Bibliography

- [ABF⁺08] E. Aimeur, G. Brassard, J.M. Fernandez, F.S.M. Onana, and Z. Rakowski. Experimental Demonstration of a Hybrid Privacy-Preserving Recommender System. In *The Third International Conference on Availability, Reliability and Security*, pages 161–170. IEEE, 2008.
- [AGR96] J.J. Atick, P.A. Griffin, and A.N. Redlich. Statistical approach to shape from shading: reconstruction of three-dimensional face surfaces from single two-dimensional images. *Neural Computation*, 8(6):1321–1340, 1996.
- [AKS03] A. Adelsbach, S. Katzenbeisser, and A.R. Sadeghi. Cryptography meets watermarking: detecting watermarks with minimal or zero knowledge disclosure. In *XI European Signal Processing Conference*, volume 1, pages 446–449. Citeseer, 2003.
- [AS00] R. Agrawal and R. Srikant. Privacy-preserving data mining. *ACM Sigmod Record*, 29(2):439–450, 2000.
- [Ash00] J. Ashbourn. *Biometrics: advanced identity verification*. Springer-Verlag London, UK, 2000.
- [ASN^M05] R. Ang, R. Safavi-Naini, and L. McAven. Cancelable key-based fingerprint templates. In *Information Security and Privacy*, pages 242–252. Springer, 2005.

- [BBJ⁺09] E. Barker, W. Burr, A. Jones, T. Polk, S. Rose, M. Smid, and Q. Dang. Recommendation for Key Management. *NIST special publication*, 800:57, 2009.
- [BC08] J. Bringer and H. Chabanne. An authentication protocol with encrypted biometric data. In *Proceedings of the Cryptology in Africa 1st international conference on Progress in cryptology*, pages 109–124. Springer-Verlag, 2008.
- [BCP03] E. Bresson, D. Catalano, and D. Pointcheval. A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications. *Advances in Cryptology-ASIACRYPT 2003*, 1:37–54, 2003.
- [BDJ04] R. Brinkman, J. Doumen, and W. Jonker. Using secret sharing for searching in encrypted data. *Secure Data Management*, 1:18–27, 2004.
- [Ben94] J. Benaloh. Dense probabilistic encryption. In *Proceedings of the Workshop on Selected Areas of Cryptography*, pages 120–128. Citeseer, 1994.
- [BFK⁺09] M. Barni, P. Failla, V. Kolensikov, R. Lazzeretti, A. Paus, A. Sadeghi, and T. Schneider. Efficient Privacy-Preserving Classification of ECG Signals. In *Workshop on Information Forensics and Security, WIFS 2009*, 2009.
- [BFK⁺10] M. Barni, P. Failla, V. Kolesnikov, R. Lazzeretti, A.R. Sadeghi, and T. Schneider. Secure evaluation of private linear branching programs with medical applications. *Computer Security–ESORICS 2009*, pages 424–439, 2010.
- [BGN05] D. Boneh, E.J. Goh, and K. Nissim. Evaluating 2-DNF formulas on ciphertexts. *Theory of Cryptography*, pages 325–341, 2005.
- [Bis91] M. Bishop. Password management. In *Proceedings of IEEE COMPCON*, volume 91, pages 167–169, 1991.
- [Bjo67] A. Bjorck. Solving linear least squares problems by Gram-Schmidt orthogonalization. *BIT Numerical Mathematics*, 7(1):1–21, 1967.

