UNIVERSIDADE DE VIGO

ESCOLA DE
ENXEÑARÍA DE TELECOMUNICACIÓN

Ph.D. programme in Signal Theory and Communications

Ph.D. Thesis

Submitted for the European Doctor mention

ENCRYPTED DOMAIN PROCESSING FOR SIGNAL PROCESSING APPLICATIONS

Author: Juan Ramón Troncoso-Pastoriza
Advisor: Fernando Pérez-González

2012

Abstract

In modern society, digital data about individuals that could be considered to be highly personal, can be found relatively easily in the communication networks, especially the Internet. Although most people support the last decades' advances in digital networks, the sensitivity of these data motivates an increasing concern about the public availability of personal data and the processing performed on them. On the other hand, signal processing researchers have traditionally focused on continuously improving the efficiency and robustness of the applied algorithms, while often leaving aside the crucial aspect of data privacy. Thus, advances in signal processing have not taken into account the trustworthiness of the parties that manage users' signals or the sensitivity of the information contained within these signals. There are many application scenarios where the need for privacy is clearly present, mainly those in which biological signals (fingerprints, faces, iris, DNA, ECG signals, MRI images,...) are involved, as they hold extremely sensitive information about users or patients, and their privacy is traditionally addressed through written consents that represent the trust that users must put on the party or parties that process their signals.

Signal Processing in the Encrypted Domain (SPED) is an emergent research field that has arisen to effectively tackle the privacy problems involving signal processing. As an interdisciplinary area, it has faced from its birth the challenge of bringing together the views of the cryptographic and the signal processing communities in order to reach the target of efficiently applying privacy preserving techniques to common signal processing operations.

This thesis presents novel research work performed within the field of SPED to provide new general-purpose privacy-preserving primitives, establish a whole new framework in nuclear privacy-sensitive signal processing applications like adaptive filtering, solving problems like the omnipresent cipher blow-up in encrypted iterative processing and explicitly analyzing the existing trade-offs among time efficiency, bandwidth and error propagation in these solutions. Other privacy-aware application scenarios mainly dealing with biometric and biological signals are also tackled, providing efficient novel protocols for protecting the privacy of the signals' owners. The work presented in this thesis provides a comprehensive comparison between different approaches for tackling encrypted processing, like homomorphic encryption, garbled circuits, interactive protocols and zero-knowledge proofs, and finishes proposing an extension to a recently presented fully-homomorphic cryptosystem, showing the potential of this approach in achieving fully noninteractive privacy-preserving outsourced processing.

Acknowledgments

Al ver por vez primera la hoja todavía en blanco de los agradecimientos, ya con el resto de esta tesis escrita, soy consciente de la dificultad que alberga rellenar esta página con las palabras adecuadas, no porque no tenga a quien agradecer el haber llegado a este punto, sino por la casi total certeza, por olvido u omisión, de no mencionar a todas las personas que aquí deberían aparecer. Quien no me conoce sabe que soy generalmente parco en palabras, pero esta vez trataré de no escatimar en párrafos y no exluir a nadie; empezaré por orden de relevancia y causalidad en lo referente a la consecución de esta meta que se refleja en las páginas que tienes entre manos.

No puedo sino comenzar por agradecer infinitamente a mis padres que hayan estado siempre ahí, apoyándome y prestándome su ayuda en las decisiones que he tomado, incluso en la de embarcarme en la alocada tarea de intentar hacer una tesis doctoral. Éste es un paso más que doy gracias a vosotros y nunca dejaré de estar agradecido por permitirme tener la capacidad y las posibilidades de darlo. También le agradezco a mi hermano, que insensatamente ha escogido recorrer el mismo camino estudiantil (a su manera), que haya aguantado estoicamente durante estos años las rarezas que a uno le confiere haber cursado la carrera de Telecomunicación... Ya eres uno de los nuestros.

La persona que ha sabido encauzar mi período doctoral por los caminos adecuados, y sin la cual no habría podido no sólo terminar sino ni siquiera empezar esta etapa, es mi tutor, Fernando Pérez González. A él le agradezco su dedicación, motivación, interés y empuje para llevar a término esta tesis. No he conocido a una persona más volcada en su trabajo y en la consecución de las metas propuestas y, por lo tanto, no he conocido a ninguna persona que haya cosechado tantos éxitos en las empresas acometidas. Gracias por apoyarme y confiar en mí desde aquella práctica de FCD hasta hoy.

Next is Stefan, who guided me during my stay at Philips; thank you for being so close and being always willing and determined to keep on advancing, even when facing the most difficult research problems. My research work and my thesis would not have been the same without that period under your guidance.

Quiero agradecer a todos los miembros del Grupo de Procesado de Señal en

Comunicaciones el hecho de haberme acogido entre ellos y haberme prestado ayuda en los momentos necesarios; en especial a Pedro, siempre dispuesto a debatir y rebatir cualquier tema y con el que pude compartir muchas jornadas de redacción de propuestas de proyecto. También David, Gabi, Nuria, Roberto, Carlos, Carmen y todos los que habéis sufrido mi "gestión" de los servidores; gracias.

No puedo olvidarme de los antiguos miembros de nuestro grupo, Luis y Dani, que siempre han estado dispuestos a colaborar y a participar en intensas tardes de brainstorming y de escritura acelerada de artículos.

También debo agradecer a mis compañeros de laboratorio su abnegada resignación ante mi habitual incontinencia verbal y mi constante sarcasmo. No voy a citar nombres porque no quiero hacer distinciones ni ordenaciones; todos vosotros sabéis que os estoy agradecido.

Por último, y para no alargarme demasiado ni contravenir mi omnipresente principio de no decir más de lo necesario, agradezco a todos mis amigos el que hayan estado ahí apoyándome y soportándome cuando los necesitaba, y a todos aquéllos que han puesto piedras en mi camino, el que me hayan ayudado a prepararme ante las adversidades futuras.

Y termino citando una frase del filósofo austríaco Ludwig Wittgenstein, que se puede interpretar en todos los sentidos de los términos mencionados:

"*Knowledge is in the end based on acknowledgement*"

Contents

List of used Acronyms and Abbreviations

- AEPD Spanish Data Protection Agency
- ALE Adaptive Line Enhancer
- API Application Programming Interface
- AWGN Additive White Gaussian Noise
- BLMS Block Least Mean Square
- BNSA Blind Newton Sensitivity Attack
- CCTV Closed-Circuit TeleVision
- CLT Central Limit Theorem
- DARPA Defense Advanced Research Projects Agency
- DCT Discrete Cosine Transform
- DFT Discrete Fourier Transform
- DNA DesoxyriboNucleic Acid
- DOA Direction Of Arrival
- DRM Digital Rights Management
- DSP Digital Signal Processor
- DWR Data to Watermark Ratio
- ECG ElectroCardioGram
- EPO European Patent Office
- ER Event Report
- ERR Event Report Request
- ESPRIT Estimation of Signal Parameters via Rotational Invariance Technique
- FSM Finite State Machine
- GC Garbled Circuits
- GG Generalized Gaussian

- GGBA Generalized Gaussian Watermarking with Binary Antipodal Spreading Sequence
- GMP GNU Multiple Precision Arithmetic Library
- GSC Generalizable Sidelobe Cancelers
- HE Homomorphic Encryption
- HNF Hermite Normal Form
- IP Internet Protocol
- IPMP Intellectual Property Management and Protection extension
- LFW Labeled Faces in the Wild
- LMS Least Mean Square
- LOPD Spanish Organic Act on Personal Data Protection
- ML Maximum Likelihood
- MRAC Model-Reference Adaptive Control
- MRI Magnetic Resonance Image
- MSE Mean Square Error
- MUD MultiUser Detection
- MUSIC MUltiple SIgnal Classification
- NoE Network of Excellence
- NWR Noise to Watermark Ratio
- ONVIF Open Network Video Interface Forum
- OT Oblivious Transfer
- PCT Patent Cooperation Treaty
- PIR Private Information Retrieval
- PRM Privacy Risks Management
- RBF Radial Basis Function
- REL Rights Expression Language
- RLS Recursive Least Squares

- ROC Receiver Operating Characteristic
- RRP Reproducible Research Paradigm
- RSA Rivest-Shamir-Adleman
- SFE Secure Function Evaluation
- SLE System of Linear Equations
- SMC Secure Multipaty Computation
- SOAP Simple Object Access Protocol
- SPED Signal Processing in the Encrypted Domain
- SPEED Signal Processing in the EncryptEd Domain project
- SS Spread Spectrum
- ST-DM Spread Transform Dither Modulation
- SVD Singular Value Decomposition
- SVM Support Vector Machine
- TAMI Transparent Accountable Datamining Initiative
- USPTO United States Patent and Trademark Office
- VPH Virtual Physiological Human
- WNR Watermark to Noise Ratio
- WSDL Web Service Description Language
- XM2VTS eXtended MultiModal Verification for Teleservices and Security applications
- XML eXtended Markup Language
- ZK Zero-Knowledge

Notation

The notation along the chapters of this thesis is the following, unless otherwise stated: lowercase letters will be used indistinctly to represent classes in a ring $(\mathbb{Z}_n, +, \cdot)$ and a representative of that class in the interval $[0, n)$. $\lceil . \rfloor$ will represent the rounding function of a number to the nearest integer. $[a]_d$ represents the reduction of $a \mod d$.

The used vectors will be column vectors of size L unless otherwise stated, and they will be represented by lower-case boldface letters, whereas matrices will be represented by upper-case boldface letters. In some chapters, vector notation $\boldsymbol{a} = [a_0, \ldots, a_{n-1}]$ and polynomial notation $a(x) = \sum_{i=0}^{n-1} a_i \cdot x^i$ will be used indistinctly when appropriate. An element at row i and column j of a matrix $\boldsymbol{A}$ will be denoted indistinctly by $A(i, j)$ or $A_{i,j}$; $(.)^T$ will denote matrix/vector transpose, and $\boldsymbol{A}' = \{a_{i,j}\}_{r,s}^{t,u}$ represents the submatrix of $\boldsymbol{A}$ of size $(t - r + 1) \times (u - s + 1)$, defined by $a'_{i,j} = a_{i+r,j+s}$. Scalar random variables will be denoted by italicized letters, and upper-case calligraphic letters will represent sets or parties participating in a protocol.

When the used encryption system is not relevant, the encryption of a number x will be represented by $[\![x]\!]$, and the vector (matrix) formed by the encryptions of the vector $\boldsymbol{x}$ (matrix $\boldsymbol{X}$) will be represented by $[\![\boldsymbol{x}]\!]$ ($[\![\boldsymbol{X}]\!]$). When working with the binary representation of a number x, the encryption of the vector of binary bits of that representation will be denoted as $[\![x]\!]_b$. When a specific cryptosystem is used, $E_s(x)$ and $D_s(x)$ will be used for encryption and decryption respectively, being the subindex s representative for the used cryptographic algorithm.

The operations performed between encrypted and clear numbers will be indicated as if they were performed in the clear; e.g. $[\![\boldsymbol{X}]\!] \cdot \boldsymbol{b}$ will represent the encryption of $[\![\boldsymbol{X} \cdot \boldsymbol{b}]\!]$. Regarding the complexity calculations, the complexity of basic modular operations, like additions (A), products (P) and exponentiations (X) will be denoted by $\text{Cpx}_A, \text{Cpx}_P, \text{Cpx}_X$ respectively, prefixing an E (i.e. EA, EP, EX) when they are performed under encryption. The computational complexity for encryptions and decryptions will be denoted by Cpx_E and Cpx_D. The factor $ct < 1$ will denote the ratio between the size of a clear-text value and that of an encrypted value. When needed, the subscript cm will denote communication complexity, measured in number of bits, while cp will indicate computational complexity, with an indication of the party whose complexity is represented. Finally, the expression $a \in_R A$ denotes the random choice of a value a from the set A with uniform distribution.

Chapter 1

Introduction

In modern society, digital data about individuals can be found relatively easily in the communication networks, especially the Internet. Although people supports the last decades' advances in digital networks, the sensitivity of these data motivates the raise of an increasing concern about the public availability of personal data, and the processing performed on them. Focusing on the European case, this concern has been reflected in a series of Directives, dealing with the protection of individuals' personal data ([10, 12]). Directive 95/46/EC deals with *the protection of individuals with regard to the processing of personal data and on the free movement of such data*, where *personal data* means *any information relating to an identified or identifiable natural person.* One of the main mottoes of this Directive is that data processing systems must respect the fundamental rights and freedoms, specially in those aspects concerning the right to privacy.

Leaving aside the legal framework, and turning into the technical support to privacy principles, conventional cryptographic protocols deal with the problem of protecting some private information from an unauthorized third party that otherwise could modify or have access to the information. In the scenario of secure processing, where the privacy must be preserved not only against a third party, but also against the parties that process the data, secure multiparty computation constructions can be used. Nevertheless, typical multiparty computation protocols become too costly in terms of computation and communication complexity for real-world scenarios.

This is the context in which the emerging field of *Signal Processing in the Encrypted Domain* arose. This discipline tries to address the problem of efficiently processing signals in untrusted environments, where not only the communication channel between parties is insecure, but also the parties that perform the computation are not trusted.

1.1. Signal Processing in the Encrypted Domain

Signal processing is an area that comprises many techniques for the representation, analysis, transmission and restoration of signals. Among the most relevant problems in this field, we can point out the filtering problem: a filter or estimator is a system designed to extract information about a quantity of interest from noisy data. Filters and signal processing operations in general are nowadays ubiquitous, having an extremely broad field of application, ranging from voice processing performed by an embedded microcontroller in a mobile phone to complex surface and volume rendering and texturing for 3D animation films, passing through the presentation and automated analysis of MRI (magnetic resonance) images.

We live surrounded by electronic devices that perform signal processing, and they have become essential in our every-day lives; thus, their presence can have a strong impact on our privacy. In fact, some of the many application scenarios of signal processing involve contexts in which the processed signals are highly sensitive and present strong privacy constraints. Hence, in the last years there has been a growing interest from the scientific community in applying privacy preserving techniques to common signal processing operations; this emergent research field has been named *Signal Processing in the Encrypted Domain (SPED)* [17, 9].

Signal processing researchers have traditionally been focused on continuously improving the efficiency and robustness of the applied algorithms, while leaving aside the crucial aspect of data privacy. Thus, signal processing was not aware of the trustworthiness of the parties that manage users' signals or the sensitivity that the information contained within these signals might have. There are many application scenarios where the need for privacy is clearly present, mainly those in which biomedical or biometric signals (fingerprints, faces, iris, DNA, ECG signals, MRI images,...) are involved, as they hold extremely sensitive information about users or patients, and their privacy is traditionally addressed through written consents that represent the trust that users must put on the party or parties that process their signals; in fact, these prototypical scenarios have attracted much of the attention of the SPED community.

The secure treatment of private data is a question of great relevance for individuals as well as for organizations. Informally, data privacy in a system consists in the possibility of hiding certain data to determined users of the system. Nevertheless, traditional cryptosystems only provide an all-or-nothing security, in the sense that they do not offer any kind of access to encrypted data unless the deciphering key is available. Furthermore, conventional cryptographic protocols deal with the problem of protecting some private information from an unauthorized third party that otherwise could modify or have access to the information. When the privacy must be preserved not only against a third party, but also against the parties that participate in the protocol where the inputs are shared, more involved solutions come into play.

Signal processing in the Encrypted Domain was born in order to efficiently cover the needs of privacy for data that must be handled by authorized but non-trusted third parties, providing a greater malleability than traditional cryptosystems, as it allows different levels of access to the data, in order to perform certain operations on them without the need of decryption and, thus, without having access to their clear-text version. Hence, it addresses the problem of processing signals in untrusted environments, where not only the communication channel between parties is insecure, but also the parties that perform the computation are not trusted. This groundbreaking concept of processing has been a hot topic during the last few years, although the theoretical principles on which it is founded date back to more than two decades ago. In this period, there have been plenty of contributions aiming towards the target of performing secure processing of data by a third party, preserving the required privacy level.

1.1.1. State of the Art in SPED

Signal Processing in the Encrypted Domain is an interdisciplinary research area that joins the efforts of the signal processing and the cryptographic communities in order to provide solutions for the privacy and security problems in signal processing applications, with a special interest in the efficiency of the implementations.

The theoretical grounds on Signal Processing in the Encrypted Domain come from the field of secure function evaluation, that was firstly introduced by Yao in 1982 [245] (Secure two-party computation) through the now widely known *Millionares' problem*, and then generalized to Secure Multiparty Computation [107] (SMC). In the former setting, two millionaires wish to know who is the richest, without disclosing to the other their respective wealth. The presented solution employed the concept of *garbled circuits*, with which it is possible to execute a function, expressed as a combination of binary gates, on certain data (also in a binary representation) that belong to two (or more) parties. After the execution of a garbled circuit, both parties can obtain the result of the computation, but none of them have access at any time to the data belonging to the other party. In spite of the generality of the presented solution, the inefficiency of its implementation would constitute the biggest obstacle for the development of this technology during the following years, in such a way that the existence of efficient solutions for the secure execution of a generic function is yet nowadays an open problem. Nonetheless, many efficient and secure techniques have been developed for specific applications in the past few years, building up a set of tools that foretell the potential of this technology.

After the initial proposal of Yao up to the present, there have been many contributions related to private data processing [174]. Without intending to provide a thorough survey, a classification of the technologies nowadays available for the implementation of SPED, summarized in seven categories, is presented in order to give a glimpse of the elements that build up the current landscape on approaches to SPED[1].

1.1.1.1. Secure Multiparty Computation

Secure Multiparty Computation [245, 107, 109] (SMC) stems directly from the initial Yao's proposal, based on interactive protocols performed by two or more parties that own private data, to which they wish to apply a determined function, known by all the parties. While this approach needs to communicate a great amount of information among the involved parties, adding a significant overhead to the communication required in a non-private solution, it has the advantage of being generic, thus allowing the implementation of a great variety of functions in a distributed fashion. This category includes several basic secure primitives, like Oblivious Transfer [48, 168, 148], that allows to choose one element out of N without disclosing which was the chosen one, or Secret sharing [202], that splits stored data into several parts (shares), in such a way that all the shares are needed for the reconstruction of the original data. SMC techniques have recently received contributions that sensibly improve the computation and communication complexity [141], and that paired with homomorphic encryption can yield very efficient protocols.

Secure Function Evaluation (SFE) is a special case of SMC in which a set of players want to evaluate a function, known to all players, on their private inputs. Subsequently, various approaches to securely evaluating a function have been developed for different function representations, namely combinatorial circuits [107, 245, 130], ordered binary decision diagrams [142], branching programs [167, 166], or one-dimensional look-up tables [166]. Each of these approaches can achieve a practical and efficient oblivious protocol for evaluating a given function f, if f can be expressed in a space-efficient manner in the chosen representation.

1.1.1.2. Homomorphic Encryption

At a conceptual level, arithmetic operations can be performed in the encrypted domain using a privacy homomorphism [49]. This consists in a group homomorphism between clear-text data and encrypted data, allowing the execution of

[1] There are some surveys available in this area [17, 9], to which we refer the reader for further information.

operations (generally sums or products) on encrypted data, without the need of deciphering such data. Formally [190]:

Definition 1 (Privacy homomorphism) *Let S be a set, and S' a possibly different set, with the same cardinality as S. Let $D : S' \to S$ a bijection, called* decryption function*. Let U be an algebraic system of operations in clear-text given by:*

$$U = (S; f_1, ..., f_k; p_1, ..., p_l; s_1, ..., s_m),$$

where $f_i : S^{g_i} \to S$ are functions with g_i parameters, p_i are predicates with h_i parameters, and s_i are constants.

Let C be the counterpart of U for the calculations with encrypted data:

$$C = (S'; f'_1, ..., f'_k; p'_1, ..., p'_l; s'_1, ..., s'_m).$$

The mapping D is called a privacy homomorphism *if it satisfies the following conditions:*

- $(\forall i)(1 \leq i \leq k \Rightarrow (\forall (a_1, ..., a_{g_i}) \in S'^{g_i})$
  $(\exists c \in S')(f'_i(a_1, ..., a_{g_i}) = c \Rightarrow f_i(D(a_1), ..., D(a_{g_i})) = D(c)))$.
- $(\forall i)(1 \leq i \leq l \Rightarrow (\forall (a_1, ..., a_{h_i}) \in S'^{h_i})$
  $(p'_i(a_1, ..., a_{h_i}) \Rightarrow p_i(D(a_1), ..., D(a_{h_i}))))$.
- $(\forall i)(1 \leq i \leq m \Rightarrow D(s'_i) = s_i)$.

Furthermore, the following conditions must be satisfied for C and D to be useful as a protection:

- D and D^{-1} must be easily computable.
- The functions f'_i and the predicates p'_i in C must be efficiently computable.
- D^{-1} is a cipher without expansion or a cipher with an expansion such that the cryptotext representation is only marginally bigger than the corresponding clear text.
- The operations and predicates in C must not suffice to obtain an efficient way of calculating D.

There are certain semantically secure encryption algorithms [177, 82, 175, 111] that present a private homomorphism, allowing the execution of one operation (generally sums or products) on encrypted data, without the need of deciphering such data (cf. ArticleID 13801 in [17]). The most commonly used homomorphic cryptosystem is Paillier's, that presents an additive homomorphism (cf. Section 2.2.2.1).

These cryptosystems present several decisive advantages, such as the drastic reduction of the overhead in the communication required among parties, the efficiency of the computation, and the automatic provision of privacy while data are being processed, as they never leave their encrypted state. As a counterpart, the amount of operations that homomorphic encryption allows is restricted (either sums on encrypted data and products of ciphered data and known data, or products between encrypted data and exponentiations of ciphered data to known data, but not all of them simultaneously). There are, though, some recent contributions by Gentry [104, 105], that are able of executing any circuit without the need of decryption, through a full homomorphism. It does so through a cryptosystem based on ideal lattices with bootstrappable decryption, for which he shows that it achieves a full homomorphism. While this proposal is definitely promising, it is still not practical, due to the huge size needed for the ciphertexts. In fact, the existence of practical fully homomorphic cryptosystems is still an open problem, but there is a whole research line currently underway, with works like [208], focused on translating Gentry's scheme into a practical fully homomorphic solution, but it is still limited to very small plaintexts and very simple circuits. In this thesis we will adhere to using an additively homomorphic cryptosystem, always taking into account the advantages that an efficient and practical fully homomorphic cryptosystem would provide, and devote the last chapter to the extension and efficient use of a quasi-full homomorphism.

1.1.1.3. Commitment Schemes

Commitment schemes [76] are cryptographic tools that, given a common public parameter par_{com}, allow that one party of a protocol choose a determined value m from a finite set M and commit to his choice $C_m = \mathrm{Com}(m, r, par_{\mathrm{com}})$, such that he cannot modify it during the rest of the protocol; the committed value is not disclosed to the other party, thanks to the randomization produced by r, which constitutes the secret information needed to open the commitment.

The required security properties that the commit function must fulfill are *binding* and *hiding*; the first one guarantees that once produced a commitment C_m to a message m, the committer cannot open it to a different message m'; the second one guarantees that the distribution of the commitments to different messages are indistinguishable, so one commitment does not reveal any information about the concealed message. Each of these properties can be achieved either computationally or in an information-theoretic sense, but the information-theoretic version cannot be obtained for both properties at the same time.

The commitment scheme used throughout the present thesis, unless otherwise stated, is Damgård-Fujisaki's scheme [78], that provides statistically-hiding and computationally-binding commitments, based on Abelian groups of hidden order. Given the security parameters F, B, τ and k, the common parameters are a modulus n (that can be obtained as an RSA modulus), such that the order of $\mathbb{Z}_n^*$ can be upper bounded by 2^B, a generator h of a multiplicative subgroup of high order (the order must be F-rough) in $\mathbb{Z}_n^*$, and a value $g = h^\alpha$, such that the committer knows neither α nor the order of the subgroups. The commit function of a message $x \in [-\tau, \tau]$ with a random value $r \in [0, 2^{B+k}]$ takes the form $C_x = g^x h^r \mod n$.

Additionally, this commitment scheme presents an additive homomorphism that allows computing the addition of two committed numbers ($C_{x+y} = C_x \cdot C_y \mod n$) and the product of a committed number and a public integer ($C_{ax} = C_x^a \mod n$).

1.1.1.4. Interactive Proof Systems

Interactive proof systems were introduced by Goldwasser *et al.* [113]; they are two-party protocols in which a Prover $\mathcal{P}$ tries to prove a statement x to a Verifier $\mathcal{V}$, and both can make random choices. The two main properties that an interactive protocol must satisfy are *completeness* and *soundness*; the first one guarantees that a correct Prover $\mathcal{P}$ can prove all correct statements to a correct Verifier $\mathcal{V}$, and the second, that a cheating Prover $\mathcal{P}^*$ will only succeed in proving a wrong statement with negligible probability.

A special class of interactive protocols are Proofs of Knowledge [162], in which the proved statement is the knowledge of a witness that makes a given binary relation output a true value, such that a probabilistic algorithm called *knowledge extractor* exists, and it is able to output a witness for the common input x using any probabilistic polynomial time Prover $\mathcal{P}^*$ as an oracle, in polynomial expected time (*weak soundness*).

1.1.1.5. Zero-knowledge Protocols

Zero-knowledge protocols [113, 108, 109] allow, through the interaction between two parties (a Prover $\mathcal{P}$ and a Verifier $\mathcal{V}$), that one of them prove the validity of an statement without disclosing any additional knowledge (zero-knowledge) besides that directly derived from the proven statement. More formally, an Interactive Proof System $(\mathcal{P}, \mathcal{V})$ is statistically zero-knowledge if it exists a probabilistic polynomial algorithm (simulator) $S^{\mathcal{V}}$ such that the conversations produced by the real interaction between $\mathcal{P}$ and $\mathcal{V}$ are statistically indistinguishable from the outputs of $S^{\mathcal{V}}$.

These protocols are commonly used in combination with other techniques to prove that the operations to which data are subjected are correct, even when one has no access to such data. The main advantage of this approach consists in that it can be used to avoid any attempt of accessing the private data by malicious users, and it allows to perform more complex operations than those allowed by homomorphic encryption. The counterpart comes as an increase in the computational and communication burden needed for performing a given operation.

1.1.1.6. Data obfuscation

Data obfuscation [35] consists in adding some kind of noisy (random) signal to data in order to partially or totally conceal them, in such a way that some relatively complex operations can be performed on them, provided that the original meaning of the data is preserved. These methods have the advantage of allowing complex operations with a relatively low computational and communication burden, at the expense of privacy, as some information about the obfuscated data can be inferred from the intermediate steps of the process.

1.1.1.7. Searchable encryption

Searchable encryption [46] is a cryptographic primitive that allows for the check of a match between a given pattern inside encrypted data. It is mainly used for keyword searches in encrypted databases, and presents the advantage of conveniently protecting the performed queries in these cases, but as a counterpart, it is not very flexible, and it also presents some scalability issues. Nevertheless, the area of searchable encryption has been identified by DARPA as one of the technical advances that can be used to balance the need for both privacy and national security in information aggregation systems [13].

There are several setups in which the previous techniques can be used, and each one has its advantages and drawbacks in terms of bandwidth and computation efficiency, but they cannot be directly compared in general, as they serve for different purposes. Even though there are no generic solutions available yet that can be applied efficiently to any case [69], allowing the execution of any function on encrypted data, the previous approaches can be combined and extended, taking into account the specific requirements of a determined application, in order to obtain a solution that provides a simultaneously efficient and secure implementation. Currently, research efforts in the field of Signal Processing in the Encrypted

Domain are focusing on this direction, giving birth to numerous contributions to efficiently and securely process data and signals in many application areas, which are briefly described in the next section.

1.1.2. Application Areas

Among the most representative application areas of Encrypted Data Processing, the following are particularly relevant:

- **Secure Medical Diagnosis** [221, 39]: patients' medical data are extremely sensitive, and the execution of secure medical processing on those data in order to obtain a totally confidential diagnosis is one of the main applications of SPED. As an example, [221] presents a system for performing approximate searches on DNA sequences in order to seek for genetic diseases keeping both the DNA sequence and the search method secret.

- **Private Biometric Authentication** [89, 196]: biometric applications are also one of the paradigmatic signal processing scenarios where sensitive signals are handled. Biometric signals (faces, iris, fingerprints,...) are unreplaceable keys that users are not willing to give away for the purpose of a secure authentication. Achieving a balance between secure authentication and private biometric processing is one of the challenges in this field.

- **Secure Database Queries** [158, 50]: The scenario in which a non-trusted server stores an encrypted database and allows efficient private queries on the stored data is one of the most studied use cases of SPED. There are many proposed protocols for accessing encrypted data in a server, as searchable encryption [51, 50], with mechanisms based on trapdoor functions, secret sharing or homomorphic encryption.

- **Private Information Retrieval** [158]: This application covers the scenario in which users wish to access a (centralized or distributed) public database, but keeping their queries secret, such that the server or servers must process an encrypted query. The proposed solutions to this problem are mainly based in Oblivious Transfer.

- **Zero-knowledge based Watermarking and Fingerprinting systems** [230, 224]: Practical watermarking systems have always used symmetric key for embedding and for detection/decoding of the watermark. Zero-knowledge allows for a private detection/decoding, effectively proving that the watermark is present inside some asset (image, audio, video) without disclosing the secret key.

- **Secure Data Mining** [145, 244, 28]: This is by far the most studied field of application of Private Data Processing, and it consists in extracting knowledge from data that are distributed among several users who, due to privacy concerns, do not want to share their data in the clear. A number of works in this area are available, ranging from secure clustering algorithms [132], classification [233] or scalar products [211], to computation of correlation matrices [149], private execution of neural networks [176], and many more.
- **Private Function Evaluation** [52]: this is one of the most promising fields for Encrypted Data Processing, where the aim is to provide an environment in which a user executes a function on someone else's data, in a secure and private fashion for everyone. While currently developed functions are still low-complexity primitives and algorithms, it is foreseeable that Private Function Evaluation can provide the grounds for totally private remote computation and data outsourcing.
- **Encrypted Signal Processing**: Signal processing is one of the latest fields to which encrypted data processing has been incorporated, but there are already some initial progresses, such as private algorithms for the implementation of several discrete transforms (DFT, DCT) [44], which are widely used in audio/image/video processing.

The subsequent chapters of this thesis show some recent developments in the most promising application fields named above. Furthermore, the recent field of Cloud Computing, that has also a section devoted to it, has to be added to the previous ones. The increasing trend towards remote and distributed computing opens a new and wide range of applications to which SPED can provide an exceptional tool for achieving the required level of privacy and malleability, allowing for an environment of guaranteed trust in which users can develop their work securely and privately.

1.2. Contributions and publications

In the following, a brief summary of the contributions contained in this thesis, together with the published works and patents that support its research value and its technological transfer value, is provided.

1.2.1. Publications

The publications and patent applications that support the research undertaken during the period of this thesis are the following:

1.2.1.1. Journal papers

J1 Juan Ramón Troncoso-Pastoriza, Daniel González Jiménez, and Fernando Pérez-González. *Fully Private Noninteractive Face Verification.* Submitted to IEEE. Trans. on Information Forensics and Security, 2012.

J2 Juan Ramón Troncoso-Pastoriza and Fernando Pérez-González. *Secure Adaptive Filtering.* IEEE Trans. on Information Forensics and Security, 6(2):469-485, June 2011.

J3 Juan Ramón Troncoso-Pastoriza and Fernando Pérez-González. *Efficient zero-knowledge watermark detection with improved robustness to sensitivity attacks.* EURASIP Journal on Information Security, 2007. Special Issue on Signal Processing in the Encrypted Domain.

J4 Luis Pérez-Freire, Pedro Comesaña, Juan Ramón Troncoso-Pastoriza, and F. Pérez-González. *Watermarking security: a survey.* LNCS Transactions on Data Hiding and Multimedia Security I, 4300:41-72, October 2006.

1.2.1.2. Conference papers

C1 Juan Ramón Troncoso-Pastoriza and Fernando Pérez-González. *Fully Homomorphic Faces.* Submitted to IEEE International Conference on Image Processing, 2012.

C2 Daniel A. Rodríguez-Silva, F. Javier González-Castaño, Lilian Adkinson-Orellana, Alexandre Fernández-Cordeiro, Juan Ramón Troncoso-Pastoriza, and Daniel González-Martínez. *Encrypted Domain Processing for Cloud Privacy: Concept and Practical Experience.* In International Conference on Cloud Computing and Services Science (CLOSER 2011), Noordwijkerhout, The Netherlands, May 2011.

C3 Juan Ramón Troncoso-Pastoriza and Fernando Pérez-González. *Efficient Protocols for Secure Adaptive Filtering.* In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2011), pages 5860-5863, Prage, Czech Republic, May 2011. IEEE.

C4 Juan Ramón Troncoso-Pastoriza and Fernando Pérez-González. *CryptoDSPs for Cloud Privacy.* In CISE 2010, volume 6724 of LNCS, Hong Kong, China, December 2010.

C5 Juan Ramón Troncoso-Pastoriza and Fernando Pérez-González. *Secure and Private Medical Clouds using Encrypted Processing.* In VPH 2010, Brussels, Belgium, October 2010.

C6 Juan Ramón Troncoso-Pastoriza, Daniel González-Jiménez, and Fernando Pérez-González. *A new model for Gabor Coefficients' Magnitude in Face Recognition.* In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2010), Dallas, USA, March 2010. IEEE.

C7 Juan Ramón Troncoso-Pastoriza and Fernando Pérez-González. *Skewed Log-Stable model for natural images pixel block-variance.* In IEEE International Conference on Image Processing (ICIP'09), Cairo, Egypt, November 2009. IEEE.

C8 Juan Ramón Troncoso-Pastoriza, Pedro Comesaña, Luis Pérez-Freire, and Fernando Pérez-González. *Videosurveillance and privacy: covering the two sides of the mirror with DRM.* In ACM Workshop on Digital Rights Management, Chicago, IL, USA, November 2009. ACM.

C9 Juan Ramón Troncoso-Pastoriza, Pedro Comesaña, and Fernando Pérez-González. *Secure Direct and Iterative Protocols for Solving Systems of Linear Equations.* In SPEED Workshop 2009, pages 122-141, Lausanne, Switzerland, September 2009.

C10 Juan Ramón Troncoso-Pastoriza, Stefan Katzenbeisser, and Mehmet Celik. *Privacy preserving error resilient DNA searching through oblivious automata.* In 14th ACM Conference on Computer and Communications Security, pages 519-528, Alexandria, Virginia, USA, October 29-November 2 2007. ACM Press.

C11 Juan Ramón Troncoso-Pastoriza, Stefan Katzenbeisser, Mehmet Celik, and Aweke Lemma. *A secure multidimensional point inclusion protocol.* In 9th ACM Workshop on Multimedia and Security (MMSEC'07), pages 109-120, Dallas, Texas, USA, September 2007.

C12 Juan Ramón Troncoso-Pastoriza and Fernando Pérez-González. *Efficient Non-Interactive Zero-Knowledge Watermark Detector Robust to Sensitivity Attacks.* In Edward J. Delp III and Ping W. Wong, editors, Security, Steganography, and Watermarking of Multimedia Contents IX, San Jose, California, USA, January 2007. SPIE.

C13 Juan Ramón Troncoso-Pastoriza and Fernando Pérez-González. *Zero-Knowledge watermark detector robust to sensitivity attacks.* In 8th ACM Multimedia and Security Workshop, pages 97-107, Geneva, Switzerland, September 2006. ACM.