- [BL96] D. Boneh and R. Lipton. Algorithms for black-box fields and their application to cryptography. In *Advances in Cryptology CRYPTO96*, pages 283–297. Springer, 1996.
- [BPB08] T. Bianchi, A. Piva, and M. Barni. Implementing the discrete Fourier transform in the encrypted domain. In *IEEE International Conference on Acoustics, Speech and Signal Processing, 2008. ICASSP 2008*, pages 1757–1760, 2008.
- [BPB10] T. Bianchi, A. Piva, and M. Barni. Composite signal representation for fast and storage-efficient processing of encrypted signals. *Information Forensics and Security, IEEE Transactions on*, 5(1):180–187, 2010.
- [BPSW07] J. Brickell, D.E. Porter, V. Shmatikov, and E. Witchel. Privacy-preserving remote diagnostics. In *Proceedings of the 14th ACM conference on Computer and communications security*, page 507. ACM, 2007.
- [Cam04] JL Camp. Digital identity. *IEEE Technology and Society Magazine*, 23(3):34–41, 2004.
- [CCMR06] A. Cillard, L. Coppolino, N. Mazzocca, and L. Romano. Elliptic curve cryptography engineering. *Proceedings of the IEEE*, 94(2):395–406, 2006.
- [CG90] W.Y. Chan and A. Gersho. High fidelity audio transform coding with vector quantization. In *1990 International Conference on Acoustics, Speech, and Signal Processing, 1990. ICASSP-90.*, pages 1109–1112, 1990.
- [Cla94] R. Clarke. Human identification in information systems: Management challenges and public policy issues. *Information Technology & People*, 7(4):6–37, 1994.
- [CM94] A. Collins and NE Morton. Likelihood ratios for DNA identification. *Proceedings of the National Academy of Sciences*, 91(13):6007, 1994.
- [CQ92] C.K. Chui and E. Quak. Wavelets on a bounded interval. *Numerical methods of approximation theory*, 9(1):53–57, 1992.

- [cro] CROSSMATCH Technologies, Verifier 300, <http://www.neurotechnology.com>.
- [CS07] A. Cavoukian and A. Stoianov. Biometric encryption: a positive-sum technology that achieves strong authentication, security and privacy. *Information and Privacy Commissioner/Ontario*, 2007.
- [Dau85] J.G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America A*, 2(7):1160–1169, 1985.
- [Dau93] JG 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.
- [Dau03] J. Daugman. The importance of being random: statistical principles of iris recognition. *Pattern Recognition*, 36(2):279–291, 2003.
- [Dau05] J. Daugman. Results from 200 billion iris cross-comparisons. *University of Cambridge Technical Report UCAM-CL-TR-635*, 2005.
- [Dau06] J. Daugman. Probing the uniqueness and randomness of IrisCodes: Results from 200 billion iris pair comparisons. *Proceedings of the IEEE*, 94(11):1927–1935, 2006.
- [DD95] J.G. Daugman and C.J. Downing. Demodulation, predictive coding, and spatial vision. *Journal of the Optical Society of America A*, 12(4):641–660, 1995.
- [DGK07] I. Damgård, M. Geisler, and M. Krøigaard. Efficient and secure comparison for on-line auctions. In *Information Security and Privacy*, pages 416–430. Springer, 2007.
- [DJ01] I. Damgård and M. Jurik. A Generalization, a Simplification and Some Applications of Paillier’s Probabilistic Public-Key System. In *Public Key Cryptography*, pages 119–136. Springer, 2001.
- [DRS04] Y. Dodis, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *Advances in cryptology-Eurocrypt 2004*, pages 523–540. Springer, 2004.

- [EFG⁺09] Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Legendijk, and T. Toft. Privacy-preserving face recognition. In *Privacy Enhancing Technologies*, pages 235–253. Springer, 2009.
- [ElG85] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Advances in Cryptology*, pages 10–18. Springer, 1985.
- [Erk10] Z. Erkin. Secure signal processing: Privacy preserving cryptographic protocols for multimedia. *Volume Ph.D.*, 2010.
- [EW06] D.P.W. Ellis and R.J. Weiss. Model-based monaural source separation using a vector-quantized phase-vocoder representation. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 5. IEEE, 2006.
- [Fai10] P. Failla. Heuristic Search in Encrypted Graphs. *Fourth International Conference on Emerging Security Information, Systems and Technologies, 2010. SECURWARE'10*, pages 82–87, 2010.
- [FB10] P. Failla and M. Barni. Gram - Schmidt Orthogonalization on Encrypted Vectors. In *21st International Tyrrhenian Workshop on Digital Communications, ITWDC 2010*, 2010.
- [FBJR07] F. Farooq, R.M. Bolle, T.Y. Jea, and N. Ratha. Anonymous and revocable fingerprint recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR'07*, pages 1–7, 2007.
- [FMA94] K.R. Farrell, R.J. Mammone, and K.T. Assaleh. Speaker recognition using neural networks and conventional classifiers. *IEEE Transactions on speech and audio processing*, 2(1 Part 2):194–205, 1994.
- [Fur97] S. Furui. Recent advances in speaker recognition. *Pattern Recognition Letters*, 18(9):859–872, 1997.
- [Gen09] C. Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, pages 169–178. ACM, 2009.

- [GHV10] C. Gentry, S. Halevi, and V. Vaikuntanathan. A simple BGN-type cryptosystem from LWE. *Advances in Cryptology–EUROCRYPT 2010*, pages 506–522, 2010.
- [GM84] S. Goldwasser and S. Micali. Probabilistic encryption* 1. *Journal of computer and system sciences*, 28(2):270–299, 1984.
- [GMW87] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pages 218–229. ACM, 1987.
- [Gol04] O. Goldreich. *Foundations of cryptography*. Cambridge University Press, 2004.
- [Gra] T. Granlund. The GNU multiple precision arithmetic library, 2004.
- [GVL96] G.H. Golub and C.F. Van Loan. *Matrix computations*. Johns Hopkins Univ Pr, 1996.
- [HBP02] AL Higgins, LG Bahler, and JE Porter. Voice identification using nearest-neighbor distance measure. In *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*, volume 2, pages 375–378. IEEE, 2002.
- [Her05] I.N. Herstein. *Noncommutative rings*. The Mathematical Association of America, 2005.
- [HNR68] P.E. Hart, N.J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.
- [HWJ98] L. Hong, Y. Wan, and A. Jain. Fingerprint image enhancement: algorithm and performance evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):777–789, 1998.
- [IBG03] LLC International Biometric Group. BioPrivacy Initiative, 2003.
- [IR97] K. Inman and N. Rudin. *An Introduction to Forensic DNA Analysis*. CRC press, Boca Raton, Florida, 1997.
- [IRR90] K.F. Ireland, M. Rosen, and M.I. Rosen. *A classical introduction to modern number theory*. Springer, 1990.

- [Ita75] F. Itakura. Line spectrum representation of linear predictor coefficients of speech signals. *The Journal of the Acoustical Society of America*, 57:S35, 1975.
- [IW07] T. Ignatenko and F. Willems. On Privacy in Secure Biometric Authentication Systems. In *IEEE International Conference on Acoustics, Speech and Signal Processing, 2007. ICASSP 2007*, volume 2, pages 121 – 124, 2007.
- [IW10] T. Ignatenko and F.M.J. Willems. Information leakage in fuzzy commitment schemes. *Information Forensics and Security, IEEE Transactions on*, 5(2):337–348, 2010.
- [JHB97] A. Jain, L. Hong, and R. Bolle. On-line fingerprint verification. *IEEE transactions on pattern analysis and machine intelligence*, 19(4):302–314, 1997.
- [JHPB97] A.K. Jain, L. Hong, S. Pankanti, and R. Bolle. An identity-authentication system using fingerprints. *Proceedings of the IEEE*, 85(9):1365–1388, 1997.
- [JIP⁺04] M. Johnson, P. Ishwar, V. Prabhakaran, D. Schonberg, and K. Ramchandran. On compressing encrypted data. *IEEE Transactions on Signal Processing*, 52(10):2992–3006, 2004.
- [JJW⁺93] Pugh JA, Jacobson JM, Van Heuven WA, Watters JA, Tuley MR, Lairson DR, Lorimor RJ, Kapadia AS, and Velez R. Screening for diabetic retinopathy. The wide-angle retinal camera. *Diabetes Care*, 16(6):889–95, 1993.
- [JLZ⁺02] D.N. Jiang, L. Lu, H.J. Zhang, J.H. Tao, and L.H. Cai. Music type classification by spectral contrast feature. In *Proc. ICME*, volume 1, pages 113–116, 2002.
- [JNN08] A.K. Jain, K. Nandakumar, and A. Nagar. Biometric template security. *EURASIP Journal on Advances in Signal Processing*, 2008:1–17, 2008.