1.2.1.3. Patent Applications derived from the work performed in this thesis

P1 *Title*: METHOD AND APPARATUS FOR SECURE IMAGE PROCESSING

USPTO Application No: 61/596151
Filing Date: 08/02/2012
Inventors: J.R. Troncoso-Pastoriza (Vigo, Spain), F. Pérez-González (Vigo, Spain)
Assignee: Gradiant

P2 *Title*: METHOD, APPARATUS AND SYSTEM FOR SECURED ADAPTIVE FILTERING
USPTO Application No: 61/443823
Filing Date: 17/02/2011
Inventors: J.R. Troncoso-Pastoriza (Vigo, Spain), F. Pérez-González (Vigo, Spain)
Assignee: Gradiant

P3 *Title*: CRYPTOGRAPHIC SYSTEM FOR PERFORMING SECURE COMPUTATIONS AND SIGNAL PROCESSING DIRECTLY ON ENCRYPTED DATA IN UNTRUSTED ENVIRONMENTS
USPTO Patent Application No: 61/240177
Filing Date: 04/09/2009
Inventors: J.R. Troncoso-Pastoriza (Vigo, Spain), P. Comesaña-Alfaro (Vigo, Spain), F. Pérez-González (Vigo, Spain)
Assignee: Gradiant

P4 *Title*: CRYPTOGRAPHIC SYSTEM FOR PERFORMING SECURE ITERATIVE COMPUTATIONS AND SIGNAL PROCESSING DIRECTLY ON ENCRYPTED DATA IN UNTRUSTED ENVIRONMENTS
USPTO Patent Application No: 61/240179
Filing Date: 04/09/2009
Inventors: J.R. Troncoso-Pastoriza (Vigo, Spain), P. Comesaña-Alfaro (Vigo, Spain), F. Pérez-González (Vigo, Spain)
Assignee: Gradiant

P5 *Title*: CRYPTOGRAPHIC SYSTEM FOR PERFORMING SECURE ITERATIVE MATRIX INVERSIONS AND SOLVING SYSTEMS OF LINEAR EQUATIONS
USPTO Patent Application No: 61/240181
Filing Date: 04/09/2009
Inventors: J.R. Troncoso-Pastoriza (Vigo, Spain), P. Comesaña-Alfaro (Vigo, Spain), F. Pérez-González (Vigo, Spain)
Assignee: Gradiant

P6 *Title*: CRYPTOGRAPHIC SYSTEM FOR PERFORMING SECURE COMPUTATIONS AND SIGNAL PROCESSING DIRECTLY ON ENCRYPTED DATA IN UNTRUSTED ENVIRONMENTS
EPO Patent Application No: EP10175467
Filing Date: 04/09/2009

Inventors: J.R. Troncoso-Pastoriza (Vigo, Spain), P. Comesaña-Alfaro (Vigo, Spain), F. Pérez-González (Vigo, Spain)
Assignee: Gradiant

P7 *Title*: METHOD AND A SYSTEM FOR PERFORMING AN OBLIVIOUS QUERY ISSUED BY A FIRST PARTY ON A STRING PROVIDED BY A SECOND PARTY
International Application No.: PCT/IB2008/051771
Application Date: 08/05/2007
Inventors: S. Katzenbeisser (NL), J.R. Troncoso-Pastoriza (NL), M.U. Celik (NL)
Assignee: Koninklijke Philips Electronics N.V.

1.2.2. Contributions

The main contributions that stem from the work performed during the research period covered by this thesis can be summarized in the following points, that also indicate between brackets the produced publications related to each of them:

1. A set of novel privacy-preserving primitives has been produced using Signal Processing in the Encrypted Domain techniques, addressing general interest problems like point inclusion, automata execution, solving systems of linear equations, error-resilient approximate matching and searching protocols, together with related subprotocols (mainly zero-knowledge proofs) of independent interest. A thorough time and bandwidth complexity analysis has been performed on these protocols, showing their efficiency as well as their security in the random oracle model with semi-honest adversaries (C9,C10,C11,P3–P7)

2. This work also establishes a framework for the problem of secure adaptive filtering with semi-honest parties, nuclear in signal processing applications with privacy constraints, providing a fair comparison among different proposed strategies in terms of bandwidth, time complexity and error propagation, and solving the cipher blow-up problem through novel quantization interactive subprotocols, as well as proposing efficiency improvements for the used additive homomorphic cryptosystems that do not hinder their security (J2,C3,P2).

3. Several previously identified application scenarios for privacy-preserving signal processing have been tackled, mainly biometric recognition and biomedical signal processing; this work provides novel efficient solutions for DNA

approximate searching with incomplete templates in a privacy preserving way, as well as secure watermarking detection protocols for symmetric key schemes, through the use of zero-knowledge protocols in a detection algorithm showing improved robustness against sensitivity attacks, one of the most powerful attacks against common watermarking systems (J3,J4,C10,C12,C13,P7).

4. A framework and a conceptual high level architecture are provided for private signal processing in Cloud environments, treated as untrusted environments; this framework allows for the possibility of a fully private outsourced processing in Cloudified applications, targeting mainly medical Clouds (C2,C4,C5).

5. This work also presents a novel and original solution for privacy-preserving videosurveillance applications; this solution does not employ encrypted processing, but a DRM system combined with a smart use of video coding standards, to reverse the typical way in which DRM is used and provide an integral privacy protection and user management system for videosurveillance (C8).

6. Finally, future fully private and noninteractive encrypted processing is foreseen through the use of fully-homomorphic cryptosystem, providing an extension to current schemes that allows for a reduction in cipher expansion; its performance is showcased in a biometric application tackling private face recognition with encrypted faces, templates and parameters, in which a novel model for Gabor coefficients is presented and evaluated in terms of goodness of fit and used for applying an optimal coefficient quantization that leads to a huge plaintext reduction without hindering the recognition performance (J1,C1,C6,C7,P1).

1.3. Outline

This thesis summarizes the research work done in SPED and structures it in a comprehensive way, following a bottom-up approach, going from very specific low-level primitives (point inclusion, linear equations solving, automata execution), to a general framework covering a whole field in signal processing (secure adaptive filtering), and a high level architecture for generic applications in a non-trusted environment (Secure Cloud) together with other application-oriented solutions covering secure watermarking or medical scenarios. Finally, the last chapters also point to other directions for privacy protection besides SPED (like the use of DRM) and give a glimpse on how future fully homomorphic cryptosystems can revolutionize the unattended and interactive private encrypted processing, exemplifying it in a biometric application.

The remaining chapters of the thesis are structured as follows:

- Chapter 2 shows several generic low level secure primitives applicable in a privacy-aware scenario to problems like linear algebra (iteratively solving systems of linear equations) and determining point inclusion in a specified region. Some of the numerous applications of these primitives are also shown in this chapter.

- Chapter 3 presents the problem of Adaptive Filtering with privacy constraints and compares several protocols combining the available privacy techniques in terms of the tradeoff computational load-bandwidth-precision, and proposing solutions to the cipher blowup problem.

- Chapter 4 presents several application scenarios for which specific privacy preserving solutions are proposed, dealing with zero-knowledge watermark detection, Cloud Computing privacy-preserving architectures, an medical applications addressing the concept of Private Medical Clouds or secure systems for DNA queries.

- Chapter 5 introduces other privacy-aware scenario dealing with multimedia signals: videosurveillance. This chapter takes a different approach as the previous ones and presents a whole privacy framework based on the use of DRM and standards-compliant multimedia coding in order to conceal the sensitive parts of the involved video streams and empower the users with the control of their private data.

- Chapter 6 presents a double contribution targeted towards private biometric authentication; on the one hand, it provides a novel model for Gabor coefficients extracted from face images that allows for better feature compression with almost no impact in face recognition performance, and pairs it with a fully private authentication system based on a novel extension of a fully homomorphic cryptosystem by Gentry, opening the door to a fully noninteractive private system for outsourced processing of sensitive data.

- Chapter 7 enumerates and elaborates the conclusions that can be drawn from the concepts introduced in this thesis, and also points out the future research lines that they open.

Chapter 2

Generic Secure Primitives

This chapter deals with three important basic primitives in signal processing: the *multidimensional point inclusion problem*, the solution of *systems of linear equations*–introducing the cipher blow-up problem for iterative algorithms, that is fully developed in the next chapter–, and the secure execution of finite automata. Finally, some basic elements, mainly comprising zero-knowledge proofs and secure subprotocols, that will be used as building blocks in the following chapters are presented also in the Appendices.

Many signal processing applications reduce to a *multidimensional point inclusion problem* where two participants decide whether a point known to the first lies inside a region specified by the second. In a secure solution, neither party gains knowledge about the other's input. For instance, in biometric authentication the client can prove his identity without disclosing his biometric. In this chapter, we present a new primitive for securely solving the multidimensional point inclusion problem. Using this primitive, we first propose an efficient and provably secure protocol that solves the problem for an N-dimensional convex region bounded with hyperplanes. We subsequently extend the protocol to inclusion in multiple hyperellipsoidal regions. Considering possible reduction strategies such as input packing, we analyze the complexity of both protocols.

In the second part of the chapter, we also propose novel privacy-preserving protocols for the solution of *Linear Systems of Equations*, that improve on previous contributions in terms of security; we present secure implementations of iterative algorithms, pointing out the difficulties that arise when dealing with iterative operations on encrypted data, and proposing possible solutions to these shortcomings.

Finally, the third part of the chapter presents an error-resilient privacy-preserving automata execution protocol, with many applications, like string searching or private DNA queries (see Section 4.4).

This protocol checks if a short template, known to one party, is present inside a sequence owned by another party, accounting for possible errors and without disclosing to each party the other party's input. Each query is formulated as a regular expression over a finite alphabet and implemented as an automaton. As the main technical contribution, we provide a protocol that allows to execute any finite state machine in an oblivious manner, requiring a communication complexity which is linear both in the number of states and the length of the input string.

The work shown in this chapter has been partially presented at ACM MMSEC'07 [230], CCS 2007 [221], and SPEED Workshop 2009 [220], and some of the technical developments have been filed as patent applications (Patent pending, Application No. 61/240177, 61/240179, 61/240181, EPO EP10175467, and PCT PCT/IB2008/051771).

2.1. Introduction

In the *privacy preserving computation* framework that this thesis covers, several proposals have been recently issued to implement primitives like secure access to encrypted databases [51, 209], transcoding of an encrypted signal without prior decryption [135], or basic problems in computational geometry, such as computing scalar products [106] or solving the point inclusion problem for the 2-dimensional case [32].

The point inclusion problem refers to deciding whether a point lies in a certain spatial region. It is related to point location in computational geometry, which has been investigated for two-dimensional spaces for more than twenty years [83] optimizing the algorithms for achieving subpolynomial search time and storage. For more than two dimensions, the point location problem is still open, except for the case of arrangements of hyperplanes [58], convex subdivisions [188], special convex polytopes [153], and other subdivisions that allow efficient point location.

Point inclusion is an underlying problem in many common signal processing applications that must be run in untrusted environments; however, it rarely deals with 2-dimensional signals, but with multidimensional ones. For instance, in the case of biometric authentication, the biometric data (a feature vector embodying a point in a multidimensional space) that is presented by an individual must be matched with some template (represented by a region of acceptance in the space) that is held by a server, but both parties do not want to disclose their respective inputs to the other party.

Additionally, the efficient protocols presented up to now in the field of signal processing in the encrypted domain have been focused in linear operations, like scalar products, and non-iterative algorithms. Nevertheless, there are many basic algorithms needed for most signal processing applications that are iterative and involve not only scalar products with known values, but also products between two a priori unknown sequences. The lack of these algorithms would suppose missing a powerful and irreplaceable tool that enables almost any signal processing application.

In this chapter, we first present an efficient and provably secure two-party protocol for solving the point inclusion problem in a convex region bounded by hyperplanes in N-dimensional space. In our construction, we use the public key encryption scheme of Paillier [177] for concealing the input coordinates of the point, compute the relative position of this point and each of the hyperplanes under encryption, use the `BITREP` gate [199] to extract the result for each hyperplane, and merge them again into the binary decision. Our primitive can also be extended to multi-party scenarios, as well as to non-convex regions, as every non-convex region can be expressed as the disjoint union of convex regions. We perform a full complexity analysis for the whole protocol, including the `BITREP` subblock.

For dealing with unconnected regions, we also present an extension of our construction to regions specified as the union of several hyperellipsoids, with the same privacy properties as the former one. As a special case, our protocols also yield a solution to the 2-dimensional problem, resulting in a much more efficient and secure solution than the one proposed in [32].

We also cope with the problem of interactive algorithms in signal processing, and present efficient and provably secure two-party protocols for solving linear systems of equations and inverting matrices, useful for many applications (e.g. least squares minimization), implementing also iterative algorithms, and calculating the needed cipher size to accommodate a given number of iterations. We also perform a full complexity analysis of the presented protocols.

Finally, we present an efficient (amortized linear time) protocol for the oblivious execution of a finite state automaton; Section 4.4 will show how it can be used to solve the problem of oblivious approximate string matching and searching, allowing a maximal number of symbol errors, insertions and deletions. We extend it also to automata with non-binary output (Mealy and Moore machines), used in applications such as text parsing, computational linguistics and speech recognition.

The rest of the chapter is organized as follows: Section 2.2 surveys related previous work, and describes the building blocks and required concepts for the implementation of our protocols. In Section 2.3, we present a protocol for secure point inclusion in an N-dimensional polytope, and extend it to cope with non-connected hyperellipsoidal regions. A complete complexity analysis is undertaken in Section 2.3.4. Section 2.4 sketches our protocols for secure linear algebra and gives their measures of complexity. Section 2.4.4 evaluates the given complexity measures for a specific construction, gives some examples of use and, for the iterative protocols, plots bounds to representable numbers as a function of the performed iterations, focusing on the trade-off among the three considered parameters: complexity, representability and number of iterations. Section 2.5 presents a privacy preserving protocol for the execution of finite automata, whose complexity is evaluated in Section 2.5.2; the protocol is also extended to transducers keeping the same order of complexity, and its security is analyzed in Section 2.5.4. Finally, Section 2.6 summarizes the obtained results and sketches the future lines.

2.2. Preliminaries

In this section, we briefly survey the related work and explain the building blocks needed for implementing our protocols.

2.2.1. Related Work

To the best of our knowledge, the only proposal for solving the problem of point inclusion through secure two-party computation was presented by Atallah and Du [32] for a two-dimensional problem where the region is a convex polygon. The authors develop two primitives, namely a protocol for privately computing the scalar product of two values, and a vector dominance protocol that privately tests whether all components of one vector are greater than the components of another vector. The latter protocol is based on several parallel executions of Yao's millionaires' protocol. Finally, they require a method of equality testing. The protocol by Atallah and Du has been recently used in [197] for privately determining the positioning on the sensing area of a pervasive sensor network.

Atallah and Du's solution has several drawbacks. The first problem is related to their protocol for privately computing the scalar product. As pointed out in [243] and [106], it does not preserve privacy. With a simple attack one of the parties can, with a probability close to 1, retrieve the private input of the other party after a single execution of the protocol. The second drawback is the inefficiency of their vector dominance protocol, as it involves several executions of Yao millionaires protocol. Finally, the protocol they propose for equality testing only works when using a commutative deterministic encryption, which cannot achieve semantic security.

Regarding the problem of privately solving linear systems of equations, there was a previous approach by Du and Atallah [86]. In that work, the authors presented the problem of solving a linear system with a matrix and an independent vector partitioned between two parties. They provided a solution based on secret sharing, but the privacy that it achieves is not total; as later works have shown (cf. Wright and Yang [243]), their protocol for secure multiplication leaks information about the multiplied matrices, and it also relies on a security parameter that largely increases the needed communication in order to achieve a determined level of concealment on the multiplied values. It is also worth mentioning that Cramer and Damgård proposed in [70] a solution to distributed linear algebra problems, coping with finite fields; on the contrary, this chapter gives solutions to problems posed in $\mathbb{R}^n$.

The protocols presented in this chapter improve on previous work in terms of achieved privacy, practically limiting the leak of information to the inherent leak produced by the disclosure of the solution of the point inclusion or of a SLE.

Regarding prior work in the field of Signal Processing in the Encrypted Domain, we are not aware of any previous solution for securely executing iterative algorithms, nor any study performed on the impact that an iterative implementation has on the range of representable numbers. Hence, this chapter presents the first solution for privacy preserving iterative algorithms, that is further developed and studied in the next chapter, within the framework of secure adaptive filtering.

As for the problem of oblivious automata execution, to the best of our knowledge there is no prior work dealing with it; the closest problems are *approximate string matching and searching*; for the latter, a short sequence $\boldsymbol{x}$ (the pattern) is searched in a longer sequence $\boldsymbol{y}$, while tolerating Edit errors (insertions, deletions and substitutions of symbols): if an approximate match (using the Edit distance as metric) is found between the pattern $\boldsymbol{x}$ and some substring of $\boldsymbol{y}$, the search will report a positive answer. The problem of approximate string matching can be seen as a special case of searching when the length of the pattern and the length of the sequence $\boldsymbol{y}$ are approximately equal (up to insertions and deletions of a certain number of symbols). These two problems and the application of automata for solving them will be described in more detail in Chapter 4.

Approximate string searching is one exemplary instance where generic SMC solutions yield to particularly inefficient protocols; this is mainly due to the need for error-resilience in the search process. Furthermore, approximate searching through automata execution is a highly asymmetric problem, in the sense that only one party (the server) knows the function that is evaluated (the automaton), whereas the other party (the client) holds the corresponding input. From a higher level perspective, both parties agree on some specific *functionality* (i.e., a class of functions), while the specific function to be evaluated is considered a private input of one party.

To the best of our knowledge, only the work in [33, 34] gives an efficient solution for a problem akin to *privacy preserving approximate string searching.*

In that work, the authors present a protocol for *privacy preserving Edit distance evaluation*. The calculation of the Edit distance is performed through a dynamic programming algorithm [42] that achieves linear time complexity in the product of the lengths of the compared sequences. The authors of [33, 34] implement an oblivious version of the dynamic programming algorithm that achieves the same order of complexity. If a threshold in the number of admissible errors is established, their protocol can be regarded as a solution to a particular instance of the problem of *approximate string matching*. As a central tool in the construction, the authors run an instance of Yao's Millionaire's problem at each step of the dynamic programming algorithm, making the solution inefficient in practice.

The protocol in [33, 34] can be extended to solve the *approximate string searching* problem as well. However, the number of comparisons involved in the dynamic programming algorithm grows with the product of the length of the strings. The solution proposed in this chapter completely avoids comparisons of encrypted values, thus overcoming this scalability problem. Furthermore, our solution is more general, as it is not limited to approximate matching or searching, but can be applied to any finite automata, and, therefore, any regular expression matching problem in sequences formed by symbols of a finite alphabet.

For solving the posed problem, we use secret sharing [61], homomorphic encryption and 1-out of-m oblivious transfer OT_1^m [168], and develop a specific protocol for the secure evaluation of a finite automaton. Finally, it must be noted that trying to address the private execution of finite automata through the adaptation of generic constructions for secure function evaluation (e.g., *Generalized Indirect Indexing* [167, 166] or *Mix and Match* [130]) poses several problems, as these primitives cannot efficiently index two-dimensional matrices. For a straightforward application of these techniques, the state transition matrix of the automaton must be flattened, and in each processing step an amount of data equivalent to the whole matrix must be transferred between both parties. This results in a communication complexity of $\mathcal{O}(N \cdot |Q| \cdot |\Sigma|)$ (cf. Section 2.2.2.4). In contrast, the solution of Section 2.5 achieves a communication complexity that is both linear in $|Q|$ and $|\Sigma|$.

2.2.2. Building Blocks

In this section, we introduce some of the concepts employed in the rest of the chapter, namely *threshold homomorphic encryption*, *secret sharing*, the BITREP protocol and the formal definition of *finite automata*.

2.2.2.1. Threshold Homomorphic Encryption

In this chapter we do not restrict the used cryptosystem for the presented protocols, as far as it presents an additive homomorphism. For the sake of clarification, and for performing the numerical complexity calculations, we will use either the Paillier cryptosystem [177] or its Damgård-Jurik extension [79], both in its threshold and non-threshold form; hence, $E_P(x)$ and $D_P(x)$ (resp. $E_{DJ}(x)$ and $D_{DJ}(x)$) will denote the encryption and decryption operations on x.

A k out of M threshold public key encryption system [84] is a cryptosystem where the private key is distributed among M parties, and at least k of them are needed for decryption. Damgård and Jurik's cryptosystem presents an additive homomorphism that allows computing the addition of two encrypted numbers and the product of an encrypted number and a public integer:

$$\begin{aligned} [\![x+y]\!] =& E_{DJ}[x+y] = E_{DJ}[x] \cdot E_{DJ}[y] \mod n^{s+1}, \\ [\![x \cdot k]\!] =& E_{DJ}[x \cdot k] = E_{DJ}[x]^k \mod n^{s+1}. \end{aligned}$$

As multiplications cannot be performed homomorphically, we will use in our constructions the two-party version of one existing subprotocol for securely performing multiplication; it is sketched in Appendix 2.C.

The message space is $\mathbb{Z}_{n^s}$, where n is the product of two safe primes p, q, and the parameter s is fixed. Unless the contrary is explicitly said, we will use $s = 1$ (regular Paillier) throughout the chapter.

The encryption of a message x is obtained by picking a random $r \in \mathbb{Z}^*_{n^{s+1}}$ and computing the ciphertext $E_{DJ}[x]$ as

$$E_{DJ}[x] = g^x r^{n^s} \mod n^{s+1}.$$

For the threshold decryption of $c = E_{DJ}[x]$, every party calculates a decryption share with his share of the secret key. These decryption shares are distributed among all the parties, and combined to obtain the wanted decryption. In case of malicious parties, they must also generate a zero-knowledge proof [205] for the correctness of the decryption share. For further details, we refer the reader to [79].

We must again draw attention to the fact that currently there is no practical fully homomorphic cryptosystem, that is, there is no secure cryptosystem that allows for the homomorphic computation of additions and products without restrictions. There have been recent contributions by Gentry [104], that present a cryptosystem based on ideal lattices with bootstrappable decryption, and it is shown that it achieves a full homomorphism. Nevertheless, the authors argue that making the scheme practical remains an open problem. Thus, until

Chapter 6, we will adhere to using an additively homomorphic cryptosystem and briefly comment the advantages that an efficient and practical fully homomorphic cryptosystem would provide.

2.2.2.2. Secret Sharing

Secret sharing is a technique introduced by Adi Shamir [202], by which a given value (the secret) is divided among several parties, such that the cooperation among a number of these parties is needed in order to recover the secret. None of the parties alone can have access to the secret.

Shamir's scheme is based on polynomials, and the need of k points in order to completely determine a degree $(k-1)$ polynomial. Secret sharing is a widely used primitive in cryptographic protocols. In this work we focus on two-party protocols; thus, we are only interested in the two-party version of the secret sharing scheme, that is based on linear functions and, consequently, it naturally supports the computation of sums and products directly on the shares: let $\mathbb{Z}_n$ be the domain of the secrets. Then, a share of a secret x is defined as two values x_A and x_B, owned by their respective parties, such that $x_A + x_B \equiv x \mod n$. Hereinafter, randomizing an encrypted value x will mean obtaining one share and providing the encryption of the other (through homomorphic addition).

2.2.2.3. `BITREP` Protocol

While homomorphic computation and secret sharing are very efficient for implementing arithmetic operations, circuit evaluation is still more efficient when dealing with binary tests [77]. Thus, there exist efficient protocols for binary comparison [77, 173] or Prefix-OR [77]. Traditionally, the search for efficient solutions has led to proposals for changing between integer and binary representation in order to efficiently implement both arithmetic and binary operations; e.g., there are solutions like BITREP protocol [199], that converts Paillier encrypted integers to Paillier encryptions of their corresponding bit representation.

The BITREP protocol was presented by Schoenmakers and Tuyls [199], as a means for securely converting a Paillier encrypted l-bit number into l Paillier encryptions of its individual bits. The protocol is based on random bit gates and addition circuits, and comes in two variants:

- The first one, called `BITREP` gate, performs bit extraction for any number in $\mathbb{Z}_n$, using an information-theoretic blinding of the given number through additive randomness jointly generated among all the parties, and two addition circuits.

- The second one, called `LSBs` gate, is a *light* version of the previous one, that substitutes the information-theoretic blinding by a statistically indistinguishable one, with a security parameter k, that imposes a bound on the bit-size of the input number. To use the protocol, each input must be at least k bits smaller than the modulus. In this way, the most costly circuit evaluation is eliminated, and the efficiency is greatly improved.

In the following, the bit-extraction gate will be referred to as `BITREP`, independently of its implementation. The distinction between `BITREP` and `LSBs` will only be made explicit in the complexity analysis of Appendix 2.A.

2.2.2.4. Finite Automata

A deterministic finite automaton [123] (or finite state machine, FSM) is denoted by a 5-tuple $M = (Q, \Sigma, \boldsymbol{\Delta}, q_0, F)$, where Q is a finite set of states, Σ is a finite input alphabet, $q_0 \in Q$ is the initial state, $F \subseteq Q$ is the set of final states, and $\boldsymbol{\Delta}$ denotes the transition function. Without loss of generality, we restrict ourselves to 'complete' finite automata, where it is possible to make a transition at each state with every input symbol (each FSM can be transformed into an equivalent complete automaton by adding a sink state). We represent the states as integers in $\mathbb{Z}_{|Q|}$, the inputs as integers in $\mathbb{Z}_{|\Sigma|}$, and the transition function as a matrix $\boldsymbol{\Delta} \in \mathcal{M}_{|\Sigma|\times|Q|}(\mathbb{Z}_{|Q|})$, such that $\Delta(i, j)$ represents the next state when the FSM sees an input $i \in \Sigma$ and is in current state $j \in Q$. A string $\boldsymbol{x} = x_0 x_1 \dots x_{N-1} \in \Sigma^N$ is said to be accepted by the finite automaton M if the state

$$q_N = \Delta(\cdots \Delta(\Delta(q_0, x_0), x_1) \cdots, x_{N-1})$$

is a final state $q_N \in F$.

The language accepted by a finite automaton M is the subset of all strings from Σ^* it accepts. It is known that the sets accepted by FSMs and regular sets coincide. Thus, for every regular expression there is a finite automaton that accepts only words that match that expression, and vice-versa.

Finite automata can only express decision problems; thus, they are also called acceptors. The theory of automata has been extended to finite state machines that are capable of producing a string over a finite alphabet Π as output. Automata with non-binary output are called transducers, and they can be classified into two groups:

- *Moore machines:* At each transition, the automaton produces one symbol as output, being this a function of the current state of the machine. Formally, a Moore machine is a 6-tuple $(Q, \Sigma, \Pi, \boldsymbol{\Delta}, \boldsymbol{\lambda}, q_0)$, where $Q, \Sigma, \boldsymbol{\Delta}, q_0$ have the same meaning as for FSMs. Π denotes the output alphabet, and $\boldsymbol{\lambda} \in \Pi^{|Q|}$ is a vector whose components $\lambda(q)$ encode the output symbol of the machine at a given state q.

- *Mealy machines:* At each transition, the automaton produces one output symbol, which can depend on the transition taken. Formally, a Mealy machine is a 6-tuple $(Q, \Sigma, \Pi, \boldsymbol{\Delta}, \boldsymbol{\Lambda}, q_0)$, where $Q, \Sigma, \boldsymbol{\Delta}, q_0$ have the same meaning as for FSMs. Π denotes the output alphabet, and $\boldsymbol{\Lambda} \in \mathcal{M}_{|\Sigma|\times|Q|}(\Pi)$ is a matrix whose components $\lambda(a, q)$ encode the output of the machine for a given state q and input symbol a.

In general, Moore machines are as expressive as Mealy machines; however, a Mealy machine may need a smaller number of states than its equivalent Moore machine.

2.3. Point Inclusion Protocol

This section is devoted to the development of secure protocols for the point inclusion problem in two different cases. Firstly, we address the case of a convex region whose boundary is given by hyperplanes. We then address non-connected regions given by several hyperellipsoids. As a final remark, we point out how the problem of Vector Dominance, that can be seen as a subproblem of point inclusion in our construction, can be solved by employing the same technique as for the two main protocols.

2.3.1. Convex Region bounded by Hyperplanes

We present a two-party protocol that implements a privacy preserving solution to the point inclusion problem. The employed method for determining the point inclusion assumes convex polytopes, but it can be easily extended to non-convex polytopes just by taking into account that any polytope can be expressed as a disjoint union of convex polytopes. As we are working in an N-dimensional space, vectors will have length N.

The problem can be stated as follows: Let $\mathcal{S}$ be a convex polytope in an N-dimensional space that is delimited by K hyperplanes given in Hessian normal form $\left\{\boldsymbol{n}_m^T\boldsymbol{x} = -\eta\right\}_{m=0}^{K-1}$, with their normal vectors heading toward the inner region of $\mathcal{S}$. Each hyperplane defines a hemispace $\mathcal{S}_m = \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{n}_m^T\boldsymbol{x} + \eta < 0\}$, such that $\mathcal{S} = \cap_m \mathcal{S}_m$. The scenario comprises two parties: Alice, who has a point $\boldsymbol{a} = \{a_i\}_{i=0}^{N-1}$, and Bob, who has the description of the region $\mathcal{S}$. Alice wants to check if $\boldsymbol{a} \in \mathcal{S}$, without disclosing any knowledge of the coordinates of $\boldsymbol{a}$ to Bob, and without getting any knowledge of the description of $\mathcal{S}$.

The clear-text algorithm for solving the posed problem consists in checking the position of the point with respect to each of the hyperplanes: if $\boldsymbol{a} \in \mathcal{S}_m, \forall m = 1, \ldots, K$, which can be checked by computing the inner product between the point and the normal vector of each plane and adding the corresponding offsets, then the point lies inside $\mathcal{S}$. The time efficiency of this algorithm can be optimized: once a hyperplane that leaves the point outside of the bounded region is found, the algorithm stops without computing the remaining scalar products (in such a case, this algorithm is the most efficient one also for the two-dimensional point inclusion problem in convex polygons of small area). However, a direct implementation of this stop-test is not possible in our scenario, as it would leak too much information about the position of the point w.r.t. individual hyperplanes.

We assume semi-honest parties in this section, i.e. we will consider that the parties honestly execute the protocol even though they may record all interchanged messages and try to deduce information about the secret input of other parties. We will indicate also the modifications needed for considering malicious parties in Appendix 2.B. Assuming that Alice's point will be at most at a distance of $\tau = 2^l$ units from each of Bob's hyperplanes, Bob can pack his planes in only one vector $\boldsymbol{n}_{\text{packed}}$ and one offset η_{packed}, expressed as

$$\boldsymbol{n}_{\text{packed}} = -\sum_{m=0}^{K-1} \boldsymbol{n}_m \cdot 2^{m(l+1)} \tag{2.1}$$

and

$$\eta_{\text{packed}} = \sum_{m=0}^{K-1} \left(-\eta_m + 2^l\right) \cdot 2^{m(l+1)},$$

where the displacement by 2^l represents a shift to work only with positive numbers. Note that this packing codes the sign of the l-th computed distance in the $(l+1)$-th bit, being 0 in case the given coordinates are inside the considered hemispace $\mathcal{S}_m$, and 1 otherwise.

The packing can be done before the execution of the protocol. Thus, if each signed distance requires at most $l+1$ bits, the maximum number of hyperplanes that can be packed together is

$$K \leq \frac{\lfloor \log_2 n \rfloor}{l+1}. \tag{2.2}$$

Intuitively, the protocol computes the scalar product of Alice's coordinates and each of the hyperplane normal vectors and adds the corresponding offset, storing the sign of the distance between the point and the m-th hyperplane at the $(m(l+1)+l+1)$-th bit of the resulting encrypted number. Then, all these bits are extracted with the `BITREP` gate and combined to form the answer.

More formally, Alice and Bob engage in the following protocol:

1. Alice encrypts the coordinates of her point $\boldsymbol{a}$ using a threshold Paillier cryptosystem to obtain a vector
$$\{E_P[a_i]\}_{i=0}^{N-1} = \{g^{a_i} r_i^n \mod n^2\}_{i=0}^{N-1},$$
and sends it to Bob.

2. Bob calculates the signed distance from Alice's point to each of the K hyperplanes, performing the scalar product between the point $\boldsymbol{a}$ and each of the hyperplane normal vectors $\boldsymbol{n}$ and adding the respective offset to obtain a *packed* distance
$$d_{\text{packed}} = \boldsymbol{n}_{\text{packed}}^T \cdot \boldsymbol{a} + \eta_{\text{packed}}.$$
This operation can be done under encryption by employing the homomorphic properties of the cryptosystem, applying a rerandomization given by r':
$$\begin{aligned} E_P[d_{\text{packed}}] = (r')^n \cdot E_P[\eta_{\text{packed}}]\cdot \\ \prod_{i=0}^{N-1} (E_P[a_i])^{n_{packed,i}} \mod n^2. \end{aligned}$$
The signs for the distances—considering zero as negative, as the polytope is an open set—to each of the planes are coded in the bits of d_{packed} as indicated before
$$d_{\text{packed}}[m(l+1)+l].$$

3. Both parties run the `BITREP` protocol presented in [199] on $E_P[d_{\text{packed}}]$ for getting the bit representation of d_{packed}, and take the encryptions corresponding to bits numbered $\{m(l+1)+l\}_{m=0}^{K-1}$, which we will denote as
$$\{E_P[b_m]\}_{m=0}^{K-1} = \{E_P[\boldsymbol{a} \notin \mathcal{S}_m]\}_{m=0}^{K-1}.$$

4. In order to determine the number of hyperplanes that leave Alice's point outside the polytope one can add the values $[\boldsymbol{a} \notin \mathcal{S}_m]$ to obtain
$$D = \sum_{m=0}^{K-1} b_m.$$

The point lies inside the polytope $\mathcal{S}$ if and only if $D = 0$, as

$$\boldsymbol{a} \in \mathcal{S} \Leftrightarrow \bigwedge_{m=0}^{K-1} (b_m) \Leftrightarrow D = 0.$$

The encryption of D can be calculated independently by Alice and Bob using the homomorphic properties of E_P:

$$E_P[D] = \prod_{m=0}^{K-1} E_P[b_m].$$

5. Alice and Bob jointly multiply the encrypted number $E_P[D]$ by a random number $r = r_A \cdot r_B$, where r_A and r_B are random choices of Alice and Bob respectively.

6. The result of the previous blinding $E_P[D \cdot r \mod n]$ is jointly decrypted by both parties to obtain a zero value in case $\boldsymbol{a} \in \mathcal{S}$, or a random (uniformly distributed) value in case this condition is not fulfilled.

The protocol can be proven secure, as all private inputs are encrypted and the semantic security of the cryptosystem guarantees no information leakage about these values. Furthermore, [199] proves the security of `BITREP`. Given this property and the statistical indistinguishability of the encrypted values, the view for each party can be straightforwardly simulated without interacting with the other party; this simulation has indistinguishable distribution from real interactions, thus showing the security of the whole protocol.

The packing (2.1) bounds the number of possible hyperplanes that can be used to describe the geometrical region by $K \leq \frac{\lfloor \log_2 n \rfloor}{l+1}$. In a typical scenario, with modulus size of 1024 bits and 32 bits for the magnitude of the distances ($d \in [-2^{32}, 2^{32} - 1]$), 31 hyperplanes can be used. If the region is defined by more hyperplanes, there are two possible solutions:

- Use Damgård-Jurik encryptions [79], with $\mathbb{Z}_{n^s}$ as the clear-text domain instead of Paillier encryptions. This would allow $\frac{\lfloor s \log_2 n \rfloor}{l+1}$ hyperplanes, at the cost of having a longer and more complex `BITREP` protocol.
- Perform steps (1)-(3) of the protocol in parallel, each time packing at most K hyperplanes together and combining all extracted bits in step (4). This solution is more efficient than the previous one, as will be seen in Section 2.3.4.

Though the previous exposition has been done for a two-party protocol, it can be generalized to allow multi-party sharing of the boundaries of the region $\mathcal{S}$ among several servers, or of the coordinates $\boldsymbol{a}$ among several clients, just by extending the number of parties required for decryption.

2.3.2. Extension to Hyperellipsoidal Regions

In this section, we show how the protocol of Section 2.3.1 can be extended to regions specified by hyperspheres or hyperellipsoids. In the case of hyperspheres, the server possesses K centroids in an N-dimensional space (with their respective coordinates $\{\boldsymbol{b}_m\}_{m=0}^{K-1}$, and the squared radius of the closed balls centered at them $\{\rho_m^2\}_{m=0}^{K-1}$). The client wants to know if her point $\boldsymbol{a}$ is inside one of those balls.

The modified protocol can be applied to group authentication, in which there are several acceptance regions, and the features of the client must be tested for inclusion in all these regions to find if it lies inside one of them.

Hyperelliptical regions can be coped with by applying a strain γ_i to each of the dimensions in order to transform the hyperspheres in hyperellipsoids, such that the i-th dimension semiaxis of the m-th hyperellipsoid has a length $l_{m,i} = \rho_m/\gamma_{m,i}$.

The point $\boldsymbol{a}$ lies inside the m-th hyperellipsoid if the following condition is satisfied:

$$\begin{gathered}
(\mathrm{Diag}(\boldsymbol{\gamma}_m) \cdot (\boldsymbol{a} - \boldsymbol{b}_m))^T \cdot \mathrm{Diag}(\boldsymbol{\gamma}_m) \cdot (\boldsymbol{a} - \boldsymbol{b}_m) \leq \rho_m^2 \Leftrightarrow \\
\sum_{i=0}^{N-1} \gamma_{m,i}^2 (a_i - b_{m,i})^2 \leq \rho_m^2 \Leftrightarrow \qquad (2.3) \\
-\underbrace{\sum_{i=0}^{N-1} \gamma_{m,i}^2 a_i^2}_{\boldsymbol{\gamma^2}_m^T \cdot \boldsymbol{a^2}} + \underbrace{\sum_{i=0}^{N-1} 2 b_{m,i} \gamma_{m,i}^2 a_i}_{\boldsymbol{2b'}_m^T \cdot \boldsymbol{a}} + \underbrace{\rho_m^2 - \sum_{i=0}^{N-1} \left(\gamma_{m,i}^2 b_{m,i}^2\right)}_{\rho'^2_m} \geq 0.
\end{gathered}$$

The strains can be generalized to any linear transformation, just by substituting the strain diagonal matrix by a full matrix; this would correspond to the case in which it is necessary to eliminate linear dependencies among the features, and this operation has to be performed in the encrypted domain.

To solve the hyperelliptical point inclusion problem, one can use the following modified version of the protocol of Section 2.3.1:

1. The server packs the coordinates of the centroids, the radii (including the

squared values of the centroids), and the strain vector

$$\begin{aligned}
\boldsymbol{2b'}_{\text{packed}} &= 2\sum_{m=0}^{K-1} \operatorname{Diag}(\boldsymbol{\gamma}_m) \cdot \boldsymbol{b}_m \cdot 2^{m(l+1)}, \\
\rho'^2_{\text{packed}} &= \sum_{m=0}^{K-1} \left(\rho_m^2 - \boldsymbol{\gamma}_m^T \cdot \boldsymbol{b}_m^{\boldsymbol{2}} + 2^l\right) \cdot 2^{m(l+1)}, \\
\boldsymbol{\gamma}^{\boldsymbol{2}}_{\text{packed}} &= -\sum_{m=0}^{K-1} \boldsymbol{\gamma}_m \cdot 2^{m(l+1)}.
\end{aligned}$$

2. The client encrypts her coordinates and the squared value of her coordinates with the threshold Pallier cryptosystem, and sends both vectors $E_P[\boldsymbol{a}]$ and $E_P[\boldsymbol{a}^{\boldsymbol{2}}]$ to the server.

3. The server computes the signed distance of the client's coordinates to each of the hyperellipsoidal surfaces according to (2.3), where the points on the hyperellipsoid are considered inside the accepted region, as it is a closed set. The distances are packed into

$$d_{\text{packed}} = \left(\boldsymbol{\gamma}^{\boldsymbol{2}}_{\text{packed}}\right)^T \cdot \boldsymbol{a}^{\boldsymbol{2}} + \boldsymbol{2b'}^T_{\text{packed}} \cdot \boldsymbol{a} + \rho'^2_{\text{packed}}.$$

   This operation can be done under encryption by

$$\begin{aligned}
E_P[d_{\text{packed}}] = x_m'^n \cdot &\left(\prod_{i=0}^{N-1} \left(E_P[a_i^2]\right)^{\gamma^2_{\text{packed},i}}\right) \cdot \\
&\left(\prod_{i=0}^{N-1} \left(E_P[a_i]\right)^{2b'_{\text{packed},i}}\right) \cdot \\
&E_P[\rho'^2_{\text{packed}}] \mod n^2.
\end{aligned}$$

After these computations, the rest of the protocol comprises steps 3 to 5 of the one in Section 2.3.1, with a different interpretation of the obtained result: $D = 0$ means that the client's coordinates are not within one of the acceptance regions, whereas $D \neq 0$ means that the coordinates are inside one (or more) of the acceptance regions.

2.3.3. Vector Dominance Problem

The *two-party secure vector dominance problem*, also referred to as *multidimensional Yao's millionaires problem*, was first addressed in [32], and consists of the private comparison of two vectors $\boldsymbol{c}$ and $\boldsymbol{b}$, each one owned by one of the parties participating in the protocol, in such a way that the output of the protocol is the binary value of

$$\boldsymbol{c} \succ \boldsymbol{b} \equiv \bigwedge_i (c_i > b_i) \, .$$

During the execution of the protocol, no other information must be disclosed, such as the values of the vector components or the indexes for which $c_i \not> b_i$. Several protocols have been proposed to solve this problem. Atallah and Du [32] use parallel executions of Yao's Millionaires' protocol for each component, with a blinding of each individual comparison. Sang and Shen [197] use prefix-encoding and a privacy preserving prefix test. At last, the solution proposed by Ibrahim [127], is based on secret sharing, in particular, a subroutine to transform multiplicative shares in additive ones, that requires several invocations of 1 out of 2 Oblivious Transfer; if domination exists, this protocol discloses the sum of the elements of both vectors.

The protocol presented in Section 2.3.1 gives an implicit solution to the Vector Dominance Problem. In fact, the result of the protocol is the answer to

$$\boldsymbol{N}^T \cdot \boldsymbol{a} \succ -\boldsymbol{\eta},$$

where $\boldsymbol{N}$ is the matrix that has as columns the normal vectors of Bob's hyperplanes, and $\boldsymbol{a}$ and $\boldsymbol{\eta}$ have the same meaning as in the previous section. This means that, given two bounded vectors $\boldsymbol{c}$ and $\boldsymbol{d}$ (with components in the range $[-\tau, \tau]$), each owned by a party, it is possible to privately determine if one of them dominates the other if each party packs its own vector (c_{packed} and b_{packed}) in the same way as Bob does with $\boldsymbol{n}$ in (2.1), one party sends his packed and encrypted vector to the other, who subtracts his own vector to this, and both run steps 3 to 6 of the presented protocol in order to obtain the solution to the dominance problem.

2.3.4. Complexity Evaluation

We analyze both the communication and computation complexity of the protocols proposed in Sections 2.3.1 and 2.3.2. Let N be the dimensionality of the problem, K the total number of hyperplanes, $m = \lceil \log_2 n \rceil$ the bit size of the modulus n, s the factor for Damgård-Jurik encryptions, l the bit-size bound for distances ($|d_m| \leq 2^l$), and k the security parameter for the employed `BITREP` gate. Finally, the communication (cm), computation (cp) and precomputation (pcp) complexity for `BITREP`, as well as its round complexity (Rounds) are the ones computed in Appendix 2.A.

As mentioned at the end of Section 2.3.1, we consider the case in which only h of the K hyperplanes are packed together and the steps 1 to 3 of the protocol are executed in parallel. This number h is bounded by

$$h \leq \left\lfloor \frac{m \cdot s - (k+1)}{l+1} \right\rfloor .$$

Thus, the number of BITREP gates that both parties must run (in parallel) in step 3 of the protocol is $\lceil \frac{K}{h} \rceil$, all of them executed on numbers of size $n_1 = h \cdot (l+1)$ bits, but the last gate, that will be executed on a number of size $n_2 = K - h \cdot \left(\lceil \frac{K}{h} \rceil - 1 \right)$.

We denote by $|E_P|$ the size of the threshold Paillier encryption. The communication complexity (cm) for the whole protocol is given by

$$\begin{aligned} \mathrm{Cpx}_{cm} = & \left(\left\lceil \frac{K}{h} \right\rceil - 1 \right) \left(|E_P| + \mathrm{Cpx}_{cm,\mathtt{BITREP},n_1} \right) \\ & + (N+5)|E_P| + \mathrm{Cpx}_{cm,\mathtt{BITREP},n_2} \end{aligned}$$

The round complexity for the whole protocol can also be calculated, resulting in

$$\mathrm{Rounds} = \mathrm{Rounds}_{\mathtt{BITREP},\min(n_1,K(l+1))} + \mathrm{Rounds}_D + 3,$$

where Rounds_D represents the number of rounds needed for a threshold decryption operation.

Table 2.1 indicates the computation and precomputation complexity for the protocol of Section 2.3.1. Taking into account that the server packing operations can be performed before starting the interaction, as well as some operations in the client side, as the initial encryptions, the computational complexity for each party is given by equations (2.4) and (2.5), where $\mathrm{Cpx}_E, \mathrm{Cpx}_D$ represent the complexity of encryption and threshold decryption operations, and $\mathrm{Cpx}_{X,n^c}, \mathrm{Cpx}_{P,n^c}$ denote the complexity of modular exponentiations and products in $\mathbb{Z}_{n^c}$. The precomputation complexity for each party is indicated in equations (2.6) and (2.7), where $\mathrm{Cpx}_{SHL,n}$ and $\mathrm{Cpx}_{A,n}$ are the complexity of respectively left bit-shifts and additions of $\lceil \log_2 n \rceil$-bit numbers.

For the case of hyperelliptic regions (protocol of Section 2.3.2), the communication complexity is

$$\mathrm{Cpx}_{cm} = \left(\left\lceil \frac{K}{h} \right\rceil - 1 \right) \left(|E_P| + \mathrm{Cpx}_{cm,\mathtt{BITREP},n_1} \right) + (2N+5)|E_P| + \mathrm{Cpx}_{cm,\mathtt{BITREP},n_2}.$$

The round complexity corresponds to

$$\mathrm{Rounds} = \mathrm{Rounds}_{\mathtt{BITREP},\min(n_1,K(l+1))} + \mathrm{Rounds}_D + 3.$$

Table 2.1: Computation (cp) and Precomputation (pcp) complexity for each party (A,B) in the protocol presented in Section 2.3.1

$$
\begin{align}
\mathrm{Cpx}_{cp,A} &= \mathrm{Cpx}_D + \left(\left\lceil\frac{K}{h}\right\rceil - 1\right)\mathrm{Cpx}_{cp,\mathtt{BITREP},n_1} + \mathrm{Cpx}_{cp,\mathtt{BITREP},n_2} + \mathrm{Cpx}_{X,n^{s+1}} + (K-1)\mathrm{Cpx}_{P,n^{s+1}} \tag{2.4}\\
\mathrm{Cpx}_{cp,B} &= \mathrm{Cpx}_D + \left(\left\lceil\frac{K}{h}\right\rceil - 1\right)\mathrm{Cpx}_{cp,\mathtt{BITREP},n_1} + \mathrm{Cpx}_{cp,\mathtt{BITREP},n_2} \notag\\
&\quad + \left\lceil\frac{K}{h}\right\rceil (N+1)\cdot(\mathrm{Cpx}_{X,n^{s+1}} + \mathrm{Cpx}_{P,n^{s+1}}) + (K-1)\mathrm{Cpx}_{P,n^{s+1}} \tag{2.5}\\
\mathrm{Cpx}_{pcp,A} &= \left(\left\lceil\frac{K}{h}\right\rceil - 1\right)\mathrm{Cpx}_{pcp,\mathtt{BITREP},n_1} + \mathrm{Cpx}_{pcp,\mathtt{BITREP},n_2} + (N+1)\cdot\mathrm{Cpx}_E \tag{2.6}\\
\mathrm{Cpx}_{pcp,B} &= \left(\left\lceil\frac{K}{h}\right\rceil - 1\right)\Big(\mathrm{Cpx}_{pcp,\mathtt{BITREP},n_1} + \mathrm{Cpx}_E + \mathrm{Cpx}_{X,n^{s+1}} + (N+1)(h-1)\cdot\mathrm{Cpx}_{SHL,n^s} \notag\\
&\quad +((N+2)h-1)\cdot\mathrm{Cpx}_{A,n^s}\Big) + (N+1)\left(K - \left(\left\lceil\frac{K}{h}\right\rceil - 1\right)h - 1\right)\cdot\mathrm{Cpx}_{SHL,n^s} \notag\\
&\quad + \mathrm{Cpx}_{pcp,\mathtt{BITREP},n_2} + \mathrm{Cpx}_E + \mathrm{Cpx}_{X,n^{s+1}} + \left((N+2)\left(K - \left(\left\lceil\frac{K}{h}\right\rceil - 1\right)h\right) - 1\right)\cdot\mathrm{Cpx}_{A,n^s} \tag{2.7}
\end{align}
$$

Table 2.2: Computation (cp) and Precomputation (pcp) complexity for each party (A,B) in the protocol presented in Section 2.3.2

$$
\begin{align}
\mathrm{Cpx}_{cp,A} &= \mathrm{Cpx}_D + \left(\left\lceil\frac{K}{h}\right\rceil - 1\right)\mathrm{Cpx}_{cp,\mathtt{BITREP},n_1} + \mathrm{Cpx}_{cp,\mathtt{BITREP},n_2} + \mathrm{Cpx}_{X,n^{s+1}} + (K-1)\mathrm{Cpx}_{P,n^{s+1}} \tag{2.8}\\
\mathrm{Cpx}_{cp,B} &= \mathrm{Cpx}_D + \left(\left\lceil\frac{K}{h}\right\rceil - 1\right)\mathrm{Cpx}_{cp,\mathtt{BITREP},n_1} + \mathrm{Cpx}_{cp,\mathtt{BITREP},n_2} \notag\\
&\quad + \left\lceil\frac{K}{h}\right\rceil (2N+1)\cdot(\mathrm{Cpx}_{X,n^{s+1}} + \mathrm{Cpx}_{P,n^{s+1}}) + (K-1)\mathrm{Cpx}_{P,n^{s+1}} \tag{2.9}\\
\mathrm{Cpx}_{pcp,A} &= \left(\left\lceil\frac{K}{h}\right\rceil - 1\right)\mathrm{Cpx}_{pcp,\mathtt{BITREP},n_1} + \mathrm{Cpx}_{pcp,\mathtt{BITREP},n_2} + (2N+1)\cdot\mathrm{Cpx}_E + N\mathrm{Cpx}_{P,n^s} \notag\\
\mathrm{Cpx}_{pcp,B} &= \left(\left\lceil\frac{K}{h}\right\rceil - 1\right)\Big(\mathrm{Cpx}_{pcp,\mathtt{BITREP},n_1} + \mathrm{Cpx}_E + \mathrm{Cpx}_{X,n^{s+1}} + (2N+1)(h+N)\cdot\mathrm{Cpx}_{SHL,n^s} \notag\\
&\quad +(3N+1)h\cdot\mathrm{Cpx}_{P,n^s} + ((3N+2)h-N-1)\cdot\mathrm{Cpx}_{A,n^s}\Big) + \mathrm{Cpx}_{pcp,\mathtt{BITREP},n_2} + \mathrm{Cpx}_E + \mathrm{Cpx}_{X,n^{s+1}} \notag\\
&\quad + (3N+1)\left(K - \left(\left\lceil\frac{K}{h}\right\rceil - 1\right)h\right)\cdot\mathrm{Cpx}_{P,n^s} + (2N+1)\left(K - \left(\left\lceil\frac{K}{h}\right\rceil - 1\right)h - 1\right)\cdot\mathrm{Cpx}_{SHL,n^s} \notag\\
&\quad + \left((3N+2)\left(K - \left(\left\lceil\frac{K}{h}\right\rceil - 1\right)h\right) - N - 1\right)\cdot\mathrm{Cpx}_{A,n^s} \notag
\end{align}
$$

Computation (cp) and Precomputation (pcp) complexity is indicated in Table 2.2, following the same notation as for the complexity of the previous protocol. It is remarkable that performing the protocol on a region determined by K hyperellipsoids is of the same order as the complexity of the protocol for one polytope bounded by K hyperplanes. In fact, only some of the involved factors are doubled, so the complexity when using an unconnected region determined by K hyperellipsoids is less than twice the complexity of using a convex region determined by K hyperplanes.

In order to observe the behavior that these equations present, it is necessary to take into account the complexity of the BITREP subblock (Appendix 2.A). In light of BITREP complexity, the following analysis will be based on the use of an LSBs gate with non-constant number of rounds. Packing will slightly increase the complexity of this gate, but it will also reduce the computational complexity of the server.

Table 2.3: Communication (cm), Computation (cp) and Precomputation (pcp) Complexity for each party (A, B) in the Point Inclusion Protocol with $m = 1024$, $k = 80$ and $l = 40$ for packed $(k > 1)$ and unpacked $(k = 1)$ hyperplanes.

N	K	h	cm [kB]	cp [10^6 ops.]		pcp [10^6 ops.]		Rounds
				A	B	A	B	
100	100	1	20201	$431,26$	$514,07$	$265,69$	$266,52$	89
		20	20510	$436,72$	$440,88$	$268,80$	$268,66$	1647
5	100	1	20177	$431,26$	$436,21$	$265,50$	$266,51$	89
		$--$	$--$	$--$	$--$	$--$	$--$	$--$
100	10	1	$2043,5$	$43,156$	$51,429$	$26,755$	$26,652$	89
		10	$2072,8$	$43,672$	$44,495$	$27,050$	$26,855$	827

Table 2.3 gives a glimpse of the complexity of the whole protocol with representative values of the parameters, comparing the effect of no packing with packing the number of hyperplanes that gives the minimum server complexity. The units employed for computation complexity are modular additions in the ring $\mathbb{Z}_n$, and we consider that additions have linear complexity $\mathcal{O}(sn)$, multiplications have quadratic complexity $\mathcal{O}\left((sn)^2\right)$, and exponentiations, cubic $\mathcal{O}\left((sn)^3\right)$.

The number of packed hyperplanes/hyperellipsoids represents a trade-off between computation and precomputation complexity at the server side. We can see that the second term in Cpx_B (Eq. (2.5)) is increasing in h, due to the non-sublinear behavior of the BITREP complexity, while the fourth term is decreasing in h. Thus, when N is high, the fourth term (the homomorphically performed encrypted operations) is the dominating one, what justifies that the h that produces the minimum complexity is the maximum number of hyperplanes that can be packed without increasing s; on the other hand, communication complexity and the computation complexity for the client are still dominated by the BITREP complexity, so their increase after packing is almost negligible. Nevertheless, when N is low, the dominating term for server complexity is the second one (BITREP), so there is no complexity gain with packing.

Finally, as mentioned before, using $s > 1$ produces an increase in the complexity of every encrypted operation, as the size of the operands is multiplied by this factor, so a value $s > 1$ is not justified for this protocol.

2.4. Secure Linear Algebra

The second block of secure primitives that we address in this chapter consists in protocols for solving systems of linear equations (SLEs). For all these protocols, we consider two parties, $\mathcal{A}$ and $\mathcal{B}$, both using an additively homomorphic cryptosystem (for whose encryptions we adopt the notation $[\![.]\!]$) in an asymmetric scenario, where $\mathcal{A}$ can only encrypt, but $\mathcal{B}$ possesses also the decryption key, and can perform both encryption and decryption. For the problem of solving an SLE $\boldsymbol{A} \cdot \boldsymbol{x} = \boldsymbol{b}$ [74], we will consider that $\mathcal{A}$ owns an encrypted version of the system matrix $[\![\boldsymbol{A}]\!]$, and of the independent vector $[\![\boldsymbol{b}]\!]$. This scenario can be straightforwardly reached from many initial situations, covering all the possible ways of sharing $\boldsymbol{A}$ and $\boldsymbol{b}$ between both parties. For the sake of brevity, we focus on this initial situation, and obviate the way of reaching it.

The assumption we make about the system consists in $\boldsymbol{A}$ being either a positive definite matrix or a strictly diagonally dominant matrix, in order to guarantee both a solution to the system and the convergence of the studied methods, as will be detailed later. This assumption is not a severe limitation, as many matrices found in statistics calculations fulfill it [74].

Regarding the privacy requirements, we will assume that both parties are semi-honest, in the sense that they will adhere to the established protocol, but they can be curious about the information they can get from the interaction. In this scenario, our protocols can be proven private; informally, both parties $\mathcal{A}$ and $\mathcal{B}$ can only get the information leaked from the solution to the system, and no information is leaked from the intermediate steps of the protocols.

As sparsity of the matrices cannot be used as an advantage under encryption, we will focus only on direct methods for solving linear systems of equations (Section 2.4.1), and we will not cope with methods based in decompositions of the system matrix (like LU or Cholesky decomposition). Furthermore, we will also provide protocols for iterative methods of SLE solving (Section 2.4.2) and matrix inversion (Section 2.4.3).

2.4.1. Direct method: Gaussian elimination

Firstly, we will implement the method of Gaussian elimination, using a secure multiplication protocol (cf. Appendix 2.C) for implementing the needed multiplications. Due to the lack of a division operation under encryption, the obtained result vector is scaled, but the scale factors are stored in a second vector $\boldsymbol{s}$, so that the solution can be recovered after decryption through a component-wise division. The protocol ends with two vectors $\boldsymbol{x}'$ and $\boldsymbol{s}$, being the solution to the system $x_i = \frac{x'_i}{s_i}, \quad i = 0, \ldots, L-1$.

Let $\boldsymbol{A} \in \mathcal{M}_{L\times L}(\mathbb{Z})$ be a quantized symmetric positive-definite matrix, or a diagonally dominant matrix, and $\boldsymbol{b} \in \mathbb{Z}^L$ be a quantized column vector. The quantization step Δ is such that the absolute value of every quantized element is upper bounded by a constant τ.

In our scenario, we will assume that $\mathcal{B}$ knows the decryption key of an additive homomorphic cryptosystem, and both $\mathcal{A}$ and $\mathcal{B}$ can produce encryptions using this cryptosystem; $\mathcal{A}$ possesses the encrypted matrix $[\![\boldsymbol{A}]\!]$ and the encrypted vector of independent terms $[\![\boldsymbol{b}]\!]$. Both parties will engage in an interactive protocol in order to obtain the solution $\boldsymbol{x}$ to the linear system $\boldsymbol{A}\cdot\boldsymbol{x} = \boldsymbol{b}$. The protocol is sketched next.

Following the Gaussian elimination algorithm, we will call $\boldsymbol{G}^{(0)} = \boldsymbol{G}$ to the concatenation of $\boldsymbol{G} = [\boldsymbol{A}|\boldsymbol{b}]$. The algorithm is executed in $L-1$ steps. At each step k, the matrix $\boldsymbol{G}$ is modified for obtaining an equivalent system $\boldsymbol{G}^{(k)}$ in which the k-th unknown is not present in the last $L-k$ equations.

For the k-th step of the algorithm, the first $k-1$ elements of the $L-k+1$ last rows of $\boldsymbol{G}^{(k-1)}$ are zero; $\mathcal{A}$ owns an encrypted version of the non-zeroed elements of $\boldsymbol{G}^{(k-1)}$. The secure protocol proceeds as follows

1. $\mathcal{A}$ provides randomized encrypted versions of the submatrix $\boldsymbol{C}^{(k)}$ formed by the last $(L-k+2)$ columns of the last $(L-k+1)$ rows of $\boldsymbol{G}^{(k-1)}$;
2. $\mathcal{B}$, through decryption and reencryption, calculates the (randomized) products of the $(L-k)\times(L-k+1)$ matrices $\boldsymbol{D}^{(k)}$ and $\boldsymbol{E}^{(k)}$, defined as $\left[\!\!\left[d^{(k)}_{j,m}\right]\!\!\right] = \left[\!\!\left[c^{(k)}_{0,m+1}\cdot c^{(k)}_{j+1,0}\right]\!\!\right]$, and $\left[\!\!\left[e^{(k)}_{i,j}\right]\!\!\right] = \left[\!\!\left[c^{(k)}_{0,0}\cdot c^{(k)}_{i+1,j+1}\right]\!\!\right]$, and sends the randomized encryptions to $\mathcal{A}$.
3. $\mathcal{A}$ derandomizes the received encryptions and, using homomorphic operations, obtain the next iteration of $\boldsymbol{G}$:

$$[\![\boldsymbol{G}^{(k)}]\!] = \left(\begin{array}{c|c} \multicolumn{2}{c}{\left\{\left[\!\!\left[g^{(k-1)}_{i,m}\right]\!\!\right]\right\}^{(k-1,L)}_{(0,0)}} \\ \hline \mathbf{0}_{L-k,k} & [\![\boldsymbol{F}^{(k)}]\!] \end{array}\right),$$

   where $[\![\boldsymbol{F}^{(k)}]\!]$ is an $(L-k)\times(L-k+1)$ matrix with elements $\left[\!\!\left[f^{(k)}_{i,m}\right]\!\!\right] = \left[\!\!\left[e^{(k)}_{i,m}\right]\!\!\right] - \left[\!\!\left[d^{(k)}_{i,m}\right]\!\!\right]$.

After $L-1$ iterations, $\mathcal{A}$ has $[\![\boldsymbol{G}^{(L-1)}]\!]$, an encrypted upper triangular matrix appended to an encrypted vector, that constitute a system with the same solution as the original one.

In order to solve the system, both parties initiate the process of back substitution under encryption, consisting in L iterations: in each iteration, an element of the vector $\boldsymbol{x}'$ and the corresponding element of the scale vector $\boldsymbol{s}$ are obtained. As they will be revealed as the output, and they are needed in order to calculate the subsequent elements of $\boldsymbol{x}'$, they can be decrypted before the next iteration in order to lower the complexity by reducing the number of the needed multiplication protocols. For the first step:

1. $\mathcal{A}$ sends $\{\llbracket g_{i,i}^{(L-1)} \rrbracket\}_{i=0}^{L-1}$ and $\llbracket g_{L-1,L}^{(L-1)} \rrbracket$.

2. $\mathcal{B}$ obtains, through decryption, the scaling vector $\boldsymbol{s}$, with $s_i = \prod_{l=i}^{L-1} g_{l,l}^{(L-1)}$, and the value $x'_{L-1} = g_{L-1,L}^{(L-1)}$, and sends them back to $\mathcal{A}$.

In each subsequent k-th step, $\mathcal{A}$ calculates, using homomorphic operations:

$$\llbracket x'_{L-k} \rrbracket = \llbracket g_{L-k,L}^{(L-1)} \rrbracket \cdot s_{L-k+1} - \sum_{l=L-k+1}^{L-1} \llbracket g_{L-k,l}^{(L-1)} \rrbracket \cdot x'_l \frac{s_l}{s_{L-k+1}},$$

and sends $\llbracket x'_{L-k} \rrbracket$ to $\mathcal{B}$ to obtain its decryption.

With the proposed protocol, we are not disclosing any element of the original matrix $\boldsymbol{A}$ nor of the independent terms vector $\boldsymbol{b}$. Furthermore, every step of the protocol can be proven secure with semihonest parties, due to the semantical security of the underlying homomorphic cryptosystem, the security of the used multiplication protocols, and the fact that all the unencrypted values (besides the result and the scaling vector) that each party can access are random and uncorrelated. Although the scaling vector reveals the diagonal of the upper triangular matrix of an equivalent system, which gives information about the eigenvalues of the original matrix, this information affects L scaled elements out of $\frac{L(L+1)}{2}$.

It must be noted that having the values of the principal diagonal of the upper-triangular matrix of the equivalent system yields the possibility of calculating its condition number, or at least, its bound

$$\kappa(\boldsymbol{U}) \geq \frac{\max_i(|u_{ii}|)}{\min_i(|u_{ii}|)}.$$

Thus, this disclosure constitutes a clear advantage in terms of conditioning and efficiency: before executing the back substitution protocol, the rows of $\boldsymbol{G}^{(L)}$ can be multiplied by appropriate factors in order to lower the condition number and minimize error propagation due to working with a fixed point precision. Also, the vector of multiplicative factors s_i can be adequately quantized in the clear to achieve this same goal.

As a last remark, this protocol does not limit the number N of SLEs sharing the same system matrix $\boldsymbol{A}$ and with different independent term vectors $\boldsymbol{b}_i$ that can be solved in parallel; all the vectors $\boldsymbol{b}_i$ can be appended to the system matrix, forming a $L \times (L+N)$ matrix $\boldsymbol{G}_{ext}$ and in each step of the previous protocol, the operations that must be performed on the last column of $\boldsymbol{G}^{(k)}$ will be replicated for the last N columns of $\boldsymbol{G}_{ext}^{(k)}$.

2.4.1.1. Complexity

When solving one system $\boldsymbol{A} \cdot \boldsymbol{x} = \boldsymbol{b}$, the Gaussian Elimination (GE) protocol is performed in $(L-1)$ rounds of communication, with total complexity

$$\begin{aligned}
\mathrm{Cpx}_{cm,GE} &= (L^3 + L^2 - 2)\\
\mathrm{Cpx}_{cp,GE,A} &= \frac{1}{3}\left(L^3 + 3L^2 + 2L - 6\right)\mathrm{Cpx}_E +\\
&\quad \frac{1}{3}\left(2L^3 + 3L^2 + L - 6\right)\mathrm{Cpx}_{EA}\\
\mathrm{Cpx}_{cp,GE,B} &= \frac{1}{3}\left(L^3 + 3L^2 + 2L - 6\right)\mathrm{Cpx}_D +\\
&\quad \frac{2}{3}\left(L^3 - L\right)(\mathrm{Cpx}_E + \mathrm{Cpx}_P).
\end{aligned}$$

The protocol of Back Substitution (BS) is performed in L rounds of communication, with total complexity

$$\begin{aligned}
\mathrm{Cpx}_{cm,BS} &= 2L \cdot (1 + ct)\\
\mathrm{Cpx}_{cp,BS,A} &= \frac{1}{2}(L^2 + L - 2)\mathrm{Cpx}_{EP} + \frac{1}{2}(L^2 - L)\mathrm{Cpx}_{EA}\\
\mathrm{Cpx}_{cp,BS,B} &= 2L\mathrm{Cpx}_D.
\end{aligned}$$

2.4.1.2. Representable numbers

We have assumed that the coefficients of the system matrix $\boldsymbol{A}$ are quantized versions of the real-valued coefficients, with a quantization step Δ. Furthermore, the absolute value of the quantized coefficients is bounded by an integer $\tau > 0$. Then, it is possible to estimate the value of τ needed to fit all the performed operations inside a cipher that can represent integers in the range $[0, n)$ without rounding problems.

For the first part of the protocol (the Gaussian elimination), each iteration multiplies two numbers that were obtained in the previous iteration and adds them up, so the previous bound gets squared and doubled:

$$|t_1|, |t_2|, |t_3|, |t_4| < \tau \Rightarrow |t_1 \cdot t_2 - t_3 \cdot t_4| < 2\tau^2.$$

Then it is straightforward to conclude that all the elements of the k-th row of the resulting $\boldsymbol{G}^{(L-1)}$ will be bounded by $(2^{2^{k-1}-1})\tau^{2^k}$, and will constitute the representation of their real-valued equivalents, quantized by $\Delta^{2^{k-1}}$. Thus, the cipher must be such that $n > (2^{2^{k-1}-1})\tau^{2^k}$ in order to fit all the numbers involved in this protocol. This means that the bit size of the modulus of the cipher must grow exponentially with the dimensionality of the system, what gives a poor scalability.

For the second part of the protocol, after the diagonal elements are disclosed, they can be requantized in order to make them relative to the lowest scale and lower the bit-size requirements of the cipher; but in the worst case, without requantizing the scale factors, the largest number present after running the whole protocol will be $2^{2^L-L-1}\tau^{2^{1+L}-4}$. That will also constrain the size of the cipher.

2.4.2. Iterative methods: Jacobi's Method

The general form of *stationary iterative methods* for solving SLEs is

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{M} \cdot \boldsymbol{x}^{(k)} + \boldsymbol{c}.$$

Jacobi's method is a particular case of stationary iterative methods, where the system matrix is decomposed into $\boldsymbol{A} = \boldsymbol{D}(\boldsymbol{L} + \boldsymbol{I} + \boldsymbol{U})$, a diagonal matrix $\boldsymbol{D}$, a lower triangular matrix $\boldsymbol{L}$ and an upper triangular matrix $\boldsymbol{U}$, having both $\boldsymbol{L}$ and $\boldsymbol{U}$ zeros in their principal diagonals. Then, $\boldsymbol{M} = -(\boldsymbol{L} + \boldsymbol{U})$ and $\boldsymbol{c} = \boldsymbol{D}^{-1}\boldsymbol{b}$. As divisions are not supported homomorphically, the previous iteration cannot be implemented directly. Thus, the division is simulated by multiplying each row of $\boldsymbol{A}$ by the diagonal elements of the remaining rows, what results in multiplying the matrix $\boldsymbol{M}$ of Jacobi's method by a scalar factor $\gamma = \left(\prod_{i=0}^{L-1} a_{ii}\right)$.

$$\boldsymbol{A}' = -\gamma \boldsymbol{D}^{-1} \cdot (\boldsymbol{A} - \boldsymbol{D}) = \gamma \boldsymbol{M}.$$

The factor γ will be propagated at every iteration of the algorithm:

$$\gamma^k \boldsymbol{x}^{(k)} = -\gamma \boldsymbol{D}^{-1} (\boldsymbol{A} - \boldsymbol{D}) \cdot \gamma^{k-1} \boldsymbol{x}^{(k-1)} + \gamma^k \boldsymbol{D}^{-1} \boldsymbol{b}.$$

Let us assume that $\mathcal{B}$ can decrypt and both $\mathcal{A}$ and $\mathcal{B}$ can encrypt with an additive homomorphic scheme, and that $\mathcal{A}$ owns encryptions of $[\![\boldsymbol{A}]\!]$ and $[\![\boldsymbol{b}]\!]$ with this homomorphic system. In order to allow for efficient computation, the following protocol is executed:

1. $\mathcal{A}$ can blind the principal diagonal of $[\![\boldsymbol{A}]\!]$ and send it to $\mathcal{B}$.

2. $\mathcal{B}$ decrypts it, both parties ending up with additive shares of the diagonal elements $\{a_{ii}\}$.

3. With this shares, both parties can securely compute shares of the diagonal matrix $(\gamma \boldsymbol{D}^{-1})_{jj} = \prod_{\substack{i=0 \\ i \neq j}}^{L-1} a_{ii}$, through $\lceil \log_2(L-1) \rceil$ rounds of parallel secure multiplication protocols. They can also calculate the value of γ, and disclose it for use in the following steps of the protocol.

4. $\mathcal{A}$ can then calculate the encryption of $[\![\gamma \boldsymbol{M}]\!] = [\![-\gamma \boldsymbol{D}^{-1}]\!] \cdot [\![\boldsymbol{A} - \boldsymbol{D}]\!]$ and $[\![\gamma \boldsymbol{c}]\!] = [\![\gamma \boldsymbol{D}^{-1}]\!] \cdot [\![\boldsymbol{b}]\!]$, invoking the secure multiplication protocol.

5. Then, $\mathcal{A}$ sends $\mathcal{B}$ a blinded and encrypted version of $[\![\gamma \boldsymbol{M}]\!]$, that $\mathcal{B}$ decrypts for use in the following iterations.

After these initial steps, for the *first iteration* of the secure protocol both parties agree in an initial vector $\boldsymbol{x}^{(0)}$ and $\mathcal{A}$ calculates, through homomorphic additions and multiplications, the encryption of $[\![\gamma \boldsymbol{x}^{(1)}]\!] = [\![\gamma \boldsymbol{M}]\!] \cdot \boldsymbol{x}^{(0)} + [\![\gamma \boldsymbol{c}]\!]$.