- [JPH99] A.K. Jain, S. Prabhakar, and L. Hong. A multichannel approach to fingerprint classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):348–359, 1999.
- [JPHP00] A.K. Jain, S. Prabhakar, L. Hong, and S. Pankanti. Filterbank-based fingerprint matching. *IEEE Transactions on Image Processing*, 9(5):846–859, 2000.
- [JRP06a] A. K. Jain, A. Ross, and S. Pankanti. Biometrics: A Tool for Information Security. *IEEE transactions on information forensics and security*, 1(2):125–143, June 2006.
- [JRP06b] A.K. Jain, A. Ross, and S. Pankanti. Biometrics: a tool for information security. *IEEE transactions on information forensics and security*, 1(2):125–143, 2006.
- [JS05] H. Jones and J.H. Soltren. Facebook: Threats to privacy. *Project MAC: MIT Project on Mathematics and Computing*, 2005.
- [JS06] A. Juels and M. Sudan. A fuzzy vault scheme. *Designs, Codes and Cryptography*, 38(2):237–257, 2006.
- [JW99] A. Juels and M. Wattenberg. A fuzzy commitment scheme. In *Proceedings of the 6th ACM conference on Computer and communications security*, pages 28–36. ACM, 1999.
- [KLC⁺08] S. Katzenbeisser, A. Lemma, M.U. Celik, M. van der Veen, and M. Maas. A buyer–seller watermarking protocol based on secure embedding. *IEEE Transactions on Information Forensics and Security*, 3(4):783–786, 2008.
- [Kob94] N. Koblitz. *A course in number theory and cryptography*. Springer, 1994.
- [Koc96] P. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In *Advances in Cryptology CRYPTO96*, pages 104–113. Springer, 1996.
- [LBW08] H.R. Lipford, A. Besmer, and J. Watson. Understanding privacy settings in facebook with an audience view. In *Proceedings of the 1st Con-*

- ference on Usability, Psychology, and Security*, pages 1–8. USENIX Association, 2008.
- [Lee99] C.J. Lee. Fingerprint feature extraction using Gabor filters. *Electronics Letters*, 35:288, 1999.
- [LHW98] Z. Liu, J. Huang, and Y. Wang. Classification of TV programs based on audio information using hidden Markov model. In *IEEE Workshop on Multimedia Signal Processing*, pages 27–32. Citeseer, 1998.
- [Lip05] H. Lipmaa. An oblivious transfer protocol with log-squared communication. *Information Security*, pages 314–328, 2005.
- [LLM06] S. Laur, H. Lipmaa, and T. Mielikäinen. Cryptographically private support vector machines. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 618–624. ACM New York, NY, USA, 2006.
- [LP08] Y. Lindell and B. Pinkas. Privacy preserving data mining. *Journal of cryptology*, 15(3):177–206, 2008.
- [LP09] Y. Lindell and B. Pinkas. A proof of security of yaos protocol for two-party computation. *Journal of cryptology*, 22(2):161–188, 2009.
- [LSM06] Q. Li, Y. Sutcu, and N. Memon. Secure sketch for biometric templates. *Advances in Cryptology-ASIACRYPT 2006*, pages 99–113, 2006.
- [Lyn] B. Lynn. The pairing-based cryptography (PBC) library.
- [MC09] E. Maiorana and P. Campisi. Fuzzy Commitment for Function Based Signature Template Protection. *Signal Processing Letters, IEEE*, 17(3):249–252, 2009.
- [MGH96] C.A. Melchor, P. Gaborit, and J. Herranz. Additively Homomorphic Encryption with t-Operand Multiplications. *Crypto 2010*, 1996.
- [MQQ⁺06] Y.D. Ma, C.L. Qi, Z.B. Qian, F. Shi, and Z.F. Zhang. A novel image compression coding algorithm based on pulse-coupled neural network and Gram-Schmidt orthogonal base. *Dianzi Xuebao(Acta Electronica Sinica)*, 34(7):1255–1259, 2006.