For *each subsequent iteration*, $\mathcal{A}$ calculates the encryption of $\gamma \cdot [\![\gamma^{k-1} \boldsymbol{c}]\!]$, and then both parties use the secure multiplication protocol of Appendix 2.C and homomorphic additions in order to obtain the vector for the following step

$$[\![\gamma^k \boldsymbol{x}^{(k)}]\!] = [\![\gamma \boldsymbol{M}]\!] \cdot [\![\gamma^{k-1} \boldsymbol{x}^{(k-1)}]\!] + \gamma \cdot [\![\gamma^{k-1} \boldsymbol{c}]\!] \, .$$

It must be noted that the matrix $[\![\gamma \boldsymbol{M}]\!]$ does not have to be communicated at each iteration, as its blinded version was stored by $\boldsymbol{B}$ at the initial step. Thus, only two vectors per iteration are sent between $\mathcal{A}$ and $\mathcal{B}$.

After each iteration, the factor γ multiplies the result; thus, after a number of steps, the cipher will not be able to accommodate the scaled number, and the protocol will have to stop. This is studied in more depth in Section 2.4.4. It must be noted that the accumulated factor is not only γ, but also the quantization step Δ used for the initial quantization of the coefficients of both the system matrix $\boldsymbol{A}$ and the vector $\boldsymbol{b}$ in order to make them integers so that they can be encrypted. This factor must also be taken into account every time γ multiplies vector $\boldsymbol{c}$, so that the homomorphically added vectors be quantized with the same scaling factor.

Lastly, each step of the protocol can be proven secure with semihonest parties, due to the semantic security of the underlying cryptosystem, the security of the multiplication protocol, and the fact that the unencrypted values that each party sees are random and uncorrelated.

2.4.2.1. Complexity

The complexity of the initial part (Jacobi Initial, JI) of the protocol is

$$\begin{aligned}
\mathrm{Cpx}_{cm,JI} &= 3L^2 + 2L\lceil \log_2(L-1)\rceil - 3L + 5 + ct\\
\mathrm{Cpx}_{cp,JI,A} &= (5L^2 + 4L\lceil \log_2(L-1)\rceil - 5L + 8)\mathrm{Cpx}_{EA} +\\
&\quad (L^2 + L\lceil \log_2(L-1)\rceil - L + 2)2\mathrm{Cpx}_{EP}\\
\mathrm{Cpx}_{cp,JI,B} &= (L^2 + L\lceil \log_2(L-1)\rceil - L + 2)\left(\mathrm{Cpx}_{D} + \mathrm{Cpx}_{P} +\right.\\
&\quad \left.\mathrm{Cpx}_{E}\right) + (L^2 - L + 1)\mathrm{Cpx}_{D}.
\end{aligned}$$

The first iteration ($J1$) does not involve any interaction, and $\mathcal{A}$ incurs in a computational complexity of $\mathrm{Cpx}_{cp,J1,A} = L^2(\mathrm{Cpx}_{EP} + \mathrm{Cpx}_{EA})$.

The complexity of each of the subsequent iterations of this protocol (J) is the following

$$\begin{aligned}
\mathrm{Cpx}_{cm,J} &= 2L\\
\mathrm{Cpx}_{cp,J,A} &= (3L^2 - 2L)\mathrm{Cpx}_{EA} + (2L^2 - L)\mathrm{Cpx}_{EP} + L\mathrm{Cpx}_{EA}\\
\mathrm{Cpx}_{cp,J,B} &= L\left(\mathrm{Cpx}_{D} + \mathrm{Cpx}_{E}\right) + (L^2 - L)\mathrm{Cpx}_{P} + (L^2 - 2L)\mathrm{Cpx}_{A}.
\end{aligned}$$

After a number of iterations, either the solution can be disclosed, or an error metric can be obtained to determine whether convergence has been achieved. While the choice of this error metric is arbitrary, one possibility is to homomorphically subtract $[\![\boldsymbol{x}^{(k)}]\!] - [\![\boldsymbol{x}^{(k-1)}]\!]$, and either decrypt the result or perform L parallel encrypted comparisons with a predetermined threshold.

2.4.2.2. Representable numbers

As for the direct method, we have assumed that the coefficients of the system matrix $\boldsymbol{A}$ are quantized versions of the real-valued coefficients, with a quantization step Δ, such that their quantized absolute value is bounded by an integer $\tau > 0$.

For the first part of the protocol, where the factor γ and the matrix $\gamma\boldsymbol{D}^{-1}$ are calculated, γ is the highest number that the system will have to represent, and it is bounded by τ^L; the bound for the elements of $\gamma\boldsymbol{D}^{-1}$ is τ^{L-1}. Furthermore, as γ is disclosed in the following step, it can constitute a more accurate bound to the encrypted coefficients of $\gamma\boldsymbol{D}^{-1}$. A bound for the absolute value of the coefficients of $\gamma\boldsymbol{M}$ and of $\gamma\boldsymbol{c}$ is $\min(\tau^L, \gamma\tau)$.

Lastly, the bound to which the elements of the first vector $\boldsymbol{x}^{(1)}$ are subject is $L \cdot \tau^{L+1}$.

In each iteration, the previous bound is multiplied by $L \cdot \tau^{L}$, meaning that the bound for the elements of the k-th iteration is $L^k \cdot \tau^{k \cdot L+1}$, i.e., the needed bit-size of the cipher is linear both in the dimension of the system and in the maximum number of iterations that can be performed without errors. Furthermore, the quantization step of the elements of $\boldsymbol{x}^{(k)}$ will be Δ^{kL}.

2.4.2.3. Convergence of the algorithm

When dealing with iterative algorithms like Jacobi's, it is necessary to determine whether the algorithm can converge or not before applying it. In the general case of *stationary iterative methods*, the necessary and sufficient condition for their convergence with an arbitrary initial vector $\boldsymbol{x}^{(0)}$ is that $\max_i |\lambda_i(\boldsymbol{M})| < 1$, where $\lambda_i(\boldsymbol{M})$ are the eigenvalues of $\boldsymbol{M}$. For Jacobi's method, $\boldsymbol{M} = -\boldsymbol{D}^{-1} \cdot (\boldsymbol{A} - \boldsymbol{D})$. Let us assume that $\boldsymbol{A}$ is a strictly diagonally dominant matrix with bounded coefficients $|a_{ij}| \leq \tau$. By Ostrowski's theorem [124], the eigenvalues of $\boldsymbol{M}$ are located in the union of L discs

$$\mathcal{L}_1 \triangleq \bigcup_{i=0}^{L-1} \{z \in \mathbb{C} : |z - m_{ii}| \leq \min\{R_i, C_i\}\},$$

where $m_{ii} = 0$, $m_{ij} = \frac{a_{ij}}{a_{ii}}, i \neq j$, and

$$R_i = \sum_{j=0, j\neq i}^{L-1} |m_{ij}|,$$
$$C_i = \sum_{j=0, j\neq i}^{L-1} |m_{ji}|.$$

As $\boldsymbol{A}$ is strictly diagonally dominant, $\sum_{j=0,j\neq i}^{L-1} |a_{ij}| < |a_{ii}| \Rightarrow R_i < 1$. Thus, it is possible to bound the moduli of the eigenvalues of $\boldsymbol{M}$ as

$$|\lambda_i(\boldsymbol{M})| < 1.$$

Then, Jacobi method always converges for strictly diagonally dominant matrices, and the test of convergence is not needed.

2.4.3. Matrix inversion through iterative methods

There are cases in which, instead of or additionally to solving a SLE, the inverse of the system matrix is also needed, like the case of regression analysis in statistics. For these applications, the system matrix $\boldsymbol{A}$ must be inverted. As the direct method (through Cramer's rule) is computationally too expensive, we will provide a secure protocol for performing the execution of an iterative method, namely Newton's method. One iteration of this method has the following expression

$$\boldsymbol{X}^{(k)} = \boldsymbol{X}^{(k-1)} \cdot \left(2\boldsymbol{I} - \boldsymbol{A}\boldsymbol{X}^{(k-1)}\right),$$

where $\boldsymbol{X}^{(k)}$ will converge to $\boldsymbol{A}^{-1}$.

The secure protocol for Newton's method will execute an *initial iteration* with an agreed initial value $\boldsymbol{X}^{(0)}$, performed uniquely with homomorphic operations. Then, the *following iterations* make use of the secure multiplication protocol (cf. Appendix 2.C) and homomorphic sums. Each iteration needs two rounds of communication:

1. The first one to calculate $[\![\boldsymbol{Q}^{(k)}]\!] = [\![\boldsymbol{A}]\!] \cdot [\![\boldsymbol{X}^{(k-1)}]\!]$,
2. the second one to calculate $[\![\boldsymbol{X}^{(k)}]\!] = [\![\boldsymbol{X}^{(k-1)}]\!] \cdot \left(2\boldsymbol{I} - [\![\boldsymbol{Q}^{(k)}]\!]\right)$.

As with any iterative method, the result gets multiplied after each iteration by the quantization step of the used integers, so after a sufficiently high number of iterations, as with Jacobi's method, the cipher will not be able to accommodate the scaled numbers, and the protocol will stop (cf. Section 2.4.4).

Lastly, the protocol is provably secure with semihonest parties due to the semantic security of the cryptosystem, and the security of the sequentially composed multiplication protocols.

2.4.3.1. Complexity

The first step involves only one round of interaction, and its complexity is given by

$$\begin{aligned}
\mathrm{Cpx}_{cm,NEWI} &= L^2 \\
\mathrm{Cpx}_{cp,NEWI,A} &= 2L^3\mathrm{Cpx}_{EP} + (2L^3 - 2L^2 + L)\mathrm{Cpx}_{EA} \\
\mathrm{Cpx}_{cp,NEWI,B} &= 0.
\end{aligned}$$

The complexity of each of the subsequent iterations of this protocol is the following

$$\begin{aligned}
\mathrm{Cpx}_{cm,NEW} &= 2\mathrm{Cpx}_{cm,MULT}(L, L, L) \\
\mathrm{Cpx}_{cp,NEW,A} &= 2\mathrm{Cpx}_{cp,MULT,A}(L, L, L) + L\mathrm{Cpx}_{EA} \\
\mathrm{Cpx}_{cp,NEW,B} &= 2\mathrm{Cpx}_{cp,MULT,B}(L, L, L).
\end{aligned}$$

2.4.3.2. Representable numbers

Let us assume that the elements of the matrix $\boldsymbol{A}$ are quantized with a quantization step Δ, and their absolute value is bounded by $\tau > 0$. Then, the elements of matrix resulting from the first iteration of the protocol are bounded by $L^2\tau^3 + 2\tau$. For each of the next iterations, the bound $\tau^{(k-1)}$ is updated as $\tau^{(k)} = \left(\tau^{(k-1)}\right)^2 \cdot \tau \cdot L^2 + 2 \cdot K$. Thus, the order of the bound after m iterations is $O\left(\tau^{2^{m+1}-1} \cdot L^{2^{m+1}-2}\right)$, i.e. the bit-size of the cipher is exponential in the number of iterations, like for the direct algorithm of Section 2.4.1.

2.4.3.3. Convergence of the algorithm

The convergence of Newton's method is assured whenever the initial matrix $\boldsymbol{X}^{(0)}$ satisfies $||\boldsymbol{A}\boldsymbol{X}^{(0)} - \boldsymbol{I}|| < 1$. As the initial vector is chosen by both parties, it can be such that this condition is fulfilled, given the bounds to the elements of $\boldsymbol{A}$ and the bounds to the eigenvalues obtained by the application of Ostrowski's theorem. This way, as for Jacobi's method, it would be unnecessary to check for convergence through an additional interactive protocol.

2.4.4. Practical Implementation

We study a practical implementation of the proposed protocols for SLEs, and comment on the obtained results. For this purpose, we have chosen Damgård-Jurik [79] extension of Paillier cryptosystem, due to its flexibility for fitting larger plaintexts with a constant expansion ratio. With the complexity calculations shown in the previous section, we will exemplify the figures to which the presented protocols lead, with different parameters.

Firstly, we evaluate the needed size of the cleartext group in order to safely fit all the involved numbers without errors in their representation. Figure 2.1 shows the bit-size of the plaintext group when coefficients are bounded by 2^{32} and for an SLE with $L = 10$ equations (10×10 system matrix), which are reasonable sizes for common applications. The direct method needs plaintexts of about 2^{15} bits for getting the solution of the system; this is a reasonable figure, taking into account that the output will be directly the solution of the system. On the other hand, the two studied iterative systems have a very different behavior, as anticipated by the calculations of Section 2.4. While the needed size of the plaintext in the protocol implementing Newton's method grows exponentially with the number of iterations, needing more than 2^{16} bits to fit the represented numbers after 10 iterations, Jacobi's method is far more conservative, needing less than 2^{13} bits for the same number of iterations, and with a linear growth.

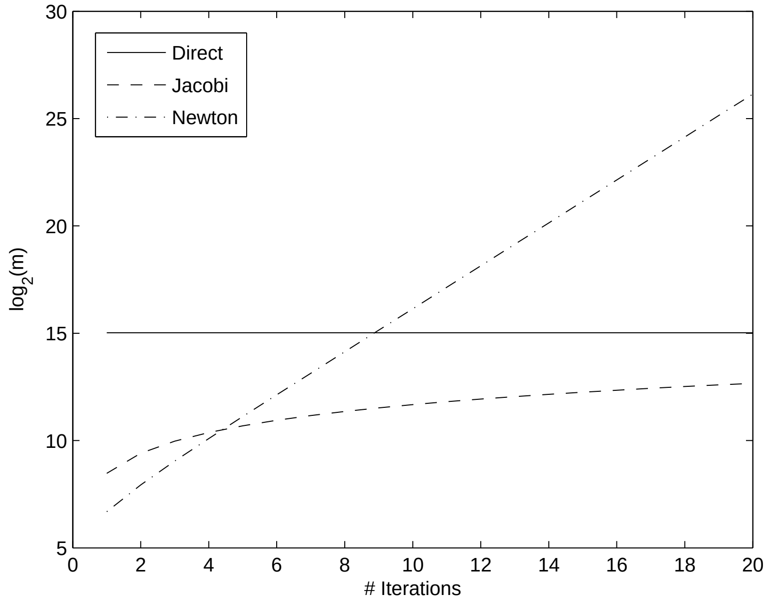


Figure 2.1: Logarithm of the bit-size of the plaintext group as a function of the performed iterations for $\tau = 2^{32}$ and $L = 10$ dimensions.

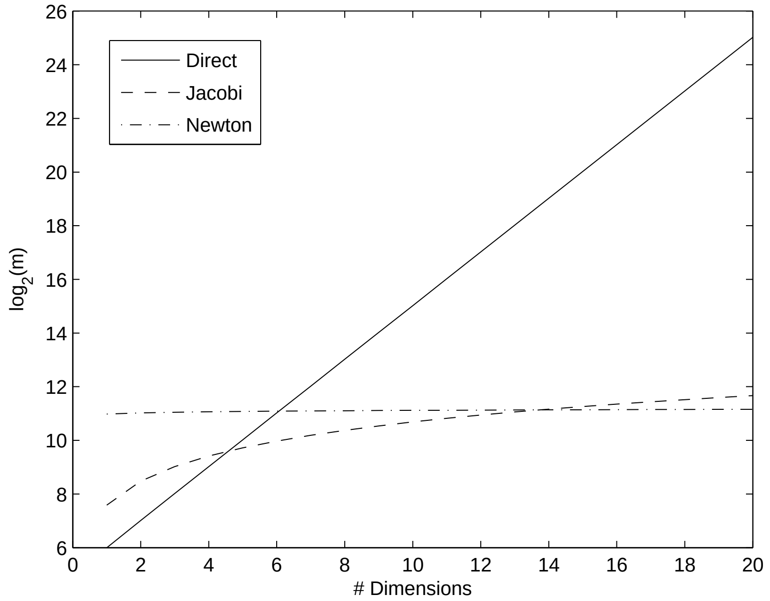


Figure 2.2: Logarithm of the bit-size of the plaintext group as a function of the number of dimensions, for $\tau = 2^{32}$ and 5 iterations.

Figure 2.2 shows also the size of the plaintext, varying the dimensionality of the problem; for the iterative algorithms, the number of performed iterations is fixed at 5. This time the direct protocol shows its exponential dependence on the dimensionality of the problem, while the protocol for Jacobi's method is linear, and Newton's algorithm logarithmic. This is a factor that is worth considering when inverting matrices with a high dimensionality.

Regarding the complexity of the developed protocols in terms of communication and computation given a maximum plaintext size, Figure 2.3 plots the communication complexity measured in bits for the three protocols when solving a system with 5 unknowns, and varying the parameter s of Damgård-Jurik cryptosystem, that gives a plaintext size of $s \cdot m$, where m has been fixed here to 1024 bits. For this system, the minimum s accepted by the direct protocol is $s = 2$. For the iterative protocols, the complexity is calculated for the maximum number of iterations that the size of the cipher can correctly fit. This quantity is indicated in Table 2.4.

Table 2.4: Number of allowed iterations as a function of s, with $\tau = 2^{32}$ and $L = 5$ dimensions.

s	2	3	4	5	6
Jacobi[iters]	13	19	26	32	38
Newton[iters]	4	5	5	6	6

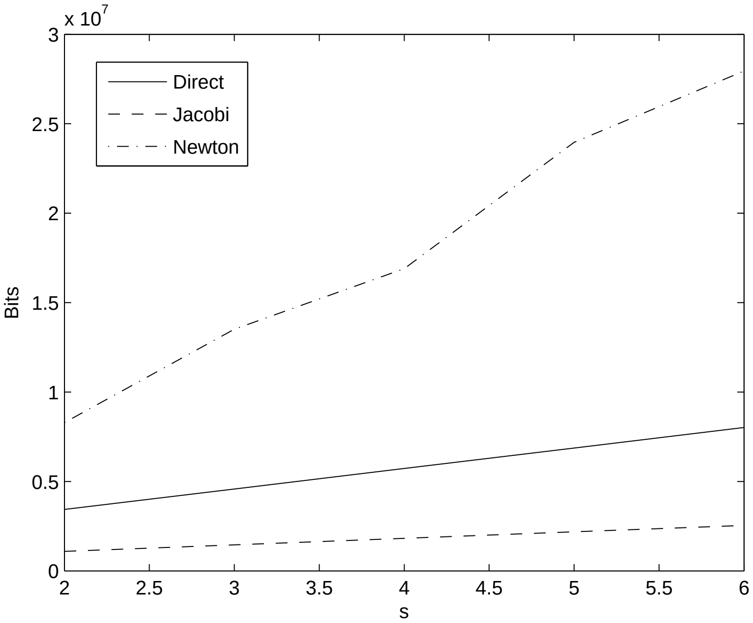


Figure 2.3: Communication complexity of the presented protocols as a function of s, with $\tau = 2^{32}$ and $L = 5$ dimensions.

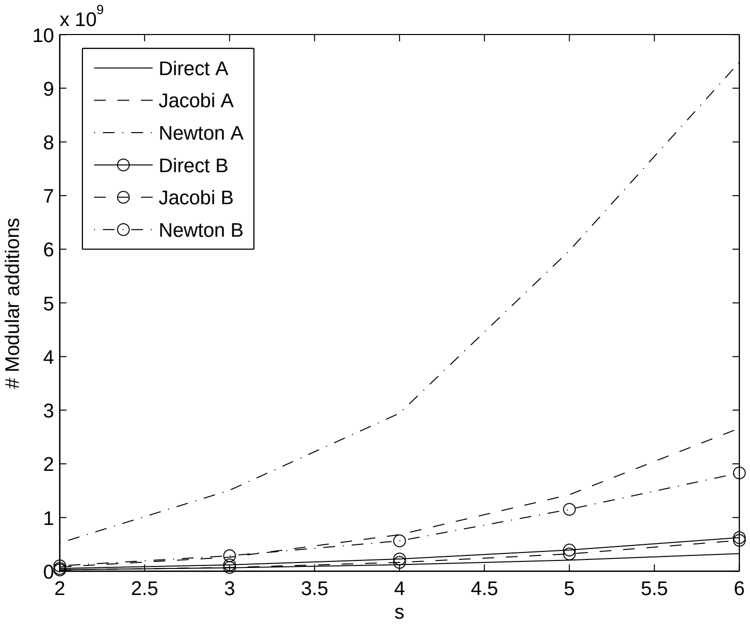


Figure 2.4: Computational complexity of the presented protocols as a function of s, with $\tau = 2^{32}$ and $L = 5$ dimensions.

While the communication complexity of the three protocols is approximately linear on s, the protocol for matrix inversion needs much more communication, with a number of iterations limited by the maximum size of the plaintext. On the other hand, the protocol that implements Jacobi's method is much more efficient, as it can perform a much larger number of iterations within the same plaintext size, while incurring in a lower complexity.

The same behavior can be observed in terms of computational complexity (Figure 2.4), that is approximately quadratic on s for the three protocols, but the multiplicative constants are much larger for Newton's protocol than for the other two. As a function of the number of dimensions, the protocol implementing Jacobi's method is also much better behaved than the other two methods, but still needs a large plaintext size when a high number of iterations must be performed. We have not included more plots illustrating this behavior due to space limitations.

Summarizing, the needed bit size for the three protocols is relatively high, and for the case of the protocol for Newton's method it grows exponentially with the number of performed iterations. Jacobi's is far more efficient, as it can accommodate a much larger number of iterations using the same maximum plaintext size. The complexity of the protocol for Jacobi's method is also lower than the other two methods for a sufficiently high number of dimensions.

Nevertheless, in order to perform an arbitrary number of iterations, and to lower the complexity of the three protocols, it would be desirable to have a means for renewing the cipher with a lower scale factor. It must be noted that, even when having a full (algebraic) homomorphic cryptosystem, this problem cannot be avoided. The full homomorphism would allow for performing all the operations without interaction, considerably lowering the communication complexity, as well as the computational complexity (depending on how the homomorphic operations must be performed). Nevertheless, with a fully homomorphic cryptosystem the growth of the ciphered numbers would be also unavoidable if the requantization cannot be implemented homomorphically.

2.5. Secure Execution of Finite Automata

The third class of secure primitives presented in this chapter deal with the problem of obliviously running an automaton. This can be informally stated as an asymmetric function evaluation problem, in which one party possesses a function f and the other party owns the input x to that function. One or both of those parties want to obtain the evaluation of $f(x)$, but neither party wants to disclose his own data. Here, the function f is implemented as a finite state machine (FSM), and the output $f(x)$ encodes whether x was accepted by the FSM.

More formally, let $(Q, \Sigma, \boldsymbol{\Delta}, q_0, F)$ be a deterministic FSM, whose description is owned by party $\mathcal{A}$. Let $\boldsymbol{x} = x_0 x_1 \dots x_{N-1} \in \Sigma^N$ be an input to that FSM; $\boldsymbol{x}$ is owned by party $\mathcal{B}$. Both parties want to run the FSM on $\mathcal{B}$'s input, in such a way that $\mathcal{A}$ will not get any information about the input string $\boldsymbol{x}$ except its length, and the only information $\mathcal{B}$ can get about the FSM is its number of states $|Q|$.

2.5.1. Proposed Solution

For the oblivious run of an automaton in the presented scenario, both parties engage in an interactive protocol, whose number of rounds is linear in the length of the input string $\boldsymbol{x}$. In particular, the protocol is composed of three subprotocols, one for performing the first state transition, one for performing an arbitrary transition of the automaton, and one for announcing the result.

The first subprotocol performs the first state transition of the automaton, starting from its initial state, and reading the first input symbol of $\boldsymbol{x}$. The subprotocol distributes shares of the following state to both $\mathcal{A}$ and $\mathcal{B}$. Subsequently, for each further state transition, the second subprotocol is executed. Starting from the shares of the current state, it jointly calculates the transition to the next state in an oblivious way. At the end of the subprotocol, shares of the subsequent state are distributed to $\mathcal{A}$ and $\mathcal{B}$. After all state transitions have been performed (i.e., all symbols of $\boldsymbol{x}$ have been consumed), the last subprotocol is used to determine whether the computation of the automaton ended in a final state.

In the following we assume that the encryption system is set up such that $\mathcal{B}$ holds the decryption key; however, it can also be implemented with a (fair) threshold encryption scheme requiring a joint decryption step. In the latter case, the result is revealed to both parties.

Subprotocol: First State Transition

This subprotocol performs the first state transition of the automaton.

1. $\mathcal{A}$ generates a random $r_a^{(1)} \in_R \mathbb{Z}_{|Q|}$; then, he selects the column q_0 of $\boldsymbol{\Delta}$ as vector and blinds every element with $r_a^{(1)}$:
$$v_i^{(0)} = \Delta(i, q_0) + r_a^{(1)} \mod |Q|, \quad i = 0, \dots, |\Sigma| - 1.$$
2. Both parties engage in an $\mathrm{OT}_1^{|\Sigma|}$, being $\mathcal{A}$ the sender and $\mathcal{B}$ the chooser, in which $\mathcal{B}$ gets the element with index x_0 of $\boldsymbol{v}^{(0)}$. This element corresponds to
$$q^{(1)} + r_a^{(1)} \mod |Q|.$$

At the end of this subprotocol, both parties share the next state $q^{(1)}$ of the automaton.

Subprotocol: k-th State Transition

In this step, both parties use their shares of the current machine state as input, i.e., $\mathcal{A}$ holds $r_a^{(k)}$ and $\mathcal{B}$ holds $r_b^{(k)} = q^{(k)} + r_a^{(k)} \mod |Q|$.

1. $\mathcal{A}$ generates a random $r_a^{(k+1)} \in_R \mathbb{Z}_{|Q|}$, and blinds every element of the matrix $\boldsymbol{\Delta}$ with it. At the same time, $\mathcal{A}$ rotates the rows of $\boldsymbol{\Delta}$, $r_a^{(k)}$ positions to the left, obtaining the matrix $\boldsymbol{\Delta}^{(k)}$ with elements
$$\Delta^{(k)}(i, j + r_a^{(k)} \mod |Q|) = \Delta(i, j) + r_a^{(k+1)} \mod |Q|.$$
2. $\mathcal{B}$ generates a binary vector $\boldsymbol{e}^{(k)}$ of length $|Q|$, consisting in all zeros and a one at position $r_b^{(k)}$. $\mathcal{B}$ encrypts this vector $[\![\boldsymbol{e}^{(k)}]\!]$ and sends the encryptions to $\mathcal{A}$.

3. $\mathcal{A}$ performs the matrix-vector product $\boldsymbol{v}^{(k)} = \boldsymbol{\Delta}^{(k)} \cdot \boldsymbol{e}^{(k)}$ under encryption, making use of the homomorphic properties of the encryption operation, obtaining the $|\Sigma|$-length encrypted vector $[\![\boldsymbol{v}^{(k)}]\!]$. This result corresponds to an encryption of the column at position $r_b^{(k)}$ of $\boldsymbol{\Delta}^{(k)}$, or equivalently, the column at position $q^{(k)}$ of $\boldsymbol{\Delta}$, the blinded transition vector for the current state.

4. Both parties engage in an $\mathrm{OT}_1^{|\Sigma|}$, being $\mathcal{A}$ the sender and $\mathcal{B}$ the chooser, in which $\mathcal{B}$ gets the element with index x_k of $[\![\boldsymbol{v}^{(k)}]\!]$. This element corresponds to the encryption of
$$q^{(k+1)} + r_a^{(k+1)} \mod |Q|,$$
that can be recovered by $\mathcal{B}$ through decryption.

At the end of this subprotocol, both parties share the next state $q^{(k+1)}$.

Subprotocol: Announcement of Result

Once all the elements of $\boldsymbol{x}$ have been consumed by $\mathcal{A}$'s FSM, the last step determines whether the reached state is a final state. Again, the parties use shares of the reached state as private inputs, i.e., $\mathcal{A}$ holds $r_a^{(N)}$ and $\mathcal{B}$ holds $r_b^{(N)} = q^{(N)} + r_a^{(N)} \mod |Q|$.

1. $\mathcal{A}$ generates a random binary vector $\boldsymbol{f}$ as
$$f(j + r_a^{(N)} \mod |Q|) = [j \in F], \quad j = 0, \ldots, |Q| - 1,$$
whose Boolean elements encode whether a state j is a final state, having ones in the indices corresponding to acceptance states and zeros in those indices corresponding to non-acceptance states. This vector is shifted, so that the index $r_b^{(N)}$ that $\mathcal{B}$ possesses represent the position of the acceptance of the actual final state.

2. Both parties engage in an $\mathrm{OT}_1^{|Q|}$, being $\mathcal{A}$ the sender and $\mathcal{B}$ the chooser, in which $\mathcal{B}$ gets the element with index $r_b^{(N)}$ of $\boldsymbol{f}$. This element gives the binary output of the FSM.

2.5.2. Complexity Evaluation

As 1-out-of-m oblivious transfer (OT_1^m) can be implemented with linear communication complexity, the communication complexity for each subprotocol is $\mathcal{O}(|Q| + |\Sigma|)$. Furthermore, as one subprotocol needs to be performed for each symbol of the input string, the communication complexity for obliviously running a FSM on an input of length N is

$$\mathcal{O}\left(N \cdot (|Q| + |\Sigma|)\right).$$

Thus, the complexity is linear in the number of states and in the size of the input alphabet (instead of linear in their product, as for previous approaches). This implies a great improvement in complexity with respect to generic approaches for big input alphabets and high number of states.

Regarding the computational complexity, we will use the OT protocol of Naor and Pinkas [168] for implementing OT_1^m, as this protocol has an amortized complexity of $\mathcal{O}(m)$ products for the sender and $\mathcal{O}(1)$ products for the chooser. With these magnitudes for the OT subblocks, it is easy to see that the total computational complexity for $\mathcal{A}$ is $\mathcal{O}(N \cdot |\Sigma| \cdot |Q|)$, being the matrix multiplication performed at each step the most costly operation. On the other hand, the amortized complexity for $\mathcal{B}$ is just $\mathcal{O}(N \cdot |Q|)$, being the encryption of the vector that determines the shifted current state ($\boldsymbol{e}^{(k)}$) the most costly operation. It must be noted that the operations in the protocol can be *transposed*, in such a way that ($\boldsymbol{e}^{(k)}$) represent the current symbol, and the result of the matrix multiplication in step 3 of the k-th state transition subprotocol produces a vector of blinded next states for the current symbol. In this way, the roles of the dimensions $|Q|$ and $|\Sigma|$ are interchanged. Thus, the amortized complexity for $\mathcal{B}$ can be reduced to $\mathcal{O}\left(N \cdot \min(|Q|, |\Sigma|)\right)$.

2.5.3. Extension to Transducers

The basic protocol described above can be extended to transducers, while keeping the same order of complexity. This can be achieved by including some additional steps at each state transition, and omitting the last subprotocol (announcement of results).

Following the notation of Section 2.2.2.4, the modifications for implementing each type of transducers are described in the following:

- *Moore machines*: The output depends only on the current state, so $\mathcal{A}$ will have a vector $\boldsymbol{\lambda} \in \left(\mathbb{Z}_{|\Pi|}\right)^{|Q|}$, such that λ_j holds the output for the state j.

  For the initial step, the output is trivial ($\lambda(q_0)$), and it can be sent to $\mathcal{B}$. For the k-th state transition protocol, a modification must be made in its third step, in which besides the homomorphic calculation of $\boldsymbol{\Delta}^{(k)} \cdot \boldsymbol{e}^{(k)}$, $\mathcal{A}$ also rotates $\boldsymbol{\lambda}$

  $$\lambda^{(k)}(j + r_a^{(k)} \mod |Q|) = \lambda_j,$$

  computes the encryption of $\left(\boldsymbol{\lambda}^{(k)}\right)^T \cdot \boldsymbol{e}^{(k)}$, and sends the result (one encrypted scalar) to $\mathcal{B}$.

- *Mealy machines*: In this case, the output depends on the state and the input, so $\mathcal{A}$ will have a matrix $\mathbf{\Lambda} \in \mathcal{M}_{|\Sigma|\times|Q|}(\mathbb{Z}_{|\Pi|})$, such that $\Lambda(i,j)$ gives the output for input i and state j.

  In the first state transition, two $\mathrm{OT}_1^{|\Sigma|}$ are run in parallel on vectors $\boldsymbol{v}^{(0)}$ and column q_0 of $\mathbf{\Lambda}$. This gives $\mathcal{B}$ the blinded first transition and the corresponding output.

  For the k-th state transition protocol, its third step must be modified, such that besides the homomorphic calculation of $\mathbf{\Delta}^{(k)} \cdot \boldsymbol{e}^{(k)}$, $\mathcal{A}$ also rotates $\mathbf{\Lambda}$

  $$\Lambda^{(k)}(i, j + r_a^{(k)} \mod |Q|) = \Lambda(i,j),$$

  and computes the encryption of $\boldsymbol{w}^{(k)} = \mathbf{\Lambda}^{(k)} \cdot \boldsymbol{e}^{(k)}$; the following $\mathrm{OT}_1^{|\Sigma|}$ protocol is run in parallel on both vectors $\boldsymbol{v}^{(k)}$ and $\boldsymbol{w}^{(k)}$.

2.5.4. Security

Due to the asymmetry of the problem in terms of inputs to the protocol, the security definitions are slightly different from the framework commonly used for general two-party secure function evaluation. Nevertheless, the problem can be restated as two-party secure function evaluation in the following way.

Let G be a functionality that, given the description of a function as a FSM $f(.) \equiv (Q, \Sigma, \mathbf{\Delta}, q_0, F)$ and its input $\boldsymbol{x}$, gives as output $G[f(.), \boldsymbol{x}] = f(\boldsymbol{x})$. Then, the problem for the asymmetric function evaluation may be stated as a two-party computation problem in which party $\mathcal{A}$ holds the input $f(.)$, and party $\mathcal{B}$ holds the input $\boldsymbol{x}$; both parties want to evaluate G on their inputs.

We assume the semi-honest attacker model, where neither party deviates from the protocol execution, but try to infer some information about the other party's inputs. Note that the protocol allows $\mathcal{A}$ to infer N and $\mathcal{B}$ to infer $(f(\boldsymbol{x}), |Q|)$, as this information can be obtained by inspecting the output of the protocol, the length of the computation and the amount of transmitted information. The goal is to show that all the information that each party can infer from the execution of the protocol about the other party's input is no more than what they could infer from the above mentioned quantities. For $\mathcal{A}$, the framework is exactly the same as in the general two-party computation case: $\mathcal{A}$ should be unable to decide which length-N string over Σ was used by $\mathcal{B}$ as input. For $\mathcal{B}$, we can consider the protocol to be secure if $\mathcal{B}$ cannot extract from his output more information about the tuple $(\mathbf{\Delta}, q_0, F)$ than he would be able to infer from the output of the automaton when it is run as a black box. More formally, we can state the following definition, using a standard simulation argument:

Definition 2 *We say that a protocol* Π *privately evaluates* $\mathcal{A}$*'s FSM* $M = (Q, \Sigma, \boldsymbol{\Delta}, q_0, F)$ *on* $\mathcal{B}$*'s input string* $\boldsymbol{x}$ *if, given the views for both parties*

$$\begin{aligned}\mathcal{V}_A^{\Pi} &= ((|Q|, |\Sigma|, \boldsymbol{\Delta}, q_0, F), m_1, \ldots, m_t, N)\\ \mathcal{V}_B^{\Pi} &= ((|Q|, |\Sigma|, \boldsymbol{x}), m_1, \ldots, m_t, f(\boldsymbol{x})),\end{aligned}$$

where m_i *is the i-th message interchanged between both parties,* N *is the length of* $\boldsymbol{x}$*, and* $f(\boldsymbol{x})$ *is the output of the automaton* M*, there exist two polynomial time algorithms* $S_A(|Q|, |\Sigma|, \boldsymbol{\Delta}, q_0, F, N)$ *and* $S_B(|Q|, |\Sigma|, \boldsymbol{x}, f(\boldsymbol{x}))$ *producing simulated views, which are computationally indistinguishable from the respective views of the parties, i.e.,*

$$\begin{aligned}\mathcal{V}_A^{\Pi} &\equiv S_A(|Q|, |\Sigma|, \boldsymbol{\Delta}, q_0, F, N)\\ \mathcal{V}_B^{\Pi} &\equiv S_B(|Q|, |\Sigma|, \boldsymbol{x}, f(\boldsymbol{x})).\end{aligned}$$

Considering semi-honest participants, we can state the following claim:

Claim 1 *Using a semantically secure encryption scheme and a secure OT primitive, the proposed protocol privately evaluates* $\mathcal{A}$*'s FSM on* $\mathcal{B}$*'s input.*

Proof *(Sketch)* We will first sketch a simulator S_A for $\mathcal{A}$'s view of the protocol. The input to S_A is given by $(|Q|, |\Sigma|, \boldsymbol{\Delta}, q_0, F, N)$. By our assumption of OT being secure, we can assure the existence of two simulators S_c and S_s, which produce views that are indistinguishable of those of the chooser and the sender respectively. S_A composes the simulated view by first using S_s to provide one OT view in order to simulate the subprotocol corresponding to the first state transition; subsequently, it outputs N times (once for each invocation of the state transition subprotocol) encryptions of a random vector $\boldsymbol{e}^{(k)}$ of length $|Q|$ and an OT view generated by S_s. Finally, S_A uses S_s once more to simulate the subprotocol announcing the result. Note that in the protocol a new fresh random value is generated at each state transition for blinding the transition matrix and the current state, thus the inputs to each OT protocol are statistically blinded and mutually independent. Note further, that due to the semantic security of the encryption scheme, the encryptions of the random vector cannot be distinguished from encryptions of the vectors sent by $\mathcal{B}$ in the protocol. A standard hybrid argument finally shows that the simulated view, consisting of $N + 1$ views of the OT and N encrypted vectors, is computationally indistinguishable from $\mathcal{A}$'s view.