- [NS98] D. Naccache and J. Stern. A new public key cryptosystem based on higher residues. In *Proceedings of the 5th ACM conference on Computer and communications security*, page 66. ACM, 1998.
- [O’G03] L. O’Gorman. Comparing passwords, tokens, and biometrics for user authentication. *Proceedings of the IEEE*, 91(12):2021–2040, 2003.
- [OPB07] C. Orlandi, A. Piva, and M. Barni. Oblivious neural network computing via homomorphic encryption. *European Journal of Information Systems*, 2007(1), 2007.
- [OPJM10] M. Osadchy, B. Pinkas, A. Jarrous, and B. Moskovich. SCiFI-a system for secure face identification. In *2010 IEEE Symposium on Security and Privacy*, pages 239–254. IEEE, 2010.
- [Orf90] S.J. Orfanidis. Gram-Schmidt neural nets. *Neural Computation*, 2(1):116–126, 1990.
- [Pai99] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in CryptologyEUROCRYPT99*, pages 223–238. Springer, 1999.
- [pDa] Neurotechnology, dataset Cross Match Verifier 300, <http://www.neurotechnology.com>.
- [PP05] J. Pejas and A. Piegat. *Enhanced methods in computer security, biometric and artificial intelligence systems*. Springer Verlag, 2005.
- [RAD78] R.L. Rivest, L. Adleman, and M.L. Dertouzos. On data banks and privacy homomorphisms. *Foundations of Secure Computation*, pages 169–178, 1978.
- [Rap04] D.K. Rappe. Homomorphic cryptosystems and their applications. *Volume Ph.D.*, 2004.
- [RCCB07] N.K. Ratha, S. Chikkerur, J.H. Connell, and R.M. Bolle. Generating cancelable fingerprint templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 561–572, 2007.
- [RSA78] R.L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):126, 1978.

- [RSR78] L.R. Rabiner, R.W. Schafer, and L.R. Rabiner. *Digital processing of speech signals*. Prentice-hall Englewood Cliffs, NJ, 1978.
- [Sam01] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, pages 1010–1027, 2001.
- [Ser99] G. Seroussi. Elliptic curve cryptography. In *Information Theory and Networking Workshop, 1999*, 1999.
- [SLM07a] Y. Sutcu, Q. Li, and N. Memon. Protecting biometric templates with sketch: Theory and practice. *IEEE Transactions on Information Forensics and Security*,, pages 503–512, 2007.
- [SLM07b] Y. Sutcu, Q. Li, and N. Memon. Protecting biometric templates with sketch: Theory and practice. *IEEE Transactions on Information Forensics and Security*, 2(3):503 – 512, September 2007.
- [SP07] A. Sharma and K.K. Paliwal. Fast principal component analysis using fixed-point algorithm. *Pattern Recognition Letters*, 28(10):1151–1155, 2007.
- [SPG⁺06] R. Sassi, V. Piuri, M. Gamassi, F. Scotti, and S. Cimito. Privacy issues in biometric identification. pages 40–42, 2006.
- [SS98] P. Samarati and L. Sweeney. Generalizing data to provide anonymity when disclosing information (abstract). In *Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, page 188. ACM, 1998.
- [SSW09] A.R. Sadeghi, T. Schneider, and I. Wehrenberg. Efficient privacy-preserving face recognition. In *International Conference on Information Security and Cryptology (ICISC09)*, ser. LNCS. Springer, 2009.
- [STP09] K. Simoens, P. Tuyls, and B. Preneel. Privacy weaknesses in biometric sketches. In *2009 30th IEEE Symposium on Security and Privacy*, pages 188–203. IEEE, 2009.
- [TP91] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991.

- [TVI⁺04] P. Tuyls, E. Verbitskiy, T. Ignatenko, D. Denteneer, and T. Akkermans. Privacy protected biometric templates: Acoustic ear identification. *Proceedings of SPIE: Biometric Technology for Human Identification*, 5404:176–182, 2004.
- [UNSJ09] M. Upmanyu, A. Namboodiri, K. Srinathan, and C. Jawahar. Efficient biometric verification in encrypted domain. *Advances in Biometrics*, pages 899–908, 2009.
- [vDGHV10] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. *Advances in Cryptology—EUROCRYPT 2010*, pages 24–43, 2010.
- [VDVKS⁺06] M. Van Der Veen, T. Kevenaar, G.J. Schrijen, T.H. Akkermans, F. Zuo, et al. Face biometrics with renewable templates. In *Proceedings of SPIE*, volume 6072, pages 205–216. Citeseer, 2006.
- [WAH97] J. Weng, N. Ahuja, and T.S. Huang. Learning recognition and segmentation using the cresceptron. *International Journal of Computer Vision*, 25(2):109–143, 1997.
- [Wei] Eric W. Weisstein. Poincar-Hopf Index Theorem. *From MathWorld—A Wolfram Web Resource*. <http://mathworld.wolfram.com/Poincare-HopfIndexTheorem.html>.
- [WG97] K. Wang and T. Gasser. Alignment of curves by dynamic time warping. *The Annals of Statistics*, 25(3):1251–1276, 1997.
- [Wil07] M.R. Wilson. *Frequently Asked Questions About Identity Theft*. The Rosen Publishing Group, 2007.
- [WJMM04] J.L. Wayman, A.K. Jain, D. Maltoni, and D. Maio. Biometric Systems: Technology, Design and Performance Evaluation. 2004.
- [WWP00] CL Wilson, CI Watson, and EG Paek. Effect of resolution and image quality on combined optical and neural network fingerprint matching. *Pattern Recognition*, 33(2):317–331, 2000.
- [Yao82] A.C. Yao. Protocols for secure computations. In *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science*, volume 23, pages 160–164. Citeseer, 1982.

-
- [Yao86] A.C.C. Yao. How to generate and exchange secrets. In — *27th Annual Symposium on Foundations of Computer Science*, pages 162–167. IEEE, 1986.
- [ZZY+08] G.Q. Zhang, G.Q. Zhang, Q.F. Yang, S.Q. Cheng, and T. Zhou. Evolution of the Internet and its cores. *New Journal of Physics*, 10:123027, 2008.
- [ZZZ04] W. Zheng, C. Zou, and L. Zhao. Real-time face recognition using Gram-Schmidt orthogonalization for LDA. *Pattern Recognition*, 2:403–406, 2004.