It remains to construct a simulator S_B on input $(|Q|, |\Sigma|, \boldsymbol{x}, f(\boldsymbol{x}))$ for $\mathcal{B}$'s view of the protocol, which proceeds along similar lines as S_A. The simulator uses S_c to produce $\mathcal{B}$'s view of the OT protocols and generates encrypted vectors indistinguishable from $\boldsymbol{e}^{(k)}$, again thanks to the semantic security of the encryption. Finally, for the announcement of the result, the last OT is performed on a fake vector $\boldsymbol{f}$ whose elements are all equal to the true output $f(\boldsymbol{x}_B)$. The security of the OT primitive guarantees the indistinguishability of the simulated view from $\mathcal{B}$'s view ($\mathcal{B}$ cannot decide whether there is a one or a zero in those positions different to the chosen one), while the output of the simulator is correct. Again, a hybrid argument shows that the simulated view is computationally indistinguishable from $\mathcal{B}$'s view. ∎

Furthermore, it is easy to see that the protocol is correct:

Claim 2 *The proposed protocol correctly evaluates $\mathcal{A}$'s FSM on $\mathcal{B}$'s input.*

Proof It is straightforward to see that each subprotocol is correct: if both parties follow the protocol, the transition function is correctly calculated at every step, and the output is correctly computed. By induction, the claim follows. ∎

2.6. Conclusions and Further Work

A new primitive for securely solving the N-dimensional point inclusion problem in polytopes and in hyperellyptic regions has been presented. The primitive is useful in many applications, including biometrics, classification, database queries, positioning and watermarking (cf. Chapter 4). We have analyzed the communication, round and computation complexity of the protocol and proposed input packing as a complexity reduction strategy when the number of dimensions is high.

Although the protocol is presented for the semi-honest model, a sketch for extending it to malicious parties is provided in the appendices. Extending the protocol for use with more than two parties is straightforward and requires a convexity proof when the polytope is shared among several parties.

Regarding Systems of Linear Equations, this chapter proposes new privacy-preserving protocols that make use of homomorphic computation and secret sharing. These protocols implement a direct method (Gaussian elimination), as well as iterative methods for solving SLEs (Jacobi's method) and matrix inversion (Newton's method). These protocols are secure with semi-honest parties, and, to the best of our knowledge, they are the first iterative methods under encryption proposed up to date.

There are some difficulties in the implementation of iterative methods that have been pointed out in this chapter, namely the growth of the ciphered numbers and their change in quantization scale (cipher blow-up). A more thorough analysis of this problem and some countermeasures are presented in the next chapter.

Finally, this chapter also presents an efficient protocol for the secure execution of finite automata that will find application in many fields, as shown in Chapter 4.

2.A. Complexity of BITREP

The main block of the point inclusion protocols is the BITREP primitive, that represents also the most significant contribution to their complexity. This appendix is devoted to the study of the complexity of this subblock, in order to have a complete understanding of the complexity of our protocols.

In these calculations, we have considered only semi-honest parties, but the extension to malicious parties is straightforward (Appendix 2.B). With the same notation as before, the communication complexity of BITREP can be expressed as

$$\begin{aligned}\mathrm{Cpx}_{cm,\mathtt{BITREP}} = & \frac{2^m}{n}\left(2(m+1)|E_P| + \mathrm{Cpx}_{cm,CE(\mathrm{comp})}\right)\\ & + 2|E_P| + \mathrm{Cpx}_{cm,CE(\mathrm{add})} + \mathrm{Cpx}_{cm,CE(\mathrm{sub})},\end{aligned}$$

where $\mathrm{Cpx}_{cm,CE}$ represents the communication complexity for a circuit evaluation with a comparison (comp), addition (add) or subtraction circuit (sub).

The basic operation for the circuit evaluations is the multiplication protocol. This protocol can be performed as indicated in [71], and its communication (cm), computation (cp) and precomputation (pcp) complexity in the two-party scenario is the following:

$$\begin{aligned}\mathrm{Cpx}_{\mathtt{cm},\mathtt{MULT}} &= 6|E_P|,\\ \mathrm{Cpx}_{cp,\mathtt{MULT}} &= 3\mathrm{Cpx}_{P,n^{s+1}} + \mathrm{Cpx}_D + \mathrm{Cpx}_{X,n^{s+1}},\\ \mathrm{Cpx}_{pcp,\mathtt{MULT}} &= \mathrm{Cpx}_E,\end{aligned}$$

and it has a round complexity of 3 rounds.

The complexity of the employed circuits can vary depending on the implementation, and the trade-off between round number and total complexity. Choosing constant round protocols, the complexity is increased, as redundant operations are introduced in order to parallelize the computations and achieve a round number independent of the size input. On the other hand, if only the strictly needed operations are performed, the complexity will be minimal, but the number of needed rounds will increase with the input size.

On the other hand, if the inputs are bounded—as in the application of the presented protocol—it is possible to run a *light* version of BITREP, which the authors of [199] denote LSBs gate (see Section 2.2.2.3).

As a numerical example, for a modulus of size 1024 bits and $s = 1$, considering as measure units kBytes for communication complexity and modular additions and multiplications with n-bits numbers for computation complexity, the obtained results for BITREP and LSBs gates for 943 bits numbers with constant number and non-constant number of rounds are shown in Table 2.5.

Table 2.5: Communication (cm), Computation (cp), Precomputation (pcp) and Rounds Complexity comparison between BITREP and LSBs gates using 1024 bits modulus and 943 bits numbers with constant and non-constant number of rounds.

	Constant Rounds		Non-Constant Rounds		Units
	BITREP	LSBs	BITREP	LSBs	
$\mathrm{Cpx}_{cm,\mathtt{BITREP}}$	3.546.686	1.450.497	27.223	4.712	kBytes
$\mathrm{Cpx}_{cp,A,\mathtt{BITREP}}$	11.503, 85	4.700, 19	73, 94	12, 57	10^6 ops.
$\mathrm{Cpx}_{cp,B,\mathtt{BITREP}}$	11.503, 85	4.700.19	73, 94	12, 57	10^6 ops.
$\mathrm{Cpx}_{pcp,A,\mathtt{BITREP}}$	5.750, 28	2.272, 90	41, 93	7, 72	10^6 ops.
$\mathrm{Cpx}_{pcp,B,\mathtt{BITREP}}$	5.754, 47	2.272, 90	46, 13	7, 72	10^6 ops.
$\mathrm{Rounds}_{\mathtt{BITREP}}$	269	102	9.213	1.889	rounds.

In Figure 2.5, the evolution of the complexity as a function of the bit-size of the modulus for BITREP is shown, while Figure 2.6 compares the complexity of BITREP and LSBs gates as a function of the bit-size of the encrypted number, with a fixed modulus size of 1024 bits, maintaining always a security parameter of at least 80 bits for the LSBs gate. As these figures show, the LSBs gate is much more efficient than the BITREP one, imposing only a little additional constraint on the bit-size of the encrypted number. Furthermore, the complexity for the LSBs gate are linear in the number of bits (l) of the encrypted integer:

$$\begin{aligned}
\mathrm{Cpx}_{cm,\mathtt{LSBs}} &= 20m(s+1)l - \underbrace{14m(s-1)}_{a_{cm}} \\
\mathrm{Cpx}_{cp,A,\mathtt{LSBs}} &= \frac{(s+1)^2}{2}(3s(2s+1) + 26m(s+1) + 32)l \\
&\quad - \underbrace{(s+1)^2(11 + 7m(s+1) + s(2+s))}_{a_{cp}} \\
\mathrm{Cpx}_{cp,B,\mathtt{LSBs}} &= \frac{(s+1)^2}{2}(3(12 + s(s+2)) - a_{cp} + 26m(s+1))l
\end{aligned}$$

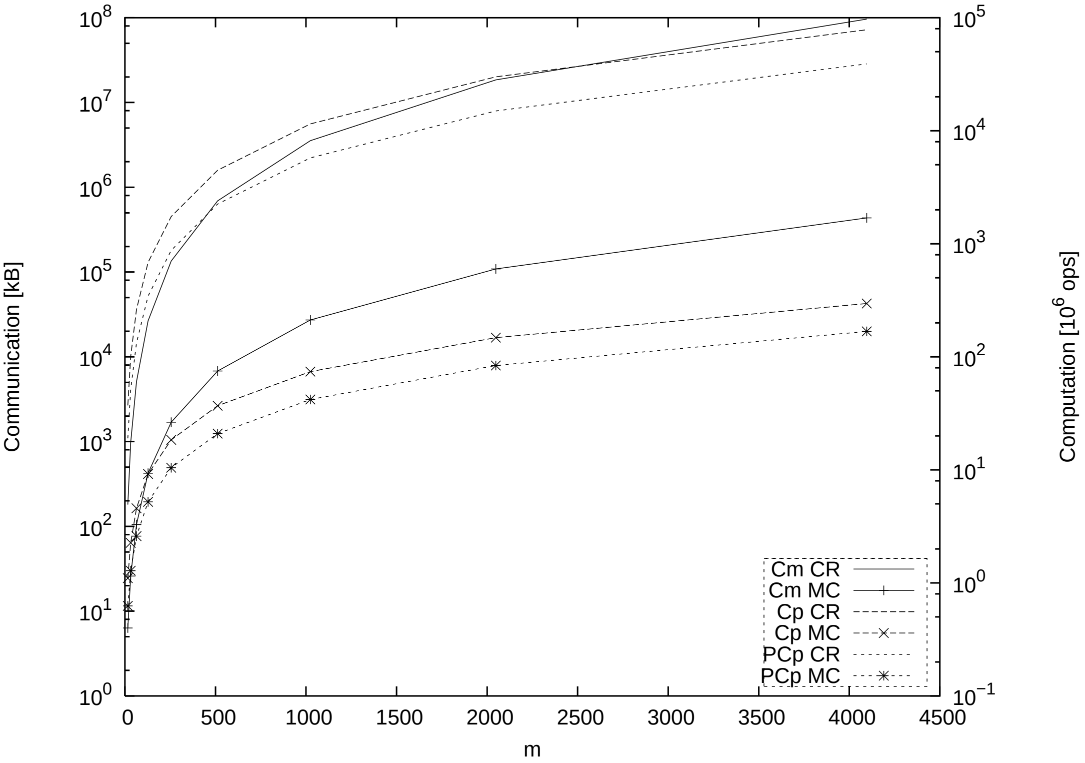


Figure 2.5: Communication (Cm), Computation (Cp) and PreComputation (PCp) complexity evolution as a function of the bit-size of the modulus for Constant Rounds (CR) and Minimum Complexity (MC) BITREP

$$\begin{aligned}\mathrm{Cpx}_{pcp,A,\mathtt{LSBs}} = \mathrm{Cpx}_{pcp,B,\mathtt{LSBs}} &= \underbrace{2(s+1)^2(2m(s+1)+1)}_{a_{cpp}}(2l-1)\\ \mathrm{Rounds}_{\mathtt{LSBs}} &= 2l+3.\end{aligned}$$

This means that the packing of numbers will affect LSBs complexity in a slightly negative way; given the additive constants a_x in the previous equations and a fixed s, the difference in complexity between running h LSBs and running one LSBs with h packed integers will be $(h-1)a_x$ (independent of the bit-size l of the integers), while the number of rounds will increase in $2l(h-1)$, as without packing the h gates can be parallelized.

On the other side, the computation complexity is $\mathcal{O}(s^4)$, so using Damgård Jurik encryptions would suppose a considerable increase in complexity, what justifies the preference for packing only using $s=1$.

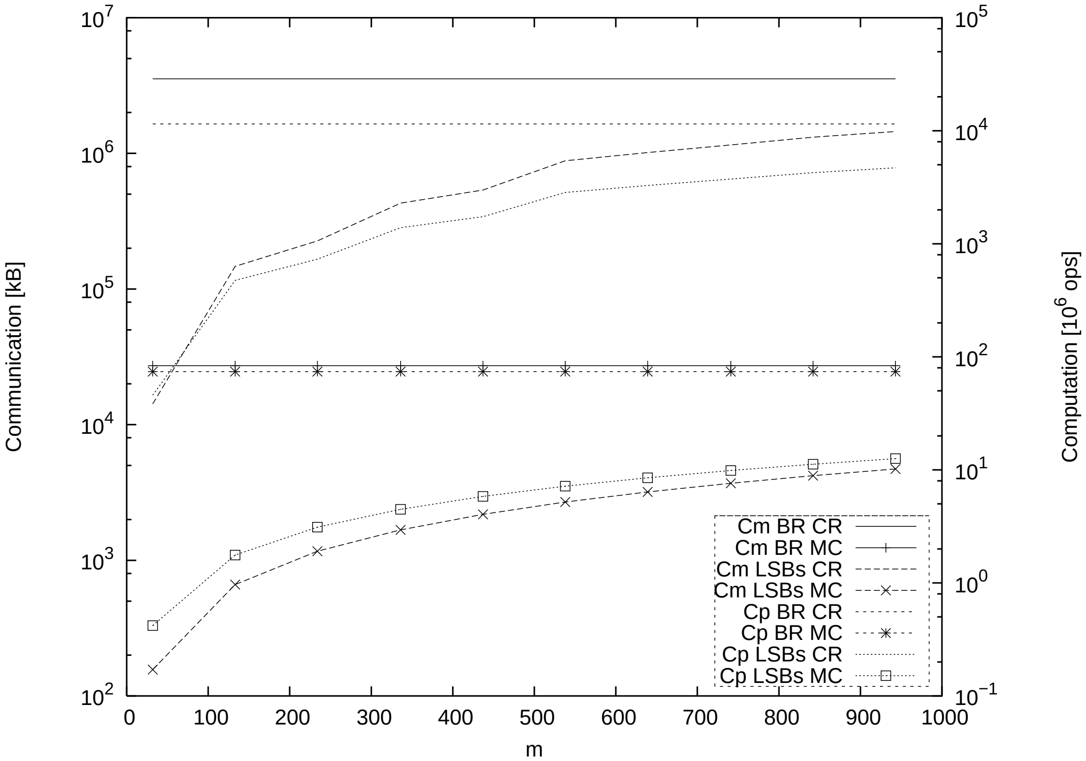


Figure 2.6: Communication (Cm) and Computation (Cp) complexity evolution as a function of the bit-size of the number, for 1024-bit modulus for Constant Rounds (CR) and Minimum Complexity (MC) BITREP and LSBs

2.B. Point Inclusion with Malicious Parties

When considering a malicious party in the point inclusion protocol (Section 2.3), it is necessary to include some Zero-Knowledge proofs in order to preserve the other party's privacy. The needed proofs are enumerated in the following, and it is indicated where they are used inside the protocol. Note that all of them can be made non-interactive substituting the verifier challenges by collision resistant hash functions.

In the following, the needed proofs for a malicious client are enumerated, and it is indicated where they are used inside the point inclusion protocol.

2.B.1. Proof of Knowledge of an Encrypted Value

This is a proof used for correct encryption, and it is based on proofs of knowledge of a discrete logarithm. Let $x \in \mathbb{Z}_{n^s}$, $r \in \mathbb{Z}_n^*$ be known by the Prover, and $c = g^x r^{n^s} \mod n^{s+1}$ be the encryption whose validity has to be proved.

1. The Prover generates $x_1 \in_R \mathbb{Z}_{n^s}$ and $r_1 \in_R \mathbb{Z}_n^*$ at random, and sends $c_1 = g^{x_1} r_1^{n^s} \mod n^{s+1}$ to the Verifier.

2. The Verifier generates a random challenge $e \in_R \{1, \ldots, 2^t\}$, and sends it to the Prover.

3. The Prover sends to the Verifier the values $x_2 = x_1 + ex \mod n^s$ and $r_2 = r_1 \cdot r^e \mod n$.

4. The Verifier checks that

$$c_1 \equiv g^{x_2} r_2^{n^s} c^{-e} \mod n^{s+1}.$$

As stated, this proof is complete, as it always succeeds for correct inputs, computationally sound, being t the security parameter, and computationally zero-knowledge, due to the computationally hiding property of Paillier encryptions. This proof must be used every time the malicious party sends to the other party a new value not homomorphically computable from the already presented encryptions.

2.B.2. Proof of Correct Multiplication

Given $n, c_a = E_{DJ}[a], c_b = E_{DJ}[b], c_f = E_{DJ}[f = a \cdot b]$, the prover knows b, but she does not know a, and wants to demonstrate that c_f corresponds to the product of the value encrypted in c_a by the one hidden in c_b. The protocol is the one described in [71, Section 8.1.2], extended to Damgård-Jurik encryptions.

This proof is needed for checking that the multiplicative blinding of D is correctly performed.

2.B.3. Range Proof

Given an encryption $c_x = g^x r^{n^s} \mod n^s + 1$, and a public interval $[a, b]$, the prover knows the values $x \in \mathbb{Z}_{n^s}$ and $r \in \mathbb{Z}_n^*$, and wants to prove that $x \in [a, b]$. This can be done using Boudot's range proof [47]. The problem with range proofs is that they are implicitly designed to work with concealed values in hidden order groups, what is not fulfilled by Paillier cryptosystem. One straightforward solution consists in generating a commitment, using for example, Damgård-Fujisaki scheme [78] for the same encrypted value and construct the range proof for this commitment, sent with a proof of equality [47] to show that the encryption and the commitment hide the same value, as indicated in [80].

This proof is used for checking the correct bounding of the inputs.

2.B.4. Non-Zero Proof

To prove that a random number $x \in \mathbb{Z}_n$, whose encryption c_x is known by the Verifier, is not zero, the Prover can generate another random number r, multiply both numbers $\mod n$, and give $c_r = E_{DJ}[r]$, $c_{xr} = E_{DJ}[xr \mod n]$ and the opening of $xr \neq 0$ to the Verifier, with ZK proofs of correct encryption and correct multiplication.

The Verifier checks the proofs and the opening of the product, and accepts if and only if all of them are correct.

The proof always succeeds if $x, r \in \mathbb{Z}_n^*$, and can only fail when x or r are zero divisors, but finding a zero divisor in $\mathbb{Z}_n$ is equivalent to factoring n (and breaking the cryptosystem), so the failure probability is negligible ($\frac{p+q}{n}$).

This proof is needed for checking the validity of the multiplicative blinding of D.

2.B.5. Square Proof

Given two encryptions $c_x = g^x r^{n^s} \mod n^s + 1$ and $c_{x^2} = g^{x^2} r_2^{n^s} \mod n^s + 1$, the prover knows the values $x \in \mathbb{Z}_{n^s}$ and $r, r_2 \in \mathbb{Z}_n^*$, and wants to prove that c_{x^2} effectively hides the squared value of the number hidden in c_x. This can be done using the proof presented in [47], that is designed to work with Fujisaki-Okamoto commitment scheme, adapting it for working with Damgård-Jurik encryptions.

This proof is employed for demonstrating the correct squaring of the input coordinates of the client when working with hyperelliptic regions.

2.C. Secure Multiplication Protocol

In order to multiply two encrypted matrices, as there is no multiplication operation in an additively homomorphic cryptosystem, it is necessary to execute an interactive protocol in order to perform each product. The generic protocol for secure multiplication gates has been known since [69]. In this work, we use a variant for non threshold encryption, that is included in this appendix for clarification and completeness. Let us assume that there exists an additively homomorphic cryptosystem with plaintext in $\mathbb{Z}_n$ such that $\mathcal{B}$ can decrypt and both $\mathcal{A}$ and $\mathcal{B}$ can encrypt. $\mathcal{A}$ owns two encrypted scalars $[\![x_1]\!]$ and $[\![x_2]\!]$ and wants to multiply them under encryption. In order to do that, $\mathcal{A}$ generates two random values $r_1, r_2 \in \mathbb{Z}_n$, and uses them to blind both numbers, using homomorphic modulo-n sum obtaining $[\![z_1]\!] = [\![x_1]\!] + r_1 \mod n$, and $[\![z_2]\!] = [\![x_2]\!] + r_2 \mod n$, and sends them to $\mathcal{B}$.

Due to his decryption capabilities, $\mathcal{B}$ can obtain z_1 and z_2 in the clear, multiply them, and reencrypt the result $[\![z_1 \cdot z_2]\!]$. $\mathcal{B}$ sends this encrypted product to $\mathcal{A}$, who, through homomorphic sums, can obtain the desired result, as

$$[\![x_1 \cdot x_2]\!] = [\![z_1 \cdot z_2]\!] - r_1 [\![x_2]\!] - r_2 [\![x_1]\!] - r_1 r_2.$$

In the scenario of a threshold homomorphic cryptosystem, the procedure is analogous, with the exception that the random values must be generated by both parties [77].

For the case of the product of an $L \times M$ matrix and an scalar, the protocol is exactly the same as the scalar-scalar case, with $L \times M$ scalar products in parallel.

For the case of matrix-matrix product, the extension is also straightforward, as all the scalar products are performed using the scalar-scalar product protocol in parallel, with only one randomization per matrix coefficient, and the remaining operations are sums, that can be performed homomorphically. Obviously, in order to minimize the computation and communication complexity, $\mathcal{A}$ may let $\mathcal{B}$ perform all the partial additions that $\mathcal{B}$ can do in the clear and $\mathcal{A}$ would need to do homomorphically.

Neglecting the complexity of the random number generation algorithms, the complexity of the whole protocol, when multiplying an $L \times M$ matrix and an $M \times N$ matrix is

$$\begin{aligned}
\mathrm{Cpx}_{cm,MULT}(L, M, N) &= M \cdot (L + N) + L \cdot N \\
\mathrm{Cpx}_{cp,MULT,A}(L, M, N) &= L \cdot N \cdot M \cdot (3\mathrm{Cpx}_{EA} + 2\mathrm{Cpx}_{EP}) \\
\mathrm{Cpx}_{cp,MULT,B}(L, M, N) &= M \cdot (L + N)\mathrm{Cpx}_D + M \cdot L \cdot N\mathrm{Cpx}_P + \\
&\quad L \cdot N \cdot ((M - 1)\mathrm{Cpx}_A + \mathrm{Cpx}_E).
\end{aligned}$$

When the previous expressions are used in this work without the parameters L, M, N it will be assumed that the product is performed between two scalars $(L = M = N = 1)$.

2.D. Novel Zero-Knowledge Subproofs

In this appendix, we present several novel zero-knowledge proofs that will be used later on in this thesis (cf. Chapter 4); they give a solution to two non-linear operations widely used in signal processing and in many secure protocols, namely rounded square root (together with a mapping needed for the square root proof to succeed), modulus and `or`.

2.D.1. Zero-Knowledge Proof that a Committed Integer is the Rounded Square Root of another Committed Integer

Adelsbach *et al.* presented in [24] a proof for a generic function approximation whose inverse can be efficiently proven, covering, for example, divisions and square roots. Here, we present a specific protocol for proving a rounded square root that follows a similar philosophy, we study its communication complexity and propose a mapping (presented in Section 2.D.3) that makes possible this zero-knowledge protocol to prove the correct calculation of square roots on committed integers (not necessarily perfect square residues):

$$PK_{\text{sqrt}}[y, r_1, r_2 : C_y = g^y h^{r_1} \mod n \wedge C_{n\sqrt{y}} = g^{n\sqrt{y}} h^{r_2} \mod n].$$

Let C_y be the commitment to the integer whose square root must be calculated. The protocol that Prover and Verifier would follow is the next:

1. First, the Prover calculates the value $x = \text{round}(\sqrt{y})$, its commitment C_x, and the commitment to its squared value C_{x^2}, and sends both commitments and C_y to the Verifier.

2. The Prover proves in zero-knowledge that C_{x^2} contains the squared value of the integer hidden in C_x, through $PK\{x, r_1, r_2 : C_x = g^x h^{r_1} \mod n, C_{x^2} = g^{x^2} h^{r_2} \mod n\}$.

3. Then, the Prover must prove that $x^2 \in [y-x, y+x]$, using a modified version of Boudot's proof [47] with hidden interval, that consists in considering also randomness in the commitments of the interval limits calculated by both parties at the first step of the proof. Using this interval instead of the one indicated in Appendix 2.D.3, the zero values are also accepted with no ambiguity when the maximum allowable value for y is below the order of the group generated by g. The counterpart is that there are two possibilities for the square root of integers of the form $k^2 + k$, with k an integer, namely k and $k+1$. The effect of this relaxation on the conditions imposed before is a small rise in the rounding error, smaller as k grows; if we take into account that the numbers that are considered integers are actually the quantization of real numbers using a step that is fixed by the precision of the system, the error is of the same order as this precision. Nevertheless, the need of working with null values without disclosing any information requires this adaptation.

4. At last, it is necessary to prove that $x \in [0, \sqrt{m}]$, if m is the order of the subgroup generated by g. If it is known—by the initialization of the commitment scheme—that $\log_2(m) = l$, then proving that $x \in [0, 2^{l/2-1}]$ is enough; if the working range for the committed integers is $[-\tau, \tau]$, with $\tau < \sqrt{m}$ (as it will be if the bit length of τ is at most $l/2 - 1$), then it suffices with the proof that x is in the working range: $x \in [0, \tau]$.

The communication complexity in bits of this protocol is

$$\mathrm{Cpx}_{cm,PK_{\mathrm{sqrt}}} = 48|F| + 9|\tau| + 18B + 53k + 6|n| + 39.$$

Claim 3 *The presented interactive proof is computationally sound and statistically zero-knowledge in the random oracle model.*

A sketch of the proof for this claim is given in the following: Completeness and soundness of this protocol are held upon the validity of the mapping of Appendix 2.D.3.

Proof *Completeness*: If both Prover and Verifier behave according to the protocol in Appendix 2.D.1, then the Verifier will accept all the subproofs and all its tests will succeed. If x is generated as the rounded square root of y, the square proof and both range proofs will be accepted because of the validity of the mapping of Appendix 2.D.3 and the completeness of these subproofs.

Soundness: Taking into account the consideration about integers of the form $k^2 + k$, the binding property of the commitment guarantees that the Prover cannot open the generated C_x and C_{x^2} to incorrect values; thus, appealing to the uniqueness property of the mapping of Appendix 2.D.3, the computational soundness of the range and squaring subproofs guarantee that a proof for a value that does not fulfill that mapping will only succeed with negligible probability.

Zero-Knowledge: We can construct a simulator S^{V^*} for the Verifier's view of the interaction. S^{V^*} must generate values C_x and C_{x^2} as commitments to random values, that will be statistically indistinguishable from the true commitments, due to the statistically hiding property of the commitment scheme. Furthermore, the statistical zero-knowledge property of the squaring and range subproofs guarantee that simulators for these proofs exist and generate the correct views, and the generation of C_x and C_{x^2} does not affect these views, due to their indistinguishability with respect to the true commitments, and that the simulators do not need knowledge of the committed values in order to succeed. ∎

2.D.2. Zero-Knowledge Proof that a Committed Integer is the Absolute Value of another Committed Integer

This proof is a zero-knowledge protocol that allows the application of the absolute value operator to a committed number, without disclosing the magnitude nor the sign of that number[1]

$$PK_{\text{abs}}[x, r_1, r_2 : C_x = g_1^x h_1^{r_1} \mod n \wedge C_{|x|} = g_2^{|x|} h_2^{r_2} \mod n].$$

Let $C_x = g_1^x h_1^{r_1} \mod n$ be the commitment to a number x, whose sign is not known by the Verifier, and $C_{|x|} = g_2^{|x|} h_2^{r_2} \mod n$ the commitment to a number which is claimed to be the absolute value of x. The scheme of the protocol is as follows:

1. Both Prover and Verifier calculate the commitment to the opposite of x, with the help of the homomorphic properties of the commitment scheme:

$$C_{-x} = C_x^{-1}.$$

2. Next, the Prover must demonstrate that the value hidden in $C_{|x|}$ corresponds to the value hidden in one of the previous commitments C_x, C_{-x}, using the ZK Proof of Knowledge described in Appendix 2.D.4.

3. At last, the Prover demonstrates that the value hidden in $C_{|x|}$ is $|x| \geq 0$, using the protocol proposed by Lipmaa [146].

The communication complexity, in bits, of this protocol is

$$\text{Cpx}_{cm, PK_{\text{abs}}} = 19|F| + 6|\tau| + 16B + 24k + 15.$$

Claim 4 *The presented interactive proof is computationally sound and statistically zero-knowledge in the random oracle model.*

[1] As in a residue group $\mathbb{Z}_q$ there is no notion of "sign", we are using the commonly known mapping:

$$\text{sign}(x) = \begin{cases} 1, & x \in \{0, \lfloor \frac{q}{2} \rfloor\} \\ -1, & x \in \{\lfloor \frac{q}{2} \rfloor + 1, n-1\} \end{cases};$$

taking into account that $-x \equiv q - x \mod q$, the mapping is consistent.

Proof *Completeness*: If both parties adhere to the protocol, then when $C_{|x|}$ hides the absolute value of the number concealed in C_x, the protocol always succeeds due to the completeness of the OR proof and the non-negativity proof.

Soundness: Due to the binding property of the commitments, the Prover cannot open C_x and $C_{|x|}$ to incorrect values. Furthermore, due to the soundness of the subproofs, if $C_{|x|}$ hides a negative number, the proof in step 3 will fail, so the complete protocol will fail (except with negligible probability); on the other hand, if $C_{|x|}$ does not hide a number with the same absolute value as the one hidden by C_x, the proof in step 2 will also fail (except with negligible probability). Thus, the whole protocol will only succeed for a non-valid input with a negligible probability given by the soundness error of the proofs in steps 2 and 3.

Zero-Knowledge: We can construct a simulator S^{V^*} such that the real interactions have a probability distribution indistinguishable from that of the outputs of the simulator. The statistical zero-knowledge property of the OR and non-negativity subproofs guarantees that simulators exist that can produce sequences that are statistically indistinguishable from these protocols' outputs, so the only quantity that the simulator S^{V^*} has to produce is C_{-x}, whose true value can be generated directly from C_x due to the homomorphic property of the used commitment scheme. Thus, the whole protocol is statistically zero-knowledge. ∎

2.D.3. Mapping for Rounded Square Root

Current cryptosystems are based in modular operations in a group of high order. Although simple operations like addition or multiplication have a direct mapping from quantized real numbers to modular arithmetic (provided that the number of elements inside the used group is big enough to avoid the effect of the modulus), when trying to cope with non-integer operations, like divisions or square roots, problems arise.

In the following, a mapping that represents quantized square roots inside integers in the range $\{1, \ldots, n-1\}$ is presented, and existence and uniqueness of the solutions for this mapping are derived. The target is to find which conditions must be satisfied by the input and the output to keep this operation secure when the arguments are concealed.

The mapping must be such that if $y \in \mathbb{Z}^+$ and $x = \sqrt{y} \in \mathbb{R}$, then ${}_n\sqrt{y} := \text{round}(x)$. For this mapping to behave like the conventional square root for positive reals, it is necessary to bound the domain where it can be applied. The formalization of the mapping would be as follows:

$$\begin{aligned} {}_n\sqrt{\cdot} : A = \{y \in \mathbb{Z}^+ | y < n\} &\rightarrow B = \{x \in \mathbb{Z}^+ | x < \text{round}\left(\sqrt{n}\right)\} \\ y &\rightarrow x =_n \sqrt{y} = \text{round}\left(\sqrt{y}\right). \end{aligned}$$

In order for this definition to be valid, and given that the elements with which this mapping works are just the representatives of the residue classes of $\mathbb{Z}_n$ in the interval $\{1, \ldots, n-1\}$, it is required that:

Lemma 1 (Existence and uniqueness of a solution) *A unique $x \in [1, x_m] \cap \mathbb{Z}^+$ exists, such that for all $y \in \{1, \ldots, \min(x_m^2 + x_m, n-1)\}$, $x_m \leq \lceil \sqrt{n} \rceil - 1$,*

$$x^2 \mod n \in [y-x, y+x)_n\,,\ x \leq y,$$

where $[,)_n$ represents the modular reduction of the given interval.

Proof *Existence:* Given $y \in \mathbb{Z}^+$, its real square root admits a unique decomposition as an integer and a decimal in this way:

$$\sqrt{y} = x + d,\ \ x = \text{round}(\sqrt{y}) \in \mathbb{Z}^+, d \in [-0.5, 0.5).$$

Squaring the previous expression, both sides of the equality must be integers, so:

$$\begin{aligned}(\sqrt{y})^2 &= x^2 + d^2 + 2dx \\ x^2 &= y - 2dx - d^2,\end{aligned}$$

and taking into account that y is integer, $2dx + d^2$ must be also an integer, and it is bounded by:

$$2dx + d^2 \in [-x + 0.25, x + 0.25) \Rightarrow 2dx + d^2 \in [-x+1, x].$$

Substituting this last equation in the previous one gives the desired result:

$$x^2 \in [y - x, y + x - 1].$$

Thus, the modular reduction of x^2 is inside the modular reduction of the interval, and x exists.

Uniqueness: Here uniqueness is concerned with modular operations, and the possibility that the interval $[y-x, y+x)$ include integers out of the initial representing range $\{0, \ldots, n-1\}$, which would result in ambiguities after applying the mod operator. In the following, all the operations are modular, and thus, the mod operator is omitted. The intervals also represent their modular reduction.

The proof is based on reductio ad absurdum. Let $y \in \{1, \ldots, x_m^2 + x_m\}$, and let $x, x' \in [1, x_m] \cap \mathbb{Z}^+$ two different integers such that both fulfill $x =_n \sqrt{y}$, $x' =_n \sqrt{y}$. This means that

$$\begin{aligned}x^2 &\in [y - x, y + x) \cap \mathbb{Z}, \\ x'^2 &\in [y - x', y + x') \cap \mathbb{Z}.\end{aligned}$$

Combining the previous relations, x and x' must be such that

$$x^2 - x'^2 \in (-x - x', x + x') \cap \mathbb{Z}.$$

Let us suppose, without loss of generality, that $x > x'$. If both x, x' are less than $x_m \leq \lceil \sqrt{n} \rceil - 1$, then their squares are below n, and follow the same behavior as if no modular operation were applied. Squares in $\mathbb{Z}$ can be represented by the following recursive formula

$$\begin{aligned} y_k =& k^2 \qquad = y_{k-1} + k + k - 1 \Rightarrow \\ y_k - y_i =& k^2 - i^2 = \begin{cases} \sum_{l=1}^{k-i-1} 2(k-l) + k + i, & k > i \\ 0 & k = i, \end{cases} \end{aligned}$$

what means that, in order for x^2 and x'^2 to be spaced less than $x + x'$ the next inequality must be satisfied:

$$\sum_{l=1}^{x'-x-1} 2(x-l) + x + x' < x + x' \Rightarrow \sum_{l=1}^{x-x'-1} 2(x-l) < 0.$$

Thus, the only solution is $x = x'$.