Index

- asymmetric, 9, 72, 99
- Big Brother, 51
- Bresson, 8
- CODIS, 51
- complex wavelets, 47
- computational complexities, 23, 84
- computational complexity, 9, 23, 24, 27, 32, 70, 77, 86, 100, 105, 111, 113, 114, 119
- cryptosystem, 8, 13, 18, 19, 21, 22, 24, 26, 29, 30, 71, 75, 83, 99, 100, 105, 108, 110, 115, 120
- cryptosystems, 7, 8, 18, 19, 21, 29, 71, 76, 108, 120
- Damgaard, 8, 29
- Damgaard – Jurik cryptosystem, 29
- DGK cryptosystem, 77
- electronic passports, 5
- ElGamal cryptosystem, 75
- elliptic curves, 24
- encrypted domain, 7–9, 19, 20, 34, 56, 60, 78, 89, 90, 98, 118, 120
- encrypted sketch, 97
- Eric Schimdt, 4
- Eric Schmidt, 3
- Facebook, 4, 5
- FBI, 51
- FingerCode, 12, 13, 24, 45, 59–61, 67–73, 77, 78, 81, 83–87, 90, 93, 118, 119
- FingerCode system, 71
- FingerCodes, 67, 78
- Fuzzy Commitment, 118
- Fuzzy Commitment Scheme, 12, 13, 93–97, 99, 101, 115, 118
- fuzzy sketch, 94, 98, 109
- Gabor filter, 63, 64
- Gabor filters, 60, 63, 64, 72
- garbled circuits, 8, 9, 32, 34
- garbled table, 33
- Gentry, 8, 19, 20, 120
- Goldwasser-Micali cryptosystem, 99

- Google, 3
- homomorphic cryptosystem, 75
- homomorphic cryptosystems, 8, 11, 13, 19, 70, 98
- Homomorphic encryption, 9
- homomorphic encryption, 7, 8, 10, 18–20, 30, 33, 34, 55, 87, 90
- homomorphic properties, 13, 19, 24, 30, 31, 79, 100, 117
- homomorphic property, 108, 115
- honest but curious, 21, 23, 73, 78, 83, 84, 110, 112, 118–120
- IND-CPA, 21, 22, 26, 99, 105, 108
- Intelligence Agency, 5
- Internet, 3, 4, 38
- iriscodes, 71
- Jurik, 8
- LinkedIn, 4
- Moore's Law, 4
- multi party computation, 7, 18, 20, 22, 90, 120
- multi party omputation, 115
- MySpace, 4
- NIST, 27
- Pailler, 8
- Pailler cryptosystem, 8
- Paillier, 24, 27, 29–31, 73, 75, 76, 78, 80, 99, 100
- Paillier cryptosystem, 13, 24–27, 29, 73, 79, 99, 100, 117
- privacy preserving, 10–14, 21, 23, 24, 30, 34, 35, 59, 73, 84, 85, 90, 94, 97–99, 106, 115, 117–121
- privacy preserving sketch, 119
- Private Information Retrieval, 99
- RSA, 8, 29, 99
- RSA moduli, 72
- RSA modulo, 29
- RSA modulus, 25, 26
- secure function evaluation, 18, 32
- secure sketch, 11, 55, 94
- secure sketches, 98
- semantic secure, 99
- semantic security, 21
- semantically secure, 83, 99
- September 11, 5
- sketch, 55, 94, 97
- symmetric, 9, 32, 63, 72
- Twitter, 4
- two party computation, 18
- Viviane Reding, 5
- Wall Street Journal, 4
- Yao, 9, 18, 32

Publication List



Accepted

- M. Barni, **P. Failla**, R. Lazzeretti, A.R. Sadeghi, T. Schneider, – *Privacy Preserving ECG Classification with Branching Program and Neural Networks*, – IEEE Transactions on Information Forensics and Security

2010

- (8) **P. Failla**, M. Barni, – *Gram - Schmidt Orthogonalization on Encrypted Vectors*, – 21st International Tyrrhenian Workshop on Digital Communications, ITWDC 2010, *Island of Ponza – Italy – 6-8 September 2010*
- (7) M. Barni, D. Catalano, M. Di Raimondo, R. Donida Labati, **P. Failla**, T. Bianchi, – *A Privacy-compliant Fingerprint Recognition System Based*

on Homomorphic Encryption and Fingercod Templates – IEEE International Conference on Biometrics: Theory, Applications and Systems, IEEE BTAS 2010, Arlington, Virginia – USA – 27-29 September 2010

- (6) **P. Failla**, Y. Sutcu, M. Barni, – *eSketch: a Privacy-Preserving Fuzzy Commitment Scheme for Authentication using Encrypted Biometrics* – ACM Workshop on Multimedia Security, MMSec 2010, Rome – Italy – 9-10 September 2010
- (5) M. Barni, D. Catalano, M. Di Raimondo, R. Donida Labati, **P. Failla**, T. Bianchi, – *Privacy Preserving Fingercod Authentication* – ACM Workshop on Multimedia Security, MMSec 2010, Rome – Italy – 9-10 September 2010
- (4) **P. Failla** – *Heuristic Search in Encrypted Graphs* – SecurWare 2010 – Acceptance Rate 30%, Venezia, Mestre – Italy – 18-25 July 2010

2009

- (3) M. Barni, **P. Failla**, V. Kolesnikov, R. Lazzeretti, A.R. Sadeghi, T. Schneider, – *Combining Signal Processing and Cryptographic Protocol Design for Efficient ECG Classification* – SPEED 2009 Workshop co-located at CHES 2009, Lousanne – Switzerland – 10 September 2010
- (2) M. Barni, **P. Failla**, R. Lazzeretti, A. Paus, A.R. Sadeghi, T. Schneider, V. Kolesnikov – *Efficient Privacy-Preserving Classification of ECG Signals* – WIFS 2009 – Acceptance Rate 32.5%, London – UK – 6-9 December 2009
- (1) M. Barni, **P. Failla**, V. Kolensikov, R. Lazzeretti, A.R. Sadeghi, T. Schneider – *Secure Evaluation of Private Linear Branching Programs with Medical Applications* – ESORICS 2009 – Acceptance Rate 19.1%, Saint Malo – France – 21-25 September 2009

Curriculum Vitae



Pierluigi Failla was born in Grosseto, Italy on November 29, 1981. He graduated from G. Marconi lycee in 2000. He obtained B.Sc. and M.Sc. degrees from Computer Science Engineering Department at University of Siena in October 2005 and September 2007, respectively. In October 2007, he started his Ph.D. in the Department of Information Engineering at University of Siena under the guidance of Prof. Mauro Barni.

During his study he visited, for six months, the group of Information Systems and Internet Security at the Polytechnic Institute of New York University where, under the guidance of Prof. Nasir Memon, he worked developing algorithms and protocols for privacy preserving biometric systems.

His main research interests are in the field of: cryptography and cryptographic protocols, secure multi party computation technique, data security and signal protection with particular regards to privacy preserving algorithm for biomedical and biometric application.

He is a System Analyst in the group of Research and Advanced System Design at Elt Elettronica S.p.A. (Rome, Italy) where his main activities and responsibilities are in the area of Cyber Warfare and Electronic Warfare.

It is commonly known that there is a trade off between the security of the systems based on biometric solutions and the privacy of the biometric data itself. In particular, the technologies behind practical privacy preserving algorithms and protocols belong to several different disciplines including signal processing, cryptography, information theory, each of which with a long standing tradition of theoretical and practical studies. At the same time, only few is known about their joint use, both at a theoretical and a practical level, the separation-paradigm being by far the most popular approach. The main goal of this thesis is to provide privacy preserving solutions to handle biometric samples avoiding the leakage of information that is intrinsic in the existing approaches and guaranteeing the privacy of the users.



The Ph.D. School of Information Engineering of the University of Siena is a school aiming at educating scholars in a number of fields of research in the Information Engineering area. The Ph.D. School of Information Engineering is part of the Santa Chiara High School of the University of Siena. A Scientific Committee of external experts recognized Ph.D. Schools belonging to Santa Chiara as excellent, according to their degree of internationalization, their research, and educational activities.