If, on the other hand, $x = x_m$, and taking into account that:

$$x^2 \in [y - x, y + x - 1] \Leftrightarrow y \in \left[x^2 - x + 1, x^2 + x\right],$$

there are two possibilities:

1. $y \in \{x^2 - x + 1, \ldots, n - 1\}$: if $x \neq x'$, then $x' < \text{round}(\sqrt{n})$, so the range $(x'^2 - x', x'^2 + x']$ cannot include y, and x is the only admissible solution.
2. $y \in \{1, \ldots, x^2 + x - n\}$: this is only possible if $x_m^2 + x_m > n$; in such case, given the condition imposed on x_m, then:

   $$y \leq x_m^2 + x_m - n \leq \sqrt{n}^2 - 1 + x_m - n = x_m - 1.$$

   As $x = x_m$, this means that $y < x$, which violates one of the conditions established at the beginning.

∎

One issue in the previous exposition is that it is possible that the mapping is not defined over the entire set $\{1, \ldots, n - 1\}$. Instead, if the modulus is not public, the full working range is not known, and it becomes necessary to upper bound the integers with which the system will work. In this case, the upper bound can be set to $y_m = x_m^2 + x_m$, and the mapping can be applied to the full working range; furthermore, the condition that $x \leq y$ can be eliminated, as $x \in \{1, \ldots, x_m\}$ already guarantees that there is no ambiguity.

A similar reasoning can be applied when the working range includes negative numbers:

$$\{-\lfloor\frac{n}{2}\rfloor, \ldots, 0, \ldots, \lceil\frac{n}{2}\rceil - 1\}.$$

In this case, it is enough if $x \in \{1, \ldots, \mathrm{round}(\sqrt{\frac{n}{2}})\}$, and $y \in \{1, \ldots, \lceil\frac{n}{2}\rceil - 1\}$, as x^2 covers all the range of positive numbers in which y is included, and there are no ambiguities with the $\mod$ operation, as the overlap in intervals can only be produced with negative numbers, already discarded by the previous conditions.

Limiting the working range is the biggest issue of this method; with sequential modular additions and multiplications in $\mathbb{Z}_n$, it is only needed that the result of applying the same sequence of operations (without applying the modulus) in $\mathbb{Z}$ belongs to the interval $\{1, \ldots, n-1\}$ to reach the same value with modular operations. In the case of the defined square root, it is necessary that the operations made before applying a root also return a number inside the interval $\{1, \ldots, n-1\}$, and it is not enough that the final result of all the computation is in this interval.

2.D.4. Zero-Knowledge Proof that a Commitment Hides the same Value as one of two given Commitments

This proof constitutes a mixture of a variation of the proof of equality of two commitments [47] and the technique shown in [193] to produce an OR proof through the application of secret sharing schemes.

Given three commitments $C_{x_1} = g_1^{x_1} h_1^{r_1}$, $C_{x_2} = g_2^{x_2} h_2^{r_2}$ and $C_x = g^x h^r$, the Prover states that $x = x_1$ or that $x = x_2$. The notation used for the security parameters $(B, \tau, k, F = C(k))$ is the same as in Section 1.1.1.3; the structure of the proof is the following:

1. Let us suppose that $x_i = x$, and $x_j \neq x$, with $i, j \in \{1, 2\}$, $i \neq j$. Then, for x_j, the Prover must generate the values

   $$W_{j1} = g_j^{u_j} h_j^{u_{j1}} C_{x_j}^{-e_j},$$
   $$W_{j2} = g^{u_j} h^{u_{j2}} C_x^{-e_j},$$

   such that e_j is a t-bit randomly chosen integer ($e_j \in [0, C(k))$), u_j is randomly chosen in $[0, C(k)\tau 2^k)$, and u_{j1} and u_{j2} are randomly chosen in $[0, C(k)2^{B+2k})$.

For x_i, the Prover chooses at random $y_i \in [1, C(k)\tau 2^k)$ and $r_{i3}, r_{i4} \in [0, C(k)2^{B+2k})$, and constructs

$$W_{i1} = g_i^{y_i} h_i^{r_{i3}},$$
$$W_{i2} = g^{y_i} h^{r_{i4}}.$$

Then, the Prover sends to the Verifier the values W_{11}, W_{12}, W_{21} ,W_{22}.

2. The Verifier generates a random t-bit number $s \in [0, C(k))$, and sends it to the Prover.

3. The Prover calculates the remaining challenge applying an XOR $e_i = e_j \oplus s$, and then generates the following values:

$$u_i = y_i + e_i x,$$
$$u_{i1} = r_{i3} + e_i r_i,$$
$$u_{i2} = r_{i4} + e_i r,$$

and sends to the Verifier $e_1, u_1, u_{11}, u_{12}, e_2, u_2, u_{21}, u_{22}$.

4. The Verifier checks that the challenges e_1, e_2 are consistent with his random key s ($s = e_1 \oplus e_2$), and then checks, for $k = \{1, 2\}$, the proofs

$$g_1^{u_k} h_1^{u_{k1}} C_{x_k}^{-e_k} = W_{k1},$$
$$g^{u_k} h^{u_{k2}} C_x^{-e_k} = W_{k2}.$$

The completeness of the proof follows from its definition, as if one of the x_k is equal to x, then all the subproofs will succeed.

The soundness of the protocol resides in the key s, that is generated by the Verifier. This protocol can be decomposed in two parts, each one consisting in the proof that $x = x_i$ for each x_i. Both are based in a protocol that is demonstrated to be sound [47]. So, without access to e_i at the first stage, the only way for the Prover to generate the correct values with non-negligible probability is that $x_i = x$; if $x_i \neq x$, he must generate e_i in advance for making that the proof succeeds. With this premise, one of the e_i must be fixed by the Prover, and he indirectly commits to it in the first stage of the protocol; but the other value e_j is determined by e_i and by the random choice of the Verifier s, so for the Prover it is as random as s, guaranteeing that the second proof will only succeed with negligible probability when $x_j = x$.

The protocol is witness hiding, due to the followed procedure for developing it [193]; thanks to the statistically hiding property of the commitments, all the values generated for the false proof will be indistinguishable from those of the true proof. Furthermore, the protocol is also zero-knowledge, as a simulator can be built that, given the random choices of the Verifier (s) can construct both proofs applying the same trick as for the false proof, and the distribution of the resulting commitments will be statistically indistinguishable from that of the real interactions; in fact, the original protocol was honest-verifier zero-knowledge, but adding the additional XOR on the Verifier's random choice for the true proof makes that the resulting value is completely random, at least if one of the parties is honest (it is like a fair coin flip), so the zero-knowledge property is gained in this process.

Applying the technique shown in [41], the previous protocol can be transformed in a non-interactive zero-knowledge proof of knowledge, by using a hash function H, so that $s = H(W_{11}||W_{12}||W_{21}||W_{22})$, and eliminating the transmission of $W_{11}, W_{12}, W_{21}, W_{22}$. This way, the Verifier checks that:

$$e_1 \oplus e_2 = s = H\left(g_1^{u_1} h_1^{u_{11}} C_{x_1}^{-e_1} || g^{u_1} h^{u_{12}} C_x^{-e_1} || g_2^{u_2} h_2^{u_{21}} C_{x_2}^{-e_2} || g^{u_2} h^{u_{22}} C_x^{-e_2}\right).$$

Chapter 3

Secure Adaptive Filtering

This chapter addresses the privacy problem of adaptive filtering, one of the most important and ubiquitous blocks in signal processing nowadays. It presents several use cases for adaptive signal processing, studying their privacy characteristics, constraints and requirements, that differ in several aspects from those of the already tackled linear filtering and classification problems. The chapter highlights the impossibility of using a strategy based solely on current homomorphic encryption systems, and proposes several novel secure protocols for a privacy-preserving execution of the LMS (Least Mean Squares) algorithm, combining different SPED techniques, and paying special attention to the error analysis of the finite-precision implementations. The best trade-offs are sought in terms of error, computational complexity and used bandwidth, showing a comparison among the different alternatives in these terms, and providing the experimental results of a prototype implementation of the presented protocols, as a proof of concept that showcases the viability and efficiency of the novel solutions. The obtained results and the proposed solutions are straightforwardly extensible to other adaptive filtering algorithms, providing a basis and master guidelines for their privacy-preserving implementation.

The work shown in this chapter has been partially presented at IEEE ICASSP 2011 [228], and IEEE Trans. on Information Forensics ans Security [229], and as a UVIGO Technical Report [227]; some of the technical developments have been filed as patent applications (Patent pending, Application No. 61/443823).

3.1. Introduction

After highlighting the goals of efficient privacy preservation that Signal Processing in the Encrypted Domain pursues, the previous chapters present several protocols and blocks related to generic primitives of use in many signal processing applications, describing the available tools and technologies for achieving those goals, mainly homomorphic processing, secure circuit evaluation and garbled circuits, and interactive protocols. During the (still) short lifetime of the discipline of SPED, many efficient and secure techniques have been developed for specific applications, building up a set of tools that evidence the potential of this technology.

But within this set of tools, the most efficient SPED primitives are those that exploit the properties of homomorphic encryption for performing some linear fixed operations; most of the times Signal Processing needs to go further, resorting to adaptive filtering algorithms, due to their greater flexibility, higher responsiveness when tracking the changes in the environment, their convergence to the optimal fixed solution when working in a stationary environment, and the fact that they are the optimal solution in settings where the information about the signal characteristics is not complete, offering a much better performance than fixed filters. Hence, a considerable number of practical signal processing applications make use of adaptive filters. As this chapter shows, current homomorphic cryptosystems cannot directly deal with adaptive filters due to cipher blowup after a given number of iterations; on the other hand, full homomorphisms, like Gentry's [104], able of executing any circuit without the need of decryption, are still not practical, due to the huge size needed for the ciphertexts. In fact, the existence of practical fully homomorphic cryptosystems is still an open problem. Even though there are some linear transforms and basic operations that can be directly translated into homomorphic processing, the set is too limited, and when privacy is a concern, the solution cannot impose that these operations be replaced by simpler non-adaptive algorithms, as the negative impact on performance could virtually destroy the usefulness of the algorithm. This is especially true when the involved signals are not stationary, and the filter must track their changes over time.

This chapter establishes the framework of Secure Adaptive Filtering, and directly tackles the cipher blowup problem through several secure solutions for privacy-preserving adaptive filtering that involve homomorphic processing, garbled circuits and interactive protocols, in order to overcome the limitations of the three technologies, while profiting from their respective advantages. We take the LMS algorithm as a prototypical example of a relatively simple but powerful and versatile adaptive filter, and compare the privacy solutions for the execution of the algorithm in terms of computation and communication complexity. Furthermore, we also perform a comparison in terms of the effect of fixed-point arithmetic on the error that the algorithm produces. We show the trade-off that the combination of these different technologies establishes between precision, computational load and required bandwidth, and we look for the optimum configuration by proposing novel interactive protocols aimed at efficiently solving the cipher blowup problem, coming to several solutions that reach an optimum balance among the involved performance figures.

The rest of the chapter is structured as follows: in Section 3.2, we recall the fundamental algorithms for adaptive filtering whose secure processing versions we provide. Section 3.3 presents several exemplifying adaptive filtering scenarios where privacy constraints make necessary the use of a privacy-preserving protocol, together with the trust model in use within those scenarios. In Section 3.4, some basic concepts are introduced. Section 3.5 reviews the existing solutions for SPED primitives, and their relationship with the posed problem of secure adaptive signal processing. Section 3.6 presents our solutions for privacy-preserving adaptive filtering. Section 3.7 is devoted to the evaluation of the presented protocols, in terms of bandwidth and computational complexity. A special attention is devoted to finite precision effects and error analysis in Section 3.7.2, as the private protocols work with fixed-point arithmetic. Finally, Sections 3.8 and 3.8.1 describe the practical implementation guidelines of the proposed algorithms, based on the prototypes we have built, and present the obtained results for their complexity evaluation. Section 7 gives some conclusions and anticipates future research lines following those initiated in this work.

3.2. Iterative algorithms for Adaptive Filters

As a brief introduction to the implemented methods, we present a summary of the most representative adaptive filtering family of algorithms, the Stochastic Gradient Algorithms.

Stochastic Gradient Algorithms are characterized by the use of a non-deterministic estimate of the gradient, opposed to other gradient descent methods. The Least Mean Squares (LMS) algorithm, developed by Widrow and Hoff in 1960 [240], is the most characteristic algorithm of this family, for being a simple yet powerful and widely used adaptive filtering algorithm. It comprises two processes that jointly form a feedback loop: 1) a transversal filter $\boldsymbol{w}_n$ with N_E coefficients applied to the input sequence u_n, and 2) an update process of the coefficients of the transversal filter, based on the instantaneous estimation error e_n between the output of the filter y_n and a desired response d_n. For real signals, these two processes are expressed as

$$
\begin{aligned}
y_n =& \boldsymbol{w}_n^T \boldsymbol{u}_n && (3.1)\\
\boldsymbol{w}_{n+1} =& \boldsymbol{w}_n + \mu \boldsymbol{u}_n \underbrace{(d_n - y_n)}_{e_n}, && (3.2)
\end{aligned}
$$

where μ is the step size and $.^T$ denotes transpose.

One of the variants of the LMS algorithm that does not update the filter coefficients after each output sample, but after a block of N_b samples, is known as Block LMS [63]. It has the advantage of being computationally more efficient and allowing parallel implementations, at the price of a slightly higher error excess. The update equations of this algorithm are the following

$$
\begin{aligned}
\boldsymbol{y}_n =& \boldsymbol{\chi}_n \boldsymbol{w}_n && (3.3)\\
\boldsymbol{w}_{n+1} =& \boldsymbol{w}_n + \underbrace{\frac{2\mu'}{L}}_{\mu} \boldsymbol{\phi}_n, && (3.4)
\end{aligned}
$$

where $\boldsymbol{\chi}_n$ is an $N_b \times N_E$ matrix in which the ith row is the vector $\boldsymbol{u}_{n \cdot N_b + i}^T = [u_{nN_b+i}, u_{nN_b+i-1}, \cdots, u_{nN_b+i-N_E+1}]$, and $\boldsymbol{\phi}_n = \boldsymbol{\chi}_n^T \boldsymbol{e}_n$ is the vector representing the opposite of the scaled averaged estimate of the error gradient for the N_b samples of the nth block (the scale constant is already embedded into μ). Furthermore, for the same convergence speed, the BLMS algorithm presents, in some cases, better numerical accuracy than the standard LMS. A study on the numerical accuracy for the BLMS algorithm is undertaken in Section 3.7.2.

There are many other variants of the LMS algorithm, but we will constrain our analysis and designs to only these two forms. For more complex adaptive algorithms, the difficulties of a privacy-preserving implementation are essentially those derived from the cipher blowup problem and, additionally, those derived from the implementation of nonlinear functions. The latter is a problem that does not come specifically from the adaptive filtering scenario and, thus, falls out of the scope of this chapter. Hence, the chosen forms of LMS are representative enough, as they hold the essential characteristics of adaptive filtering, and at the same time they are practical developments widely used in a vast number of applications, as those sketched in Section 3.3, in the context of a privacy-aware scenario.

3.3. Privacy Scenario and Trust Model

For all our protocols, we will consider two parties, $\mathcal{A}$ and $\mathcal{B}$, both using an additively homomorphic cryptosystem in an asymmetric scenario, where $\mathcal{B}$ can only encrypt, but $\mathcal{A}$ possesses also the decryption key, and can perform both encryption and decryption.

For the problem of private filtering, the studied scenario represents a problem of private data processing, in which one party possesses the input signal and other party possesses the reference signal or the system model for driving the filtering of the input signal.

Hence, we will assume that one party $\mathcal{B}$ has clear-text access to the to-be-filtered sequence u_n, while the other party $\mathcal{A}$ will provide the desired sequence d_n; both parties' inputs must be concealed from each other. The system parameters can be known by both parties or be provided by one party; in our case, we assume that the update step μ is agreed by both parties. The output of the algorithm (the filtered signal) is provided in encrypted form, in order to be input to a subsequent private protocol.

Regarding the privacy requirements, we will assume that both parties are semi-honest, in the sense that they will adhere to the established protocol, but they can be curious about the information they can get from the interaction. In this scenario, our protocols can be proven private (cf. Section 3.6.1); informally, both parties $\mathcal{A}$ and $\mathcal{B}$ can only get the information given by the disclosed output of the system, and no information is leaked from the intermediate steps of the protocols.

Adaptive filtering has a considerable number of applications in the field of signal processing. They can be classified in four categories, namely identification, inverse modeling, prediction and interference cancellation. Within these categories, numerous applications are subject to privacy constraints and can benefit from the primitives presented in this chapter. In the following paragraphs, as illustrative examples of the applicability of our secure protocols, we briefly introduce some of them, mainly related to *multiuser communications* where the privacy of the users must be protected from each other and, in the cases where it exists, from the central processing server. We also provide details of the application of our protocols to these scenarios.

3.3.1. Private Interference Cancellation

The scenario in which a received signal must be cleaned due to the presence of interfering signals is one of the prototypical applications of adaptive filtering, when the characteristics of the involved signals are not constant over time. It is in the setting involving multiuser channels, with the interfering signals coming from other users, where privacy is a concern and each user's signals must be concealed from other users'. Here, the filtering must be done in such a way that each receiver gets no information about the signals that it must cancel, and other users get no information about the receiver's desired signal. This is the scenario where our framework is most directly applicable.

In the secure scenario that we are depicting, the signal that reaches the receiver is digitally sampled and encrypted before any processing, in order to protect the privacy of the involved parties. The decoding stage can be implemented using a privacy-preserving protocol that replicates the steps of the digital decoder in fixed point arithmetic, using the rounding protocols presented in the following sections when needed. The description of the secure decoding protocol is out of the scope of this chapter, but it must be noted that a demodulator (with linear filters) followed the linear stages of a decoder can be directly implemented just with homomorphic processing, in such a way that the party that runs the decoder can obtain an encrypted *soft* decoding output without any interaction of the signal owner, and then run an interactive private decisor to obtain the encrypted decoded symbols. From now on, we will represent the composition of these protocols that take the encrypted input digital signal and output an encrypted decoded symbol sequence as a block called *private decoder*.

There are several possibilities when using an adaptive filter like the LMS for interference cancellation, that are described in the following paragraphs:

- One particular case is to use an *adaptive line enhancer* (ALE), in which the reference signal d_n is the input $u_{n-\Delta}$ delayed an interval of Δ samples, called prediction depth or decorrelation delay. This setup is used for detecting a sinusoidal signal buried in a wideband noise background [248], and it is one of the simplest configurations of the LMS filter; the ALE does not directly support the use of our protocols as they are presented, because all the input signals are in possession of one of the parties.

- More elaborated designs of adaptive cancellers make use of information about the interfering signals; one example is joint decoding, in which the interferer provides a training sequence and/or the timing of the interfering signals; this (private) information can be used to perform a first decoding stage (with a private decoder block) that extracts the interfering signal, that can be used later as the reference signal within the private LMS protocol, in order to extract the cleaned signal.

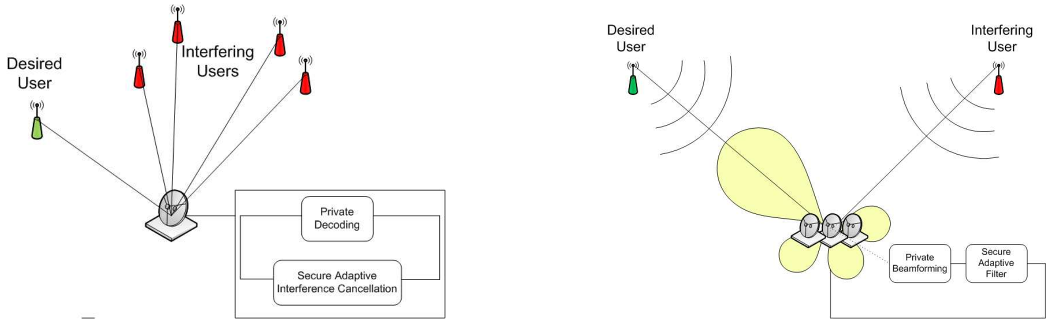
The second scheme is shown in Figure 3.1a, where the private decoder extracts the needed information about the interfering signals; this information constitutes the input to the private filtering block that provides the adaptive weights applied to the received signals in order to clean the desired one. The contents of the interfering private signal are not disclosed.

As a specific example, we can devise a scenario of a multiple access channel within an ad-hoc network, with several users transmitting simultaneously and asynchronously private signals; a decentralized multiuser detection (MUD) algorithm (cf. [234, 249, 169]) is used for decoding the signals addressed to each of the K receiving users (each one with his own decryption keys and reference signals $d_{n,i}$ or signatures), through a channel estimation step and an interference subtraction step. The signatures $\{d_{n,i}\}_{i=1,...K}$ and the sent messages $\{y_{n,i}\}_{i=1,...K}$ must be kept private to each of the corresponding receivers. The detection algorithm is run collaboratively between pairs of receivers to clean their desired signals, in such a way that each user will not have access to the signals that are not addressed to them.

3.3.2. Private Adaptive Beamforming

Adaptive beamforming is a spatial application of adaptive filtering where a system composed of an array of antennas changes the directionality of the transmitted/received signal without mechanically moving the antennas. In the most common setting, the system must determine the spatial direction of the interfering signal and/or that of the target signal, and filter the sensed signals in order to cancel the former and extract the latter; it finds use in communications, radar, sonar or speech enhancement. The interfering signal comes usually from another source. The trust model in this scenario deals with, on the one hand, the protection of the transmitted/received target signal, and, on the other hand, the protection of the interfering signal and the spatial position of the interfering source. The two parties involved in the scenario are represented in the beamformer by the adaptive filtering mechanism that cleans the desired signal, and the model and pilot information for the desired signal. Again, this model fits perfectly in our framework, and the protocols that we present can be straightforwardly adapted to this scenario. The private filtering block (Figure 3.1b) provides the adaptive weights applied to the received signals in order to adjust the directivity of the antenna array, without disclosing the contents of the interfering private signal; as in the private interference cancellation scenario, it must be complemented by another private block, denoted private beamforming block, that processes the mixed signals while concealing the private information.

Descending to a lower abstraction level, there are several possibilities for implementing an adaptive beamformer, but the most practical ones are those that, once known or calculated the directions of arrival of the desired signal and the interfering ones, make use of the LMS algorithm in order to suppress the interfering signals: u_n would be the input received from the interfering directions (θ_i) and d_n, the input coming from the direction of the desired source θ_d; the LMS filter minimizes the power of $e_n = d_n - y_n$, that is taken as the output of the beamformer. Hence, the LMS minimizes the influence of the interfering signals. Using the same terminology as for *generalized sidelobe cancellers* (GSC), the sequences coming from the antenna array must be filtered by a filter bank (represented



(a) (b)

Figure 3.1: Private Interference Cancellation (a) and Private Adaptive Beamforming (b) Scenarios.

by a *signal-blocking matrix* $\boldsymbol{C}_a$) that takes only the subspace of signals coming from θ_i, and another filter (represented by the *quiescent-weight* vector $\boldsymbol{w}_q$) that takes the signal coming from θ_d. The LMS algorithm gives the weights of the *adjustable-weight vector* $\boldsymbol{w}_a$ that filters the output of the signal-blocking bank. Informally, the LMS algorithm drives the adjustable-weight vector for obtaining the part of the interfering signals that is still present in d_n. The filters $\boldsymbol{C}_a$ and $\boldsymbol{w}_q$ must be applied privately (being linear filters, homomorphic computation is enough for their implementation) to the input sequence, and they constitute the *private beamforming block* shown in Figure 3.1b; there are two possibilities for the calculation of these filters:

- When the direction of arrival (DOA) is provided by the sources, $\boldsymbol{C}_a$ and $\boldsymbol{w}_q$ are fixed.

- When the direction of arrival of each signal is not known, a privacy-preserving implementation of a DOA algorithm (MUSIC – MUltiple SIgnal Classification, or ESPRIT – Estimation of Signal Parameters via Rotational Invariance Technique) must be used for calculating these directions and the corresponding $\boldsymbol{C}_a$ and $\boldsymbol{w}_q$.

As a specific example of this scenario, we could pose the problem of a cellular smart antenna, property of a mobile operator receiving signals (mixed into a signal $\boldsymbol{u}_n$) from his own users and also from users of a second operator that subcontracts the infrastructure of the former. The latter operator (party $\mathcal{A}$) has decryption capabilities (and reference signals $d_{n,i}$ for each of his users) and wants to perform adaptive beamforming to clean the signals $y_{n,i}$ from the clients without disclosing to the former (party $\mathcal{B}$) their positions or the contents of the cleaned signals, in such a way that the information of the users of $\mathcal{B}$ is also not disclosed to $\mathcal{A}$.

3.3.3. Private Model-Reference Adaptive Control (P-MRAC)

There are many control applications [207] where the parameters of the controlled system are either not fully known or vary over time. Adaptive control yields a solution for maintaining consistent performance in these cases. It is used in many industrial contexts like, to name a few, robot manipulation, ship steering, aircraft control or metallurgical/chemical process control. Model-Reference Adaptive Control (MRAC) is one approach for constructing adaptive controllers. An MRAC system is composed of four elements:

- A *plant* with a known structure but unknown parameters.
- A *reference model* that specifies the desired output of the control system to the external command. It should match the performance specification while being achievable by the control system.
- A feedback control law (*controller*) with adjustable parameters. It should guarantee tracking convergence and stability.
- An *adaptation mechanism* for updating the adjustable parameters.

The trust model in this scenario can be devised as a two party model (involving privacy of system users at the plant and at the controller), where the plant outputs must be kept secret from the party that runs the controller, and the reference model that the controller applies must also be kept secret for the parties in the plant. In order to adaptively control the plant while keeping the privacy constraints, the same philosophy that we apply to LMS can be used to straightforwardly translate the protocols that we present for their use in this scenario.

The MRAC setting corresponds to a system identification problem where d_n is the signal coming from the reference model and u_n is the signal coming from the plant; the LMS algorithm provides the corresponding adaptation mechanism, yielding the coefficients that must be applied to the plant in order to conform to the reference model; finally, the feedback control law is given by the minimization of the MSE between the output of the plant and that of the reference model. The LMS algorithm might be a very simple control mechanism in this case, but it captures the essence of the adaptive control problem, and more complex adaptation mechanisms can be directly obtained by appropriately pre-processing the signals coming from the plant and the reference model, and post-processing the filter coefficients before they are inputted to the plant. The implementation of these pre- and post-processing modules must also preserve the privacy constraints to have a globally secure protocol.

As a specific example for this scenario, we could devise a spacecraft control system working with classified information coming from a vehicle in orbit, using an antenna under the control of a non-trusted party; the control information cannot be disclosed for keeping secret the management of the vehicle behavior. In this case, the party that emits the control (reference d_n) signal has decryption capabilities, while the non-trusted party that receives the vehicle's signals (u_n) can only encrypt.

Current privacy-preserving solutions cannot be directly applied to these scenarios due to the cipher blowup problem, that prevents the use of homomorphic computation alone. We will present in the subsequent sections our novel solutions to that problem; they have a direct application in the aforementioned scenarios and present efficient private protocols that overcome cipher blowup, finding an optimal trade-off between precision and complexity.

3.4. Secure Computation and Garbled Circuits

In this chapter we make use of and evaluate several already introduced concepts:

- Homomorphic encryption (cf. Sections 1.1.1.2 and 2.2.2.1): as before, we do not restrict the used cryptosystem for the presented protocols, as long as it presents an additive homomorphism; additionally, for evaluation purposes, we employ the Damgård and Jurik's cryptosystem (cf. Section 2.2.2.1), for which we propose several efficiency improvements, shown in the Appendices.
- Secret sharing (cf. Section 2.2.2.2).
- General Secure Multiparty Computation (cf. Section 1.1.1.1) and interactive protocols.

Besides these tools, we also make use of garbled circuits, that were not explored in depth in the previous chapters.

Garbled circuits are based on the solution that Yao initially proposed for the Millionaire's problem; it consists in two parties evaluating a given circuit, gate by gate, without knowing the output of each gate. Yao's solution was not efficient, and later, many protocols based on other principles like homomorphic computation or secret sharing were proposed in order to efficiently perform other operations in a secure manner.

Nevertheless, while homomorphic computation and secret sharing are very efficient for implementing arithmetic operations, circuit evaluation is still more efficient when dealing with binary tests [77]. Thus, there exist efficient protocols for binary comparison [77, 173] or Prefix-OR [77] that will be needed, with some modifications, for the implementation of our solutions. Traditionally, the search for efficient solutions has led to proposals for changing between integer and binary representations in order to efficiently implement both arithmetic and binary operations; e.g., there are solutions like the BITREP protocol [199], that converts Paillier encrypted integers to Paillier encryptions of their corresponding bit representation (cf. Chapter 2).

For the garbled circuit constructions in this chapter, we use the efficient protocols developed in [141], and for the transformation from Paillier representation to a binary representation suitable for usage in a garbled circuit, we employ the protocols in [140].

3.5. Related Art

Previous work on private linear filtering has been presented as part of the SPEED project [9], dealing with the privacy problem in a two-party setting where one party has an input to a linear filter and another party holds the filter coefficients. Such efficient privacy-preserving solutions are based solely on homomorphic processing, as it fits perfectly the linear filtering operation without imposing any overhead on communication. Within the area of linear filtering, we can point out the works by Bianchi *et al.* [44, 43, 214], dealing with encrypted DFT and DCT transforms and frequency-domain linear filtering. Additionally, these works discuss also the problem of disclosing data derived from the inputs without any dimensionality reduction, as the original data can be inferred from the disclosed outputs.

There have been also some contributions for more complex operations, involving the combination of garbled circuits and homomorphic processing, most notably those from Kolesnikov *et al.* [140], in which homomorphic processing is used for the linear operations, while garbled circuits deal with non-linear operations.

Regarding the privacy considerations in iterative algorithms, there are some contributions in the area of private collaborative filtering, like those by Canny [53] and Erkin [90]. In the former, Canny developed a privacy-preserving iterative conjugate gradient algorithm for the calculation of the SVD decomposition of a shared preference matrix $\boldsymbol{P}$. The setting in [53] is different from ours in several aspects: a) It involves multiple parties, and the gradient estimate in each iteration is calculated as the sum of the contributions from each of these parties; b) the result of every iteration is decrypted and disclosed before the next iteration; hence, it does not involve successive products of encrypted values, as each party uses only clear-text values for producing the results at every iteration; c) as a drawback, the disclosure of the approximation of the preference matrix and the global gradient calculated at each iteration are publicly known; hence, the security relies on those matrices having a very high dimension and the system having a very high number of users. In this chapter, we are dealing with protecting the signals coming from one party during their adaptive filtering by another untrusted party; in this setting, Canny's solution loses its privacy properties, as the value disclosed after each iteration allows each party to calculate the secret input from the other party. Furthermore, we must keep all the intermediate values encrypted in order to effectively preserve the privacy of the involved users, and this involves repeated products of encrypted numbers that will have direct consequences on the viability of the used privacy-preserving techniques due to the cipher blowup problem.

Other private iterative algorithms involve K-means clustering of a database shared between two parties, like the one proposed by Jagannathan *et al.* [131]; again, in this setting, the results of each iteration (the current classification of the elements) are disclosed before the next, and the security relies on the dimensionality of the databases, unlike the case of private adaptive filtering.

Hence, to the best of our knowledge, there are no specific solutions within the emerging field of Signal Processing in the Encrypted Domain for securely executing iterative or adaptive algorithms besides [220] (cf. Chapter 2), nor any study performed on the impact that an iterative implementation has on the range of representable numbers when the results of each iteration cannot be disclosed. Hence, our solutions are presented here as the first ones dealing with privacy preserving adaptive filtering algorithms.

3.6. Proposed Protocols

In this section, we present different approaches in order to tackle the private implementation of the LMS algorithm, and to overcome the limitations that the sole application of current homomorphic encryption and garbled circuits has in our scenario.

3.6.1. Homomorphic processing

The LMS algorithm, and most of the adaptive filters currently in use, while having an essentially non-linear behavior due to their adaptive nature, comprise only linear operations. Thus, it is foreseeable that homomorphic processing can yield a quite efficient solution. Unfortunately, there are two drawbacks in following this approach:

- There are no practical fully homomorphic cryptosystems (cf. Chapter 6); the most promising contribution in this sense is Gentry's poly-time and poly-space fully homomorphic cryptosystem, whose constant factors make it impractical [104]; hence, using only additive homomorphic processing implies resorting to interactive protocols for performing multiplications between encrypted values, or for any other more complex operation.
- The inputs to the secure protocol must be quantized prior to encryption. Hence, it is necessary to work in fixed point arithmetic, keeping a scale factor that affects all the values under encryption. This factor will increase with each encrypted multiplication, limiting the number of allowed iterations of the adaptive algorithm, until the encrypted numbers cannot fit the cipher, when it is said that the cipher blows up.

There are two approaches for devising a private LMS protocol, depending on whether the output is either disclosed or given in encrypted form. The simplest approach is the one in which the output of the LMS algorithm can be disclosed to both parties; in this case, a secure protocol could be quite efficient, as the problem of the increased scale factor can be easily solved by requantizing the outputs in the clear after every iteration with no additional overhead, requiring only homomorphic additions and multiplications and interactive multiplication gates. Nevertheless, besides its simplicity, this scenario is of no interest due to the fact that disclosing the output gives both parties all the necessary information for retrieving the other party's private input and rendering the privacy-preserving solution unnecessary and unusable.

The private output scenario is more realistic, and it is the one on which we will focus, as it corresponds to the case where the LMS block can be used as a module of a more complex system whose intermediate signals must not be disclosed to any party. We will adhere from now on to this scenario, and we will begin by presenting a protocol that uses only homomorphic computations (Algorithm 1), in order to have a complexity reference and show its limitations. In Algorithm 1, interactive multiplication protocols are avoided due to the division of the roles of both parties: the party that provides the private input $\boldsymbol{u}$, without decryption capabilities, is the one that will perform the homomorphic operations between the encrypted intermediate values and $\boldsymbol{u}$. In this case, there is a constant scaling factor (`updateFactor`) that is accumulated after every iteration, and that forces to scale the inputs and the intermediate results in order to have the correct output. This accumulated factor limits the maximum number of iterations that the protocol will be able to execute before the cipher blows up:

$$N_{\text{max iter}} = \left\lfloor \frac{n_{\text{cipher}}}{n_x + \log_2(\texttt{updateFactor})} \right\rfloor, \tag{3.5}$$

where n_x bits are used for representing each input, and n_{cipher} is the bit size of the maximum representable number inside the cipher.

The communication complexity of this protocol, assuming Damgård-Jurik encryptions, is

$$\mathrm{Cpx}_{cm,HP} = (2N_{\mathrm{iter}} + N_E - 1)|E|, \tag{3.6}$$

where N_{iter} is the number of performed iterations, N_E is the length of the filter and $|E|$ represents the number of bits of an encryption.

It is important to note that the iteration limit imposed by this protocol, due to cipher blowup, is a serious drawback and impedes the use of only homomorphic processing (in its current development stage) to perform adaptive filtering. For typical values of the used precision (48-bit numbers, 24 bits for the fractional part) and medium-term security (2048 bits for Paillier modulus), this protocol is limited to approximately 17 iterations, what is insufficient even for reaching the steady-state regime, and prevents its use in any practical application. Therefore, we present it as a reference that sets the minimum of computation and communication complexity that can be achieved for a private LMS. It must be noted that this iteration limit could be improved through the use of a different encoding of the inputs, like the logarithmic encoding presented in [95], but such approach comes at the price of an increased communication and computation complexity even for additions and multiplications.

In the following subsections, we propose several novel alternatives and extensions, through the combination of other privacy-preserving techniques, aimed at overcoming the cipher blowup problem with the minimum overhead in communication and computation complexity, while preserving an acceptable excess error with respect to the infinite precision non-private LMS algorithm.

Algorithm 1 Homomorphic Processing (HP) PrivateLMS Protocol

Inputs: $\mathcal{A}$: $d_n, \boldsymbol{w}_0$; $\mathcal{B}$: $u_n, \boldsymbol{w}_0$
Outputs: $[\![y_n]\!]$.

$\mathcal{A}$	$\mathcal{B}$
Initialize carriedFactor$= 2^{n_f}$, updateFactor$= 2^{3n_f}$.	
Encrypt inputs and send $[\![d_n]\!]$ to $\mathcal{B}$.	
for $k = 1$ **to** N_{iter}	
	Perform the vector multiplication $[\![y_k]\!] = [\![\boldsymbol{w}_k]\!] \cdot \boldsymbol{u}_k$. Scale $[\![d'_k]\!] = [\![d_k]\!]\cdot$carriedFactor. Obtain $[\![e'_k]\!] = \mu \cdot ([\![d'_k]\!] - [\![y_k]\!])$. Perform the scalar multiplication $[\![\boldsymbol{\Delta w}_k]\!] = [\![e'_k]\!] \cdot \boldsymbol{u}_k$. Update the coefficients vector $[\![\boldsymbol{w}_{k+1}]\!] = [\![\boldsymbol{w}_k]\!]\cdot$updateFactor$+[\![\boldsymbol{\Delta w}_k]\!]$.
Update carriedFactor$=$carriedFactor$\cdot$updateFactor.	
	Output $[\![y_k]\!]$.
endfor	

Security

Regarding the security of this protocol and the ones presented in the following sections, it can be proven, using a simulator argument, that all of them are statistically secure under the random oracle model, assuming semi-honest parties: due to the use of sequentially composed secure subblocks and the semantic security of the underlying cryptosystems, the views that each party gets are statistically indistinguishable from a random sequence, and the parties cannot derive from those views any extra information about the private inputs of the other party. We will not go into details about these proofs, as they are rather straightforward.

3.6.2. Garbled Circuits Implementation

This protocol represents the whole LMS algorithm as a binary circuit, in which we include a rounding operation in each multiplication circuit in order to preserve a constant bit-size for the handled numbers. The protocol is sketched as Algorithm 2. It is straightforward to derive the binary circuit implementing Eqs. (3.1) and (3.2), so we do not detail its construction in Algorithm 2; as for the garbled implementation, we use the XOR-free version of [141], with the efficient extensions for the Oblivious Transfer (OT) protocol of [129], and an Elliptic Curve version of ElGamal [139, 147] for the encryptions. This implementation uses fixed precision, and rounds the numbers after every multiplication in order to preserve this precision. Hence, it overcomes the quantization problems that the previous one presents, but it requires working at a bit level, thus being its performance highly dependent on the bit-size of the represented numbers.

Additionally, every transferred bit must be independently encrypted, which also multiplies the communication complexity of the whole protocol by a large factor, resulting in

$$\begin{aligned}\mathrm{Cpx}_{cm,GC} =&|E|\left(4n_x^2(N_{\mathrm{iter}}+2N_EN_{\mathrm{iter}})+2n_x(-1+10N_{\mathrm{iter}}+4N_{\mathrm{iter}}n_f+2N_E(1+5N_{\mathrm{iter}}+4N_{\mathrm{iter}}n_f))\right.\\&\left.-4N_{\mathrm{iter}}(5+n_f(3+n_f)+N_E(7+2n_f(3+n_f)))\right), \qquad (3.7)\end{aligned}$$

where N_E is the length of the filter, $|E|$ represents the number of bits of an EC-ElGamal encryption, n_x is the total number of bits for representing each number, and n_f is the number of bits used for the fractional part.

The complexity has, as expected, a linear dependence on the product of the number of iterations and the size of the filter, while it has a quadratic dependence on the bit-size of the used numbers and the bit-size of the fractional part, due to the presence of multiplication circuits. The communication complexity is much higher than in Algorithm 1, due to the need of communicating the whole garbled circuit prior to its execution.

A remark worth noting on Algorithm 2 is that inputs get to the circuit once per iteration, even when they could be joined all together (in long enough blocks) and apply OT reduction techniques [129] for lowering the computational complexity of the whole protocol. The reason behind this structure is that we are assuming that the system must work with some real time constraints, and offer the outputs at the same rate as the input, without a significant delay. Hence, the inputs might be packed together for reducing the computation overhead of the OTs in small blocks, whenever the delay is affordable; it must be noted that the communication overhead is not reduced though: the reduction techniques in [129] replace public key encryptions with computationally lighter hash functions; since we are using elliptic curves for the public key encryption, their size is comparable to that of a collision resistant hash for the same security level. The effect of the OT reductions is shown for the hybrid block protocol in Section 3.8.1.

Algorithm 2 Garbled Circuit (GC) PrivateLMS Protocol

Inputs: $\mathcal{A}$: $d_n, \boldsymbol{w}_0$; $\mathcal{B}$: $u_n, \boldsymbol{w}_0$.
Outputs: $[\![y_n]\!]_b$.

$\mathcal{A}$	$\mathcal{B}$
Obtain the bit representation of their respective inputs.	
Execute `generateGC()` for the first $m \leq N_{\text{iter}}$ iterations, and send the garbled circuit and the keys corresponding to her inputs to $\mathcal{B}$; the garbled gates for the remaining iterations can be generated and sent in parallel with the execution of the previous ones.	
for $k = 1$ **to** N_{iter}	
Perform parallel *OT* protocols so that $\mathcal{B}$ get the input keys to initialize the circuit corresponding to the k^{th} iteration.	
	Execute the circuit, using the received input keys from $\mathcal{A}$. Output $[\![y_k]\!]_b$.
endfor	

3.6.3. Hybrid Implementation

In order to overcome the quantization problem in Algorithm 1 and lower the communication complexity of Algorithm 2, we have developed a hybrid algorithm (Algorithm 3) that uses homomorphic processing for the bulk of the algorithm, and a quantization circuit to avoid carrying factors. Conversion protocols from homomorphic encryption to binary representation and vice-versa are used to connect both parts of the protocol.

There are several possible combinations of homomorphic processing and garbled circuits that yield different results in the complexity balance. We can argue that the optimal point for applying quantization in terms of efficiency is at every iteration, when the scaled output of the filter y'_k is obtained (cf. Algorithm 3), using a quantization step of 2^{3n_f} to recover the initial precision of n_f fractional bits. When this strategy is chosen, only one scalar value is input to the quantization circuit at every iteration, which means reaching the minimum of communication complexity for the used garbled circuit. Furthermore, this quantization allows to keep a constant scaling factor for the rest of the handled values, avoiding the rescaling operation that is performed in Algorithm 1 for every input value and for the filter coefficients; hence, the computation complexity also reaches its minimum with this strategy. Lastly, the bounded size of the represented values makes possible the use of a packing strategy for the homomorphic processing, such that more than one input value can be packed into the same encryption. This will be further commented in Section 3.6.4.

The communication complexity of the protocol is

$$\mathrm{Cpx}_{cm,Hy} = (2N_{\mathrm{iter}} + N_E - 1)|E_H| + N_{\mathrm{iter}}|E_C|(19n_x + 7n_{\mathrm{sec}} + 24n_f), \tag{3.8}$$

where N_E is the length of the filter, $|E_H|$ and $|E_C|$ represent the bit-size of a homomorphic and a garbled encryption respectively, n_x is the total number of bits for representing each number, n_f is the number of bits used for the fractional part, and n_{sec} is the number of security bits used for the conversion protocols. As the circuit part involves only rounding operations, and the multiplications are performed homomorphically, the complexity is linear on the bit-length of the inputs and the number of iterations, instead of quadratic, as in the garbled-circuit solution.

In this case, the quantization step used for the filter coefficients is not the same as the one used for the input/output values: filter coefficients are quantized with a finer step, using $3 \cdot n_f$ bits for their fractional part, instead of n_f. This is needed in order to keep the bit-size of the outputs constant and avoid any further quantization operations. Furthermore, as stated in Section 3.7.2, the quantization step of the filter coefficients is the one that has the highest impact on the quantization error that is propagated to the outputs, so this measure will make this method have a much better behavior than Algorithm 2 in terms of mean square error (MSE).

3.6.4. Hybrid Block Protocol and Packing Strategy

As pointed out in the preceding section, the hybrid implementation of the algorithm has the advantage of working always with bounded numbers, and it allows for a parallel block implementation in the form of packed coefficients within a cipher, as introduced in [230].

Typically, the numbers involved in signal processing calculations can be bounded, and their bit-size represents just a very small fraction of the size of a secure cipher modulus; the extra bit size is unused, but it is necessary due to security constraints on the cryptosystem. Nevertheless, this space can be utilized; assuming that every involved calculation result x is bounded at the moment of unpacking such that it occupies at most n_b bits (i.e., $|x| \leq 2^{n_b-1}$; for the hybrid protocol, $n_b = n_x + 3 \cdot n_f$), every K inputs $\{x_i\}_{i=0}^{K-1}$, with $K \leq \lfloor \frac{n_{\text{cipher}} - n_{\text{sec}}}{n_b} \rfloor$ (being n_{sec} the number of security bits needed for the conversion protocol), can be packed in only one encryption as $[\![\boldsymbol{x}_{\text{packed}}]\!] = [\![\sum_{m=0}^{K-1}(x_m + 2^{n_b-1}) \cdot 2^{m \cdot n_b}]\!]$, being 2^{n_b-1} a shift factor for considering only positive numbers[1]. This packing allows for the computation of vector products and additions with a reduced complexity (it gets divided by the number of packed elements), taking advantage of the unused huge space that the cipher allows.

Algorithm 3 Hybrid (Hy) PrivateLMS Protocol

Inputs: $\mathcal{A}$: $d_n, \boldsymbol{w}_0$; $\mathcal{B}$: $u_n, \boldsymbol{w}_0$.
Outputs: $[\![y_n]\!]$.

<table>
<tr><th>$\mathcal{A}$</th><th>$\mathcal{B}$</th></tr>
<tr><td>Encrypt her inputs.
Execute generateGC() for the rescaling circuit in each of the first $m \leq N_{\text{iter}}$ iterations, and send the garbled circuit to $\mathcal{B}$; the remaining circuits can be generated and sent during the execution of the previous ones.</td><td></td></tr>
<tr><td colspan="2"><b>for</b> $k = 1$ <b>to</b> N_{iter}</td></tr>
<tr><td></td><td>Perform the vector multiplication $[\![y'_k]\!] = [\![\boldsymbol{w}_k]\!] \cdot \boldsymbol{u}_k$.</td></tr>
<tr><td colspan="2">Convert $[\![y'_k]\!]$ to its bit-representation using the bit conversion protocol.
Perform parallel OT protocols so that $\mathcal{B}$ get the input keys to initialize the circuit corresponding to the k^{th} iteration.</td></tr>
<tr><td></td><td>Execute the rescaling circuit $[\![y_k]\!]_b = \left[\!\!\left[\left\lceil \frac{y'_k}{2^{3 \cdot n_f}} \right\rceil\right]\!\!\right]_b$, using the received input keys from $\mathcal{A}$.</td></tr>
<tr><td colspan="2">The shared output of the circuit $[\![y_k]\!]_b$ is converted back to a homomorphic encryption $[\![y_k]\!]$.</td></tr>
<tr><td></td><td>Obtain $[\![e'_k]\!] = \mu \cdot ([\![d_k]\!] - [\![y_k]\!])$.
Perform the scalar multiplication $[\![\boldsymbol{\Delta w}_k]\!] = [\![e'_k]\!] \cdot \boldsymbol{u}_k$.
Update the coefficients vector $[\![\boldsymbol{w}_{k+1}]\!] = [\![\boldsymbol{w}_k]\!] + [\![\boldsymbol{\Delta w}_k]\!]$.
Output $[\![y_k]\!]$.</td></tr>
<tr><td colspan="2"><b>endfor</b></td></tr>
</table>

This strategy was later generalized to an arbitrary base in [44], but due to the use of binary circuits, 2^{n_b} is the most efficient choice, as divisions and multiplications by this factor in the circuit are just implemented for free as bit-shifts in the clear.

While the packing operation improves the efficiency of the homomorphic computations, on the garbled circuit side of the protocol, it has the effect of increasing

[1] The shift factor fixes the sign convention between the bit representation ($-a \equiv 2^{n_b} - a$) and the modular arithmetic ($-a \equiv n - a$), working always in the range $[0, 2^{n_b})$, and avoiding errors in the conversion between both representations. Hence, it is not an integral part of the packed formulation, and shall only be applied before a conversion protocol.

the size of the used circuits, multiplying it by the number of values packed into the same encryption. Thus, the complexity of the executed garbled circuits is preserved after packing (lowered if OT reduction techniques are used for each packed block), while the conversion protocols also get an increase in performance, as only one conversion is needed for each encryption containing several packed numbers.

Turning to the secure hybrid block protocol, the packed elements must be processed all together, applying the same coefficients to all of them. Hence, the normal LMS algorithm cannot take advantage of packing, as the filter is kept constant for each group of packed samples, and the update equation has to be slightly modified in order to account for the average error of the whole set of packed samples instead of the error of individual samples; this filter is known as the Block LMS algorithm [63], in which the update equation is

$$\boldsymbol{w}_{n+1} = \boldsymbol{w}_n + \mu \sum_{i=0}^{N_b-1} \boldsymbol{u}_{n \cdot N_b + i} \cdot e_{n \cdot N_b + i}, \tag{3.9}$$

where N_b represents the size of the block. The usual choice of N_b for the Block LMS filter is $N_b = N_E$, as it yields the minimum computational complexity.

Since the packing factors 2^{n_b} are chosen to be powers of two, the bit-conversion protocol automatically unpacks the numbers without any extra complexity, and the conversion to homomorphic encryption after the circuit evaluation is performed for each unpacked number in parallel.

The communication complexity of the hybrid block protocol, taking into account that the XOR gates are free of communication for the used implementation, is exactly the same as for the hybrid protocol:

$$\mathrm{Cpx}_{cm,HB} = (N_E - 1 + 3N_{\mathrm{iter}} + 5N_E N_{\mathrm{iter}})|E_H| + N_{\mathrm{iter}}|E_C|(19n_x + 7n_{\mathrm{sec}} + 24n_f). \tag{3.10}$$

This complexity is linear in the number of iterations, the size of the filter and the bit size of the numbers, and it is independent of the number of packed coefficients.

3.6.5. Fast Implementation

The hybrid block protocol is far more efficient than the one based solely on garbled circuits. Nevertheless, the conversion protocols introduce an overhead, and the fact that the input values to the rounding garbled circuits are generated on the fly prevents much of the preprocessing that garbled circuits would need to compensate the complexity of the oblivious transfers. The gap in computational complexity with respect to the solution based on homomorphic processing is too big (cf. Section 3.8.1), especially when using a high precision bit representation. Thus, we have come to a much more efficient solution that, in order to tighten that gap, avoids the use of circuits, and substitutes them by an approximate rounding protocol with statistical security. The block implementation can also profit from the use of this solution, with a decrease on the maximum packing efficiency, as now the number of packed coefficients is bounded by $N_b^{(FB)} \leq \lfloor \frac{n_{\text{cipher}}}{n_b+n_{\text{sec}}} \rfloor$, instead of $N_b^{(HB)} \leq \lfloor \frac{n_{\text{cipher}}-n_{\text{sec}}}{n_b} \rfloor$, where $n_b = n_x + 3n_f$ is the maximum number of bits that a coefficient can occupy, and n_{sec} is the number of security bits required for the protocol. In this case, the approximate rounding protocol also performs the unpacking of the results; it is described in its complete form in the next subsection. The implementation of this fast protocol replicates exactly the implementation of the hybrid protocols, without the generation and use of the garbled circuits, substituted by the much more efficient approximate rounding protocol; thus, for the sake of brevity, we omit its sketch. The disadvantage is that the rounding error rises with this protocol; however, it is compensated by a reduction of the complexity gap with respect to the solely homomorphic solution.

Algorithm 4 Hybrid Block (HB) PrivateLMS Protocol

Inputs: $\mathcal{A}$: $d_n, \boldsymbol{w}_0$; $\mathcal{B}$: u_n,$\boldsymbol{w}_0$.
Outputs: $[\![y_n]\!]$.

$\mathcal{A}$	$\mathcal{B}$
Encrypt her inputs. $\mathcal{A}$ executes `generateGC`() for the unpacking, parallel rescaling and output circuits in each of the first $m \leq N_{\text{iter}}$ iterations, and sends these garbled circuits to $\mathcal{B}$; the circuits for the remaining iterations can be generated and sent during the execution of the previous ones.	Pack the input vector as $X_j^{(k)} = \sum_{i=0}^{N_b-1} 2^{n_x+3n_f} \cdot u_{k\cdot N_b+i-j}, j = \{0, \dots, N_E - 1\}$.
for $k = 1$ **to** $\lceil N_{\text{iter}}/N_b \rceil$	
	Perform the packed vector multiplication $[\![\boldsymbol{y}_k]\!] = [\![\boldsymbol{w}_k]\!] \cdot \boldsymbol{X}^{(k)}$.
Convert $[\![\boldsymbol{y}_k]\!]$ to its unpacked bit-representation using the bit conversion protocol. Perform parallel OT protocols so that $\mathcal{B}$ get the input keys to initialize the circuit corresponding to the k^{th} iteration.	
	Execute the unpacking and parallel rescaling circuit, using the received input keys from $\mathcal{A}$.
The output of the circuit $[\![y_{k\cdot N_b+i}]\!]_b, i = \{0, \dots, N_b - 1\}$ is converted back to a homomorphic encryption $[\![y_{k\cdot N_b+i}]\!], i = \{0, \dots, N_b - 1\}$.	
	Obtain $[\![e'_{k\cdot N_b+i}]\!] = \mu \cdot ([\![d_{k\cdot N_b+i}]\!] - [\![y_{k\cdot N_b+i}]\!]), i = \{0, \dots, N_b - 1\}$. Perform the scalar multiplication $[\![\boldsymbol{\Delta w}_k]\!] = \sum_{i=k\cdot N_b}^{(k+1)\cdot N_b-1} [\![e_i]\!] \cdot \boldsymbol{u}_{i-N_E+1}$. Update the coefficients vector $[\![\boldsymbol{w}_{k+1}]\!] = [\![\boldsymbol{w}_k]\!] + [\![\boldsymbol{\Delta w}_k]\!]$. Output $[\![y_{k\cdot N_b+i}]\!], i = \{0, \dots, N_b - 1\}$.
endfor	

The communication complexity of the fast implementation, in normal and block forms respectively, is

$$\mathrm{Cpx}_{cm,FP} = (4N_{\mathrm{iter}} + N_E - 1)|E_H|, \quad \mathrm{Cpx}_{cm,FB} = \left(\left(3 + \frac{1}{N_b}\right) N_{\mathrm{iter}} + N_E - 1\right) |E_H|, \tag{3.11}$$

where N_b is the number of packed coefficients for the block protocol. This complexity is of the same order as that of the protocol that uses only homomorphic processing.

3.6.5.1. Approximate Rounding and Unpacking protocol

We have developed several protocols for quantization under encryption. In Appendix 3.B, we present two versions of them, with unconditional blinding of the used values; one is an exact protocol that produces the same results as the clear-text quantization, and the other is an approximate faster version; both use comparison circuits for performing the quantization operation. We sketch in Algorithm 5 a third version of the secure quantization protocol where a statistical blinding is used instead of an unconditional one, avoiding the need for comparison circuits. The security of the algorithm is controlled by the parameter n_{sec}, chosen such that $2^{-n_{sec}}$ is negligible; then, the distribution of the blinded values is indistinguishable from a random sequence (a distinguisher will succeed with probability $2^{-n_{sec}}$); hence, due to the sequential composition of statistically secure protocols and the semantic security of the encryption system, the protocol can be proven statistically secure under the random oracle model using a simulator argument.

It can be seen that the rounding error that it introduces is higher than that of a linear quantizer, and it is not uniform between $[-\frac{1}{2}, \frac{1}{2})$, but triangular between $[-1, 1)$, thus duplicating the quantization MSE.

The communication complexity of the protocol is

$$\mathrm{Cpx}_{cm,RP} = (N_b + 1)|E_H|, \tag{3.12}$$

where N_b is the number of packed elements in one cipher, and $|E_H|$ is the bit size of a homomorphic encryption. Due to the great benefit in efficiency with respect to the impact on accuracy (cf. Section 3.7.2), this is the chosen protocol for the fast implementation of the private LMS algorithm.

We must point out that this solution to the cipher blowup problem represents the minimum increase in computation and communication complexity with respect to plain homomorphic processing. We have discarded the possibility of using a different number encoding due to the following reasoning: our approximate rounding protocol is approximately equivalent to a secure multiplication protocol in terms of bandwidth and total computation (at most, one per iteration in the implementation of the whole LMS); using a different encoding like the one in [95], would introduce the overhead of working with triplets of encryptions for each number, adding two multiplication protocols per encrypted multiplication, and twelve multiplication protocols and two comparison protocols per encrypted addition; hence, our solution is notably more efficient.

Algorithm 5 Approximate Rounding and unpacking Protocol

Inputs: $\mathcal{A}$: Quantization step $\Delta = 2^l$ and a security parameter n_{sec};
$\mathcal{B}$: $\llbracket x_{\text{pack}} \rrbracket = \llbracket \sum_{i=0}^{N_b-1} x_i \cdot 2^{i\cdot(n_b+n_{sec}+1)} \rrbracket$, $\Delta = 2^l$, n_{sec}
Outputs: $\{\llbracket Q'_\Delta(x_i) \rrbracket\}_{i=0}^{N_b-1}$.

$\mathcal{A}$	$\mathcal{B}$
	Generate $x_i^{(b)} \in_R \{2^{nb-1}, \ldots, 2^{n_b-1} + 2^{n_b+n_{sec}}\}, i = \{0, \ldots, N_b - 1\}$, with which he shifts and additively blinds the packed encryptions: $\llbracket x_p^{(a)} \rrbracket = \llbracket x_{\text{pack}} \rrbracket + \llbracket \sum_{i=0}^{N_b-1} x_i^{(b)} \cdot 2^{i\cdot(n_b+n_{sec}+1)} \rrbracket$, homomorphically. Send $\llbracket x_p^{(a)} \rrbracket$ to $\mathcal{A}$.
Decrypt and unpack the received encryptions, obtaining $\{x_i^{(a)}\}_{i=0}^{N_b-1}$.	
Apply a linear quantizer with step $\Delta = 2^l$ to their clear-text vectors component-wise, obtaining $\{Q_\Delta(x_i^{(a)})\}_{i=0}^{N_b-1}$ and $\{Q_\Delta(x_i^{(b)})\}_{i=0}^{N_b-1}$, respectively.	
Encrypt her quantized vector component-wise, and send the encryptions back to $\mathcal{B}$.	
	Unblind the quantized encrypted values obtained from $\mathcal{A}$, obtaining the encrypted quantizations of the original values $\{\llbracket Q'_\Delta(x_i) \rrbracket\}_{i=0}^{N_b-1} = \{\llbracket Q_\Delta(x_i^{(a)}) \rrbracket - Q_\Delta(x_i^{(b)})\}_{i=0}^{N_b-1}$.

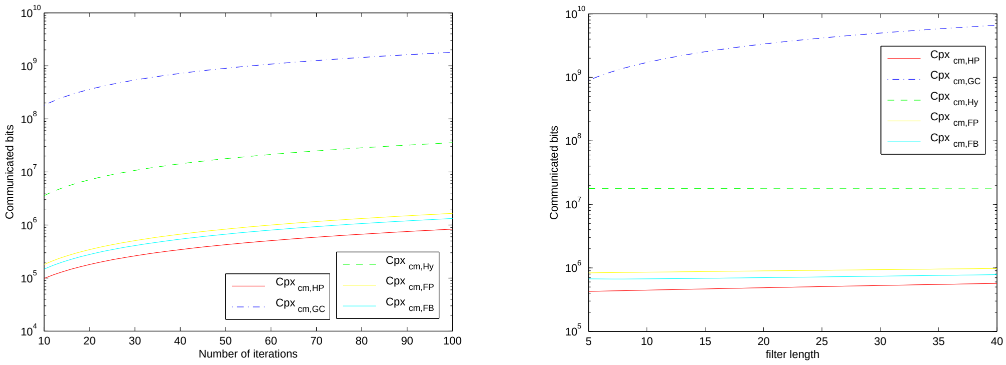
3.7. Evaluation

In this section, we perform a comparison of the developed protocols in terms of bandwidth, computational complexity and finite precision effects, providing also an evaluation of the chosen techniques for each of the solutions, and their suitability for the application scenarios. In the next section we also introduce a practical implementation of our protocols, that we have used for measuring actual execution times on real machines.

3.7.1. Bandwidth

In terms of communication complexity, the estimated transferred bits for each of the protocols have been given together with their description in the previous section. All the protocols have a communication complexity linear on the number of iterations, the size of the filter and the size of the encryptions; nevertheless, the constants are not the same and the difference is perceptible and significant for normal values of the LMS parameters. As an exemplifying case, Figures 3.2a and 3.2b show the number of communicated bits for each of the protocols for a varying number of iterations and filter length respectively; the length of the encryptions is chosen for mid-term security (2048 bits for Damgård-Jurik modulus, 224 bits for the elliptic curve modulus, and 80 bits for the statistical security parameter used in the conversion protocols).

The obtained results using 32-bit numbers with 16-bit fractional precision are shown for a 5 tap filter in Figure 3.2a and for 50 iterations in Figure 3.2b. It can be seen that the bandwidth of the garbled circuit solutions–only garbled circuits (GC) and hybrid protocol (Hy)–is several orders of magnitude higher than that of the solutions including only homomorphic processing (HP). While the HP protocol needs to transfer two encryptions per iteration (8192 bits), the GC protocol communicates around 165 Mb per iteration for the chosen parameters. Hence, the communication complexity for the HP protocol and the fast protocols (FP and block FB) is higher than that of the clear-text protocols, but still practical; on the other hand, the bandwidth needed by the solutions that include garbled circuits make them almost totally infeasible for practical purposes, even when using small encryptions based on Elliptic Curves. The hybrid protocol presents, though, an intermediate complexity, due to the overhead, w.r.t. the HP solution, imposed by the use of conversion subprotocols for changing between bit-representation and homomorphic encryptions. This overhead will be translated in a decrease in computation load for the hybrid block protocol (cf. Section 3.8.1).



(a)

(b)

Figure 3.2: Communication complexity as a function of the number of executed iterations with $N_E = 5$ (a) and the filter length with 50 iterations (b) for $|E_H| = 4096$, $|E_C| = 224$, $n_{\text{sec}} = 80$, $N_b = \min\left(N_E, \lfloor \frac{n_{\text{cipher}}}{n_x + 3 \cdot n_f + n_{\text{sec}}} \rfloor\right)$, $n_x = 32$, $n_f = 16$.

3.7.2. Error Analysis and Finite Precision Effects

One of the limitations of the presented protocols, inherent to privacy-preserving techniques that deal with encryption based on finite-fields, is the need of using fixed point arithmetic. This is actually not a severe issue, as current implementations of the traditional insecure algorithms also work with finite precision, but the flexibility of floating point yields a much wider range of representable values, and greatly improves on the quantization error propagated to the outputs of the algorithm. Numerical stability and numerical accuracy of the filters, that determine the resilience to quantization errors, come into play when dealing with fixed-point arithmetic.

While this issue is commonly avoided or mitigated by the use of a sufficiently large plaintext size to accommodate the needed precision, we believe that it is necessary to devote some space to calculating which is the needed precision and plaintext size for keeping the output Mean Square Error (MSE) within a given bound. In this section we review the error analysis of adaptive algorithms working with fixed-point arithmetic and apply it to the specific cases that our protocols involve. We assume that the inputs and outputs are quantized with n_f bits for their fractional part (of the total n_x bits used for coding), and the filter coefficients and some intermediate results are quantized with n_{wf} bits and n_{If} bits for their fractional part respectively. The use of a different quantization level for vector coefficients is explained in Section 3.6.

Neglecting the overflow effects and assuming stationary d_n and u_n with variances σ_d^2 and σ_u^2, i.i.d.[2] u_n, and uniform and independent quantization errors of the inputs (with variance $\sigma^2 = \frac{2^{-2n_f}}{12}$) and intermediate values (with variance $\sigma_I^2 = \frac{2^{-2n_{If}}}{12}$, and $\sigma_w^2 = \frac{2^{-2n_{wf}}}{12}$ for the filter coefficients), it can be shown that the average power of the error (MSE, or *Mean-Square Error*) at the output in steady-state is [54]

$$\begin{aligned}\sigma_o^2(c,d) =& \sigma_{\min}^2 + \frac{\mu\sigma_{\min}^2 \mathrm{tr}\boldsymbol{R}}{2-\mu\mathrm{tr}\boldsymbol{R}} + \left(||\boldsymbol{w}^*||^2 + \frac{1}{2}\mu\sigma_{\min}^2 N_E\right)\sigma^2 + c\sigma_I^2 + \\ & \frac{N_E\sigma_w^2 + d\cdot\mathrm{tr}\boldsymbol{R}\sigma_I^2 + \mu^2\cdot\sigma^2\left(\left(1 + c\frac{\sigma_I^2}{\sigma^2} + ||\boldsymbol{w}^*||^2\right)\cdot\mathrm{tr}\boldsymbol{R} + \sigma_{\min}^2 N_E\right)}{2\mu - \mu^2\mathrm{tr}\boldsymbol{R}},\end{aligned} \tag{3.13}$$

where the first two terms correspond to the error of the LMS filter with infinite precision, and the rest of the terms stem from quantization. In Eq. (3.13), $\sigma_{\min}^2 =$

[2] The calculations can be generalized to any u_n through the rotated or uncoupled coordinate space [31], but the i.i.d. case is representative enough of the effects of fixed-point precision on the output error.

$\sigma_d^2 - \boldsymbol{w}^* E\{d_n \mathbf{u}_n\}$ is the error of the optimum Wiener filter $\boldsymbol{w}^*$, tr$\boldsymbol{R}$ represents the trace of the input covariance matrix, and c and d are factors that depend on the way quantization is handled in multiplications:

$$c = \begin{cases} 1, & \text{if only the result of } \boldsymbol{w}_n^T \cdot \boldsymbol{u}_n \text{ in (3.1) is quantized} \\ N_E, & \text{if each intermediate product of } \boldsymbol{w}_n^T \cdot \boldsymbol{u}_n \text{ in (3.1) is quantized.} \end{cases} \tag{3.14}$$

$$d = \begin{cases} 1, & \text{if the product } \mu e_n \text{ is quantized before multiplying by } \boldsymbol{u}_n \text{ in (3.2)} \\ 0, & \text{if there is no intermediate quantization in } \mu e_n \boldsymbol{u}_n \text{ in (3.2).} \end{cases} \tag{3.15}$$

Equation (3.13) is not exactly the same as in [54], as we have considered the most general case of having different quantization levels for inputs, filter coefficients, and also for intermediate values.

If only the inputs are quantized, but the intermediate operations do not perform any additional quantization, then the MSE at the output will be, following a derivation analogous to [54],

$$\sigma_{o,\mathrm{QI}}^2 = \sigma_{\min}^2 + \frac{\mu \sigma_{\min}^2 \mathrm{tr}\boldsymbol{R}}{2 - \mu \mathrm{tr}\boldsymbol{R}} + \left(||\boldsymbol{w}^*||^2 + \frac{1}{2}\mu \sigma_{\min}^2 N_E \right) \sigma^2. \tag{3.16}$$

Hence, for the studied non-block protocols, the error at the output can be expressed as

$$\sigma_{\mathrm{HP}}^2 = \sigma_{o,\mathrm{QI}}^2, \quad \sigma_{\mathrm{GC}}^2 = \sigma_o^2(N_E, 1), \quad \sigma_{\mathrm{Hy}}^2 = \sigma_o^2(1, 0). \tag{3.17}$$

For the fast protocol, the quantization error has a different shape, but the independence assumptions can be applied exactly as in the other protocols, duplicating the power of this quantization error of the intermediate values, that becomes $\sigma_I^2 = 2^{-2n_{If}}/6$.

3.7.2.1. Block LMS protocol

Following a similar derivation to that of Caraiscos and Liu [54] for the BLMS algorithm[3], with the same independence assumptions, it is possible to generalize their formula to provide the following approximation to the error in the Block

[3] The full derivation is rather direct but lengthy, and it is completely shown in Appendix 3.C.

LMS implementation:

$$\begin{aligned}\sigma^2_{o,Bk}(c,d,N_b) =&\sigma^2_{\min} + \frac{\mu\sigma^2_{\min}\mathrm{tr}\boldsymbol{R}}{2-\mu\mathrm{tr}\boldsymbol{R}} + \left(||\boldsymbol{w}^*||^2 + \frac{1}{2}\mu\sigma^2_{\min}N_E\right)\sigma^2 + c\sigma^2_I \\ &+ \frac{\frac{N_E\sigma^2_w}{N_b} + d\cdot\left(N_E\frac{N_b-1}{N_b}\sigma^2_w + \sigma^2_I\mathrm{tr}(\boldsymbol{R})\right) + \mu^2\cdot\sigma^2\left(\left(1+c\frac{\sigma^2_I}{\sigma^2} + ||\boldsymbol{w}^*||^2\right)\cdot\mathrm{tr}\boldsymbol{R} + \sigma^2_{\min}N_E\right)}{2\mu - \mu^2 N_b\mathrm{tr}\boldsymbol{R}}\end{aligned} \tag{3.18}$$

where c has the same meaning as in Eq. (3.13), N_b is the block size, and $d = 1$ when each product in $\mu\sum_k e_k\boldsymbol{u}_k$ in (3.4) is individually quantized, and $d = 0$ otherwise.

This result is coherent with the one obtained by Eweda *et al.* [92] for the adaptive system identification problem, but Eq. (3.18) is more general and takes into account more parameters that allow for a greater flexibility in predicting the error of our implementations. It can be seen that for the same step size μ, both infinite-precision LMS and BLMS have the same misadjustment (first two terms in Eq. (3.18)) and the same average time constant. For the finite-precision algorithms, Eq. (3.18) shows that the BLMS reduces the sensitivity to the quantization error in the filter coefficients when $d = 0$ (first term of the numerator), but the sensitivity to the quantization of the inputs is not altered (third term in Eq. (3.18)); quantization of the filter coefficients has a much more critical and noticeable effect than the quantization of the input values when σ^2 and σ^2_w are comparable, what motivates the conclusions in [92] about the better behavior of BLMS; nevertheless, when $\sigma^2 \gg \sigma^2_w$, the averaging performed by BLMS has a neglibible impact on quantization error resilience, as shown in Section 3.7.2.3; hence, for the same convergence speed, BLMS presents an MSE similar to that of LMS.

3.7.2.2. Transient Deviation due to Finite Precision

As shown in the previous sections, the use of fixed-point precision affects the stationary regime of the algorithms, producing a higher level of noise. Actually, the effect of finite precision is also noticeable in the transient period, introducing errors during tracking and altering the adaptation behavior. Following a similar derivation to that in [30], we have extended the theoretical adaptation curve to the BLMS algorithm. The result for the weigth vector misadjustment $\mathcal{M}_n = E\left[\Delta\boldsymbol{w}_n^T\Delta\boldsymbol{w}_n\right]$, for the same assumptions as in previous sections (cf. Appendix 3.C.1), is

$$\mathcal{M}_n = \mu^2\cdot N_b\cdot N_E\cdot\left[A\cdot n\gamma^{2(n-1)} + \frac{A}{\gamma-\gamma^2}\left(\gamma^n - \gamma^{2n}\right) + \frac{B}{1-\gamma^2}\left(1-\gamma^{2n}\right)\right] + \frac{N_E\sigma^2_w}{1-\gamma^2}\left(1-\gamma^{2n}\right), \tag{3.19}$$

with

$$A = 2\sigma^2\sigma^2_u||\boldsymbol{w}^*||^2, \quad B = \sigma^2_u\left[\sigma^2(1+||\boldsymbol{w}^*||^2) + c\sigma^2_I\right] + \sigma^2\sigma^2_{\min}, \quad \gamma = 1-\mu N_b\sigma^2_u. \tag{3.20}$$

Eq. (3.19) gives the evolution of the MSE of the filter coefficients that the finite precision algorithm introduces with respect to the infinite precision LMS during the adaptation period. The notation and parameters are the same as for Eq. (3.18). This error evolves with a fixed time constant, equal to that of the infinite precision algorithm, until reaching the stationary state for which the output error is given by Eq. (3.18). This evolution is shown in Figure 3.3 for the hybrid protocol for different values of the adaptation step and used fractional bits. For a fair comparison, it must be taken into account that the index n refers to successive updates of the vector coefficients, that in BLMS are produced every N_b output samples instead of every sample.

3.7.2.3. Comparison and Evaluation

Figure 3.4 shows a representative case of the excess MSE (i.e., $E\{e^2\}-\sigma^2_{LMS_\infty}$) with respect to the infinite precision LMS, obtained for each of the proposed protocols for varying bit-size of the fractional part. The theoretical approximations given by Eq (3.18) are labeled with the subindex *th*, and the experimental results, with the subindex *exp.* The Garbled Circuit implementation presents the highest error, mainly due to the use of the same bit size for vector coefficients as for input quantization, and the quantization performed after each multiplication. The hybrid protocol is the most robust against quantization errors, due to the use of a higher resolution for the vector coefficients, and the presence of quantization only in the outputs, and in no other internal calculations. On the other hand, the fast protocol presents a MSE slightly higher than the hybrid protocol, due to the approximate quantization of the outputs. Finally, the MSE produced by the block protocols is virtually the same as the MSE of the corresponding non-block implementations, due to the predominant effect of input quantization over that of filter coefficients quantization. The experimental results are obtained as the average error after running the algorithms for 40,968 iterations in steady-state regime, for the system identification setup with $\sigma_u^2 = 0.25$, $\sigma_d^2 = 0.2821$, $\mu = 2^{-8}$, $\sigma^2_{\min} = 2.5 \cdot 10^{-5}$ and $\sigma^2_{LMS_\infty} = 2.5147 \cdot 10^{-5}$. The homomorphic processing protocol is not shown, as its cipher blows up before reaching the steady-state in practical cases; e.g., a modulus of 2048 bits can only hold 28 iterations using 48 bit numbers with 8 bits for the fractional part. Nevertheless, in theory and with a big enough cipher, it would be the most robust protocol due to the absence of intermediate quantizations. Besides this protocol, the concordance between the theoretical approximation and the experimental results in all the other protocols is remarkable, given the magnitude of the errors with which we are working, assessing the validity of the initial assumptions for obtaining Eq (3.18).

There are several effects noticeable in Figure 3.4 that deserve a comment: on the one hand, the experimental results for the Garbled Circuit protocol are not

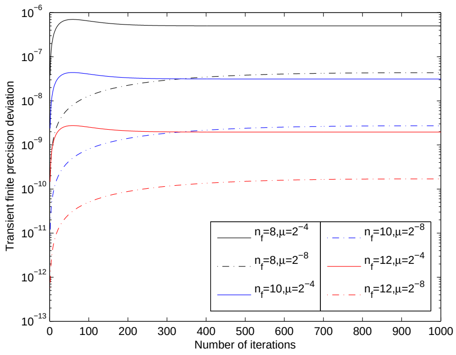


Figure 3.3: Excess error during the transient period for the hybrid protocol, with $n_x = 48$, and a 4-tap adaptive filter.

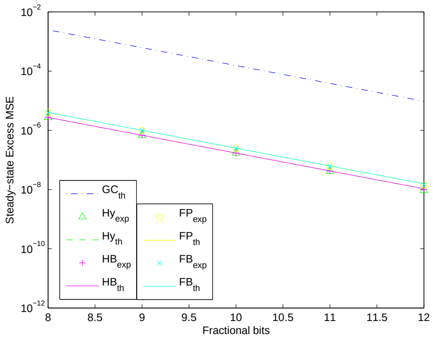


Figure 3.4: Steady-state excess error for varying fractional precision, with $n_x = 48$, and 12-tap adaptive filter, packing $N_b = N_E = 12$ coefficients in the block protocols.

shown, as for the used bit-sizes the precision used for filter coefficients is too low (equal to that of the inputs and intermediate results), and it suffers from stalling effects, that prevent it from converging; as a consequence, it needs a much higher precision in order to avoid stalling, and even when converging, as shown in the plot, the error that it produces is significantly higher than that of the other protocols. The second observable fact is that the gap of precision in block protocols is almost negligible when $\sigma^2 \gg \sigma_w^2$. This difference is not noticeable in Figure 3.4, and it would only be significant with very long blocks $N_b \gg 1$ or with $\sigma^2 \approx \sigma_w^2$. The way our protocols are designed avoids this second condition, as they use always a higher precision for the filter coefficients than for the inputs/outputs.

At last, the value of N_b is limited by the maximum plaintext size and the number of bits used for representing each number. Thus, Eq. (3.18) can be used together with the packing limits for the block protocols $N_b^{(FB)} \leq \lfloor \frac{n_{\text{cipher}}}{n_b + n_{\text{sec}}} \rfloor$, $N_b^{(HB)} \leq \lfloor \frac{n_{\text{cipher}} - n_{\text{sec}}}{n_b} \rfloor$, for finding a trade-off between the committed error due to the used precision, and the complexity of both protocols, dependent on the number of coefficients that are packed together.

3.8. Practical Implementation

In this section, we present and comment the results of a practical implementation of the proposed protocols. For this purpose, we have chosen the Damgård-Jurik [79] extension of Paillier cryptosystem, due to its flexibility for fitting larger plaintexts with a constant expansion ratio. For the protocols involving garbled circuits, we have chosen the XOR-free garbled circuit solution in [141], and the efficient oblivious transfer protocols of [129] with EC-ElGamal encryptions, aiming to the most efficient algorithms currently available for implementing garbled circuits.

For the evaluation of computational complexity, we have implemented the presented protocols and their block versions in C++ using the crypto++ library [1] for the elliptic curves cryptosystems, and the GNU GMP library [3] for multiprecision arithmetic, and we have provided our own implementation of Damgård-Jurik encryptions, with some efficiency improvements in modular exponentiations, detailed in Appendix 3.A. We use these implementations in order to plot the execution times of the three protocols and compare them in terms of CPU usage. We have made the whole software package of our implementation available at [218].

3.8.1. Computational Load

We have measured the computational load of the developed algorithms through the total computation time that their efficient implementation yields on a PC with no parallelization, for a fair comparison. Nevertheless, these protocols, and especially their block versions, are easily parallelizable, obtaining a great reduction in execution time when several cores are available. The experiments were performed using our C++ implementation on an Intel Core2Duo processor at 3 GHz with 4GB of RAM running a 64-bit linux distribution. In order to measure only computation times, we have neglected the communication stack, and we have run in the same core the client and the server sequentially, obtaining the aggregated computation times for both parties.

Figure 3.5 shows the aggregated computation time for the 48 initial iterations of each of the presented protocols, as a function of the filter size. The three protocols involving garbled circuits are the most expensive ones, due to the load that oblivious transfers impose. While this load is normally absorbed through precomputation, with an adaptive algorithm it is not possible to perform the heavy encryption operations a priori, as they involve the results generated in each iteration; hence, no precomputation is applied to any of the performed operations. This has also an impact on their parallelization, as each oblivious transfer round involves only the bits of one input. This is especially critical in the case of the hybrid protocol, as the small OTs in each iteration cannot be joined together into a longer and more efficiently reducible OT. On the other hand, the packing performed in the hybrid block protocol allows for this reduction, greatly improving computational load as the number of packed coefficients (chosen to equal the size of the filter) increases.

Finally, the execution times of the fast protocols are several orders of magnitude below those of the garbled circuits solutions, and slightly increase the complexity of the homomorphic computation protocol due to the addition of the rounding protocols. This is a remarkable result, taking into account that without this rounding subprotocols, the whole homomorphic computation protocol is completely unusable due to cipher blowup. For the fast protocol, the block-based one does not improve on the computational load, as the fast rounding protocol requires a whole unpacking protocol for each of the packed numbers, and it does not yield the same improvement as in the hybrid block protocol. Hence, the fast protocol is more time-efficient than its block version.

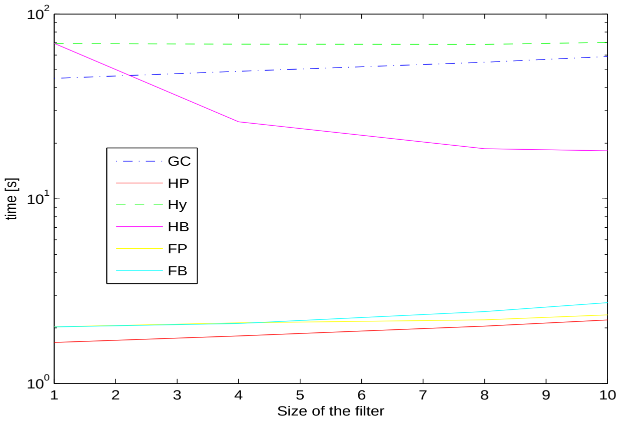


Figure 3.5: Aggregated computation time for 2048 bits moduli, $|E_C| = 224$, $n_{\text{sec}} = 80$, $n_x = 32$, $n_f = 16$, 48 iterations and increasing filter size and maximum packed coefficients.

3.9. Conclusions and Further Work

Addressing privacy in adaptive filtering applications is an important open issue in the field of Signal Processing in the Encrypted Domain. This chapter has presented the problem of privacy-preserving adaptive filtering, with several representative scenarios and their trust model and privacy requirements. Due to the impossibility of using a practical full homomorphism, we have proposed several novel solutions employing different techniques, like garbled circuits, additive homomorphisms and interactive protocols, looking for the optimal trade-off in terms of complexity and output error; we have also provided several private quantization algorithms of independent interest to tackle the cipher blowup problem; we have implemented all our novel protocols for the Private LMS algorithm in a working prototype, and we have performed a comparison in terms of bandwidth and computational complexity, concluding that garbled circuits are still far from providing an efficient solution to adaptive filtering, and interactive approximate protocols with statistical security can yield much more practical solutions.

We have also tackled the issue of the limitation to fixed-point precision when working with encrypted values, resorting to analytical studies on the impact of finite-precision in the output error of the used adaptive filters, during the transient period and in steady-state regime, particularizing the expressions to each of the studied cases. The fast protocols that we have introduced are almost as robust as the original (B)LMS algorithm with respect to quantization errors, while presenting low computational and communication complexity.

This chapter covers the two main problems of any secure adaptive filtering algorithm, namely cipher blowup and precision limits due to the use of fixed point arithmetic. Further research will aim also at the implementation of more complex nonlinear functions, being this problem not specific of adaptive filtering. Hence, the present work opens the door to further improvements in secure adaptive filtering, setting the basis and a reference implementation for the development of new solutions.

3.A. Fast Encryption and Decryption for Damgård-Jurik Cryptosystem

Encryption and decryption are two of the most costly operations, due to the heavy modular exponentiations that they must perform. For our implementations, we have used a different version of the decryption operation, and for the private encryption of the Paillier cryptosystem (and the Damgård-Jurik extension) that enhance the performance of the original methods. This appendix describes both methods. Modular exponentiations are the most computationally demanding basic operations, whose complexity is linear in the exponent size $|e|$ and quadratic in the modulus size $|n|$ (i.e., $O(|e||n|(|n|-1))$). Thus, reducing the bit size of the involved operands yields important efficiency gains. The presented reductions are based on using the knowledge of the factorization of the public modulus n, enhancing all decryption operations and encryption operations performed by a party with decryption privileges (*private* encryption). Looking at the most common two-party scenarios of homomorphic encryption, the party that owns the data and owns the decryption keys is usually the client, that normally has a processing power lower than the server; hence, it makes sense to optimize the operations that this party must perform, and this is exactly what our modifications do. We will preserve the notation used in Section 2.2.2.1.

Decryption

Let $L_a(b)$ be defined as $L_a(b) = \frac{b-1}{a}$, for $b \equiv 1 \mod a, 0 < b < a^2$, as in Paillier's work. In [79], it is suggested that the decryption operation, after the exponentiation $c^d \mod n^{s+1}$, be divided into two parts, using $L'_p(c^d) = L_p(c^d)\cdot q^{-1}$ and $L'_q(c^d) = L_q(c^d)\cdot p^{-1}$ instead of $L_n(c^d)$, and then joined using the Chinese Remainder Theorem (CRT). While this strategy can provide a speed-up in the computations, as each part of the decryption works with half-sized numbers, the initial exponentiation is still the most costly operation. We next show how the knowledge of the factorization of n allows also for breaking up this exponentiation into two parts.

For a message x, its encryption $c = (1+n)^x r^{n^s} \mod n^{s+1}$, can be reduced modulo p^{s+1} and q^{s+1}, obtaining two partial encryptions with half the size of c: $c_p = (1+n)^x r^{n^s} \mod p^{s+1}$ and $c_q = (1+n)^x r^{n^s} \mod q^{s+1}$. By Carmichael's Theorem, the order of the units in the group $\mathbb{Z}_{p^{s+1}}$ (resp. $\mathbb{Z}_{q^{s+1}}$) is a divisor of $p^s(p-1)$ (resp. $q^s(q-1)$). Hence, the minimum exponent that cancels the effect of r^{n^s} is $p-1$ (resp. $q-1$), that is

$$\begin{aligned} L'_p(c_p^{p-1}) =& L_p\left(((1+n)^x)^{p-1}\, r^{q^s\cdot p^s\cdot(p-1)}\right)\cdot q^{-1} \\ \equiv& \left(1 + x(p-1)n + \binom{x(p-1)}{2} n^2 + \ldots + \binom{x(p-1)}{s} n^s\right) \mod p^s, \end{aligned} \tag{3.21}$$

and analogously for q. Applying the decryption algorithm with p and q for both parts, and multiplying afterwards each of them by the inverses of $p-1$ and $q-1$, the desired result is obtained:

$$d_p = \mathrm{dec}_{p^s}(c_p^{p-1})\cdot(p-1)^{-1} \equiv x \mod p^s, \quad d_q = \mathrm{dec}_{q^s}(c_q^{q-1})\cdot(q-1)^{-1} \equiv x \mod q^s. \tag{3.22}$$

The application of the CRT yields that, if a_p and a_q are two integers such that $a_p\cdot p^s + a_q\cdot q^s = 1$, then $x \equiv d_p\cdot a_q\cdot q^s + d_q\cdot a_p\cdot p^s \mod n^s$.

Finally, as the values of $(p-1)^{-1} \mod p^s$, $(q-1)^{-1} \mod q^s$, $a_q\cdot q^s \mod n^s$ and $a_p\cdot p^s \mod n$ can be precalculated, and the L' functions can be executed once for the highest power of p and q and subsequently modularized for the rest of the iterations of the algorithm (as $L_b(a \mod b^{j+1}) \equiv L_b(a \mod b^{s+1}) \mod b^j$), neglecting the complexity of a modularization and the addition/subtraction of a unit, the total decryption complexity is reduced to

$$2\left(X_{\frac{(s+1)|n|}{2},\frac{|n|}{2}} + D_{\frac{(s+1)|n|}{2}} + P_{s|n|} + \sum_{k=2}^{s}\left((k-1)(P_{\frac{k|n|}{2}} + A_{\frac{k|n|}{2}})\right)\right) + A_{s|n|}, \tag{3.23}$$

where $X_{a,b}$ is the computational complexity of an exponentiation with modulus size a and exponent size b, A_b and P_b are the complexity of a modular addition and product with modulus size b respectively, and D_a is the complexity of an integer division with dividend's size a. This results can be compared to the complexity of a regular decryption, performed as stated in [79],

$$X_{(s+1)|n|,|n|} + D_{(s+1)|n|} + \sum_{k=2}^{s} \left((k-1)(P_{k|n|} + A_{k|n|})\right). \tag{3.24}$$

The reduction factor in complexity due to splitting the exponentiation is almost four.

Encryption

For regular encryption there is no additional gain to the one pointed out in Paillier's original work, by virtue of which taking $g = 1 + n$ reduces the exponentiation $g^x \mod n^2$ to a product $g^x \equiv (1 + x \cdot n) \mod n^2$, generalized in [79] to n^{s+1} as a sum of s chained products; the exponentiation r^{n^s} is, in principle, unavoidable. Nevertheless, when the encryption is performed by a party with decryption capabilities ("private" encryption), the knowledge of the private key allows for further improvements on efficiency, applying the same rationale as for fast decryption. In this case, the reduction seeks partitioning the exponentiation r^{n^s} into two exponentiations with half-sized base and exponent.

Given $a_{p^{s+1}}$ and $a_{q^{s+1}}$ such that $a_{p^{s+1}} \cdot p^{s+1} + a_{q^{s+1}} \cdot q^{s+1} = 1$, $r^{n^s} \mod n^{s+1}$ can be calculated as

$$r_p \equiv r^{p^s(q^s \mod (p-1))} \mod p^{s+1}, \quad r_q \equiv r^{q^s(p^s \mod (q-1))} \mod q^{s+1},$$
$$r^{n^s} \equiv r_p \cdot a_{q^{s+1}} \cdot q^{s+1} + r_q \cdot a_{p^{s+1}} \cdot p^{s+1} \mod n^{s+1}. \tag{3.25}$$

Precalculating the values of $a_{q^{s+1}} \cdot q^{s+1} \mod n^{s+1}$ and $a_{p^{s+1}} \cdot p^{s+1} \mod n^{s+1}$, the complexity of each encryption is reduced to

$$2X_{\frac{(s+1)|n|}{2},\frac{(s+1)|n|}{2}} + 2(s+1)P_{(s+1)|n|} + 2s \cdot A_{(s+1)|n|}, \tag{3.26}$$

compared to $X_{(s+1)|n|,(s+1)|n|} + 2s \cdot P_{(s+1)|n|} + (2s-1)A_{(s+1)|n|}$ of a normal encryption, which yields a complexity reduction almost by a factor of four.

3.B. Cipher renewal: quantization under encryption

In order to renew the cipher and eliminate part of the excess of precision accumulated by the lack of a division operation, it is necessary to quantize the encrypted values. For this purpose, and to preserve perfect secrecy, we have developed interactive protocols of independent interest for performing quantization:

Let $[x] \in \mathbb{Z}_n$ be a class in $\mathbb{Z}_n$, and x its positive representative in the interval $x \in [0, n)$. $\mathcal{A}$ and $\mathcal{B}$ possess their respective shares x_A, x_B of x (i.e. $x_A + x_B \equiv x \mod n$). Both $\mathcal{A}$ and $\mathcal{B}$ want to requantize x with a step $\Delta \in (2, \lceil n/2 \rfloor)$, with a maximum quantization error of Δ. Let us assume that $\mathcal{A}$ knows the decryption key of an additive homomorphic cryptosystem, and both $\mathcal{A}$ and $\mathcal{B}$ can produce encryptions using this cryptosystem. The scenario can be plotted also with a threshold homomorphic cryptosystem, with straightforward modifications.

If $\mathcal{B}$ owns an encryption of $[\![x]\!]$, then he generates a random $x_B \in \mathbb{Z}_n$, blinds with it the encryption of $[\![x]\!]$, and sends the result $[\![x + x_B \mod n]\!]$ to $\mathcal{A}$, who decrypts $x_A = x + x_B \mod n$. Then, both parties start with a share of x.

Each party quantizes his/her share $x_{AQ} = \left\lceil \frac{x_A}{\Delta/2} \right\rfloor$, $x_{BQ} = \left\lceil \frac{x_B}{\Delta/2} \right\rfloor$; with these values, both parties can obtain the bit representation of their respective quantities and run a binary comparison protocol (cf. Appendix 3.D) $x_{BQ} > \left\lceil \frac{n}{\Delta/2} \right\rfloor - x_{AQ}$, ending up with an encryption of the binary comparison.

Then, $\mathcal{A}$ can obtain $[\![Q_R(x)]\!] = [\![x_{AQ}]\!] + [\![x_{BQ}]\!] - \left\lceil \frac{n}{\Delta/2} \right\rfloor \cdot \left[\!\!\left[x_{BQ} \geq \left\lceil \frac{n}{\Delta/2} \right\rfloor - x_{AQ} \right]\!\!\right]$. We denote the result $Q_R(x)$ because it does not coincide exactly with the quantization $Q(x)$ when performed in the clear, because $Q_R(x)$ is quantized with a precision of $\Delta/2$, but the split in two shares introduces an error of ± 1 in the quantization of x. Thus, even when the obtained precision is Δ, the resulting encrypted number must be scaled by $\Delta/2$ after decryption in order to obtain the true quantized value.

The previous protocol could be thought of as a *fast* version of the quantization protocol, that has the drawback of introducing some noise due to the independent quantization of both shares. When the quantization must yield exactly the same results as in the clear, we can use an *exact* version of the previous protocol, that provides a perfect quantization, with the same result as if performed in the clear, at the cost of an increased computation and communication complexity. We now describe this *exact* solution.

After splitting x in two shares x_A and x_B, each party quantizes his share with step Δ, obtaining respectively $x_{AQ} = \lceil \frac{x_A}{\Delta} \rfloor$, $x_{Ar} = x_A \mod \Delta$, and $x_{BQ} = \lceil \frac{x_B}{\Delta} \rfloor$, $x_{Br} = x_B \mod \Delta$; both have the quantity $n_\Delta = n \mod \Delta$ in the clear. The quantization of x as a function of the previous four values can be expressed as

$$\begin{aligned}\llbracket Q(x)\rrbracket = \llbracket x_{AQ}\rrbracket + \llbracket x_{BQ}\rrbracket + \left(1-2\left[x_{Br}\geq\left\lceil\frac{\Delta}{2}\right\rceil\right]\right)\cdot\left(1-xor\left(\left\llbracket x_{Ar}\geq\left\lceil\frac{\Delta}{2}\right\rceil\right\rrbracket,\left[x_{Br}\geq\left\lceil\frac{\Delta}{2}\right\rceil\right]\right)\right)\\ \cdot\left(\left\llbracket x_{Ar}+x_{Br}\in\left[\left\lceil\frac{\Delta}{2}\right\rceil,\left\lceil\frac{3\Delta}{2}\right\rceil\right)\right\rrbracket + \left(\llbracket x_{Ar}+x_{Br}\in\mathcal{I}_{n_\Delta}\rrbracket - \left\llbracket x_{Ar}+x_{Br}\in\left[\left\lceil\frac{\Delta}{2}\right\rceil,\left\lceil\frac{3\Delta}{2}\right\rceil\right)\right\rrbracket\right)\right.\\ \left.\cdot\left\llbracket x_{BQ}\geq\left\lceil\frac{n}{\Delta}\right\rceil - x_{AQ}\right\rrbracket\right) - \left\lceil\frac{n}{\Delta}\right\rceil\cdot\left\llbracket x_{BQ}\geq\left\lceil\frac{n}{\Delta}\right\rfloor - x_{AQ}\right\rrbracket.\end{aligned}\tag{3.27}$$

As the only needed binary operation is the exclusive-OR, for efficiency reasons we avoid the use of garbled circuits and implement it homomorphically as $xor(a,b)=a+b-2a\cdot b$ in $\mathbb{Z}_n$. The set $\mathcal{I}_{n_\Delta}$ represents an interval reduced modulo 2Δ:

$$\mathcal{I}_{n_\Delta}=\begin{cases}\left[\left\lceil\frac{3\Delta}{2}\right\rceil+n_\Delta,\left\lceil\frac{\Delta}{2}\right\rceil+n_\Delta\right)_{2\Delta}, & \text{if } n_\Delta\geq\left\lceil\frac{\Delta}{2}\right\rceil\\ \left[\left\lceil\frac{\Delta}{2}\right\rceil+n_\Delta,\left\lceil\frac{3\Delta}{2}\right\rceil+n_\Delta\right)_{2\Delta}, & \text{if } n_\Delta<\left\lceil\frac{\Delta}{2}\right\rceil,\end{cases}\tag{3.28}$$

being $[,)_{2\Delta}$ the modular reduction of the interval with modulus 2Δ.

The binary comparisons $x_{Ab}=[x_{Ar}\geq\lceil\frac{\Delta}{2}\rceil]$ and $x_{Bb}=[x_{Br}\geq\lceil\frac{\Delta}{2}\rceil]$ are performed by each party independently. $\mathcal{A}$ can encrypt $\llbracket x_{Ab}\rrbracket$ and send it to $\mathcal{B}$, who can perform $(1-2[x_{Br}\geq\lceil\frac{\Delta}{2}\rceil])\cdot(1-xor(\llbracket x_{Ab}\rrbracket,[x_{Bb}]))$ using only homomorphic operations. Each of the two needed interval checks can be performed through two comparison circuits and a homomorphic sum ($\llbracket x\in[a,b)\rrbracket=\llbracket x\geq a\rrbracket-\llbracket x\geq b\rrbracket$). After obtaining these values, the whole expression can be evaluated with 5 homomorphic sums and 3 invocations of the secure multiplication protocol.

The total complexity calculated for the *exact* protocol, for a modulus bit-size $|n|=l$, is

$$\begin{aligned}\mathrm{Cpx}_{cm,EQ}(n,\Delta)=&|E|+3\mathrm{Cpx}_{cm,MULT}+4\mathrm{Cpx}_{cm,COMP}\left(\lceil\log_2\Delta\rceil+1\right)\\ &+\mathrm{Cpx}_{cm,COMP}\left(\left\lceil\log_2\frac{n}{\Delta}\right\rceil\right),\\ \mathrm{Cpx}_{cp,EQ,\mathcal{A}}(n,\Delta)=&\mathrm{Cpx}_{EncBit}+3\mathrm{Cpx}_{cp,MULT,\mathcal{A}}+4\mathrm{Cpx}_{cp,COMP,\mathcal{A}}\left(\lceil\log_2\Delta\rceil+1\right)+\\ &\mathrm{Cpx}_{cp,COMP,\mathcal{A}}\left(\left\lceil\log_2\frac{n}{\Delta}\right\rceil\right),\\ \mathrm{Cpx}_{cp,EQ,\mathcal{B}}(n,\Delta)=&\mathrm{Cpx}_{E}+2\mathrm{Cpx}_{EP}+10\mathrm{Cpx}_{EA}+3\mathrm{Cpx}_{cp,MULT,\mathcal{B}}+\\ &4\mathrm{Cpx}_{cp,COMP,\mathcal{B}}\left(\lceil\log_2\Delta\rceil+1\right)+\mathrm{Cpx}_{cp,COMP,\mathcal{B}}\left(\left\lceil\log_2\frac{n}{\Delta}\right\rceil\right),\end{aligned}$$

where $|E|$ represents the number of bits of an encryption (or share). The subindex *cm* stands for communication complexity, and *cp* for computational complexity for party $\mathcal{A}$ or $\mathcal{B}$, being $\mathrm{Cpx}_{xx,MULT}$ the corresponding complexity of the interactive multiplication protocol; Cpx_{EA}, Cpx_{EP} respectively denote the computational complexity of a homomorphic addition and product (by a known scalar) for the used cryptosystem (or secret sharing scheme), Cpx_E and Cpx_{EncBit} represent the computational complexity for encrypting (sharing) an integer in $\mathcal{Z}_n$ or a bit respectively, and $\mathrm{Cpx}_{xx,COMP}(l)$ is defined in Appendix 3.D.

The *fast* protocol has complexity

$$\begin{aligned}
\mathrm{Cpx}_{cm,EQf}(n,\Delta) =& |E| + \mathrm{Cpx}_{cm,COMP}\left(\left\lceil \log_2 \frac{n}{\Delta}\right\rceil + 1\right),\\
\mathrm{Cpx}_{cp,EQf,\mathcal{A}}(n,\Delta) =& \mathrm{Cpx}_{cp,COMP,\mathcal{A}}\left(\left\lceil \log_2 \frac{n}{\Delta}\right\rceil + 1\right),\\
\mathrm{Cpx}_{cp,EQf,\mathcal{B}}(n,\Delta) =& \mathrm{Cpx}_{E} + \mathrm{Cpx}_{EP} + 2\mathrm{Cpx}_{EA} +\\
& \mathrm{Cpx}_{cp,COMP,\mathcal{B}}\left(\left\lceil \log_2 \frac{n}{\Delta}\right\rceil + 1\right).
\end{aligned}$$

3.C. Finite-precision error analysis of the Block LMS protocol

Starting from Eqs. (3.3) and (3.4), we will follow a derivation similar to that of Caraiscos and Liu [54] to obtain the steady-state error of the BLMS algorithm in the presence of quantization errors. We assume stationary d_n and u_n with variances σ_d^2 and σ_u^2, i.i.d.[4] u_n. We will use the same notation of primed symbols for quantized values and unprimed symbols for infinite precision ones, and Greek letters for the corresponding quantization error. The inputs and outputs are quantized with n_f bits for their fractional part (of the total n_x bits used for coding), and the filter coefficients and some intermediate results are quantized with n_{wf} bits and n_{If} bits for their fractional part respectively, producing errors of power $\sigma_w^2 = \frac{2^{-2n_{wf}}}{12}$ and $\sigma_I^2 = \frac{2^{-2n_{If}}}{12}$. Let us assume that there are no overflows in any of the computations, and a value a is quantized with its corresponding bit-size for the fractional part (i.e., $n_{f,a}$ bits), producing a uniform and independent quantization error of power $\sigma_a^2 = \frac{2^{-2n_{f,a}}}{12}$. For the input sequences,

$$u'_n = u_n + \alpha_n \quad d'_n = d_n + \beta_n,$$

where α_n and β_n are white, mutually independent, and independent of the signals, with zero mean and variance $\sigma^2 = \frac{2^{-2n_f}}{12}$, while the filter coefficients are such that

$$\boldsymbol{w}'_n = \boldsymbol{w}_n + \boldsymbol{\rho}_n, \tag{3.29}$$

being $\boldsymbol{\rho}_n$ a vector of quantization errors of length N_E.

Finally, the output is

$$y'_n = \boldsymbol{w}'^T_n \boldsymbol{u}'_n + \eta_n = \boldsymbol{w}_n^T \boldsymbol{u}_n + \boldsymbol{\rho}_n^T \boldsymbol{u}_n + \boldsymbol{w}_n^T \boldsymbol{\alpha}_n + \eta_n,$$

[4] The calculations can be generalized to any u_n through the rotated or uncoupled coordinate space [31], but the i.i.d. case is representative enough of the effects of fixed-point precision on the output error.

where η_n is an approximately white sequence of quantization noise independent of the signals and the rest of the error sequences, with zero mean and variance $c \cdot \sigma_I^2$, with

$$c = \begin{cases} 1, & \text{if only the result of } \boldsymbol{w}_n^T \cdot \boldsymbol{u}_n \text{ in (3.1) is quantized} \\ N_E, & \text{if each intermediate product of } \boldsymbol{w}_n^T \cdot \boldsymbol{u}_n \text{ in (3.1) is quantized.} \end{cases} \tag{3.30}$$

Hence, the estimation error e'_n is

$$e'_n = d_n - y'_n = \underbrace{d_n - \boldsymbol{w}_n^T \boldsymbol{u}_n}_{e_n} - \left(\boldsymbol{\rho}_n^T \boldsymbol{u}_n + \boldsymbol{w}_n^T \boldsymbol{\alpha}_n + \eta_n \right).$$

Up to this point, the analysis does not deviate from that of the LMS algorithm, and the only difference resides at the calculation of $\boldsymbol{w}_n$, and that all the $y'_{n \cdot N_b + k}$, $k = \{0, \ldots, N_b - 1\}$ share the same $\boldsymbol{w}_{n \cdot N_b + k} = \boldsymbol{w}_{n \cdot N_b}$, $k = \{0, \ldots, N_b - 1\}$. For the sake of clarity, we will use the subindices n and k as $\boldsymbol{w}_n \equiv \boldsymbol{w}_{n \cdot N_b}$ and $a_k \equiv a_{n \cdot N_b + k}$, when there is no ambiguity. Besides that, the same independence assumptions made in [54] are applicable here:

- $\boldsymbol{\alpha}_k$, β_k and η_k are independent of the data and of each other; hence, $\boldsymbol{\rho}_n^T \boldsymbol{u}_k$, $\boldsymbol{w}_n^T \boldsymbol{\alpha}_k$, η_k and β_k are uncorrelated.
- e_k is also uncorrelated to $\boldsymbol{w}_n^T \boldsymbol{\alpha}_k$, η_k and β_k.
- ρ_n depends on data up to time $nN_b - 1$.

Then, the total output mean square error is

$$E[e'^2_k] = E[e_k^2] - 2E[e_k \boldsymbol{\rho}_n^T \boldsymbol{u}_k] + E[(\boldsymbol{\rho}_n^T \boldsymbol{u}_k)^2] + E[(\boldsymbol{w}_n^T \boldsymbol{\alpha}_k)^2] + E[\eta_k^2]. \tag{3.31}$$

$E[e_k^2]$: This term is the MSE of the infinite precision (B)LMS, and it is given by [118]

$$E[e_k^2] = \sigma_{\min}^2 + \frac{\mu \sigma_{\min}^2 \text{tr} \boldsymbol{R}}{2 - \mu \text{tr} \boldsymbol{R}}.$$

$E[(\boldsymbol{w}_n^T \boldsymbol{\alpha}_k)^2] = E[\boldsymbol{w}_n^T \boldsymbol{w}_n] \sigma^2$: For the BLMS, the update equation (3.4) can be expressed as

$$\boldsymbol{w}_{n+1} = \boldsymbol{w}_n - \frac{\mu N_b}{2} \underbrace{\frac{-2}{N_b} \sum_{l=(n-1)N_b}^{nN_b - 1} \boldsymbol{u}_l e_l}_{\widehat{\boldsymbol{\nabla}}_n}, \tag{3.32}$$

being $\widehat{\boldsymbol{\nabla}}_n = \boldsymbol{\nabla}_n + \boldsymbol{N}_n$ the estimate of the true gradient $\boldsymbol{\nabla}_n$ used for the gradient descent algorithm, together with an additive zero-mean estimation noise $\boldsymbol{N}_n$.

When $\boldsymbol{w}_n$ is near the optimal Wiener solution $\boldsymbol{w}^*$, the gradient approaches zero, and the estimate captures only the estimation noise $\boldsymbol{N}_n \approx \frac{-2}{N_b}\sum_{l=(n-1)N_b}^{nN_b-1}\boldsymbol{u}_l e_l$, being e_l and $\boldsymbol{u}_l$ uncorrelated. Hence, the covariance of the gradient is[5]

$$\begin{aligned}\operatorname{cov}(\boldsymbol{N}_n) &= \frac{4}{N_b^2}E\left[\sum_{k,m}(\boldsymbol{u}_k e_k)(\boldsymbol{u}_m e_m)^T\right] = \frac{4}{N_b^2}E\left[\sum_{k,m}\left(e_k e_m \boldsymbol{u}_k \boldsymbol{u}_m^T\right)\right]\\ &= \frac{4}{N_b^2}\sum_k E\left[e_k^2\right]E\left[\boldsymbol{u}_k\boldsymbol{u}_k^T\right] \approx \frac{4}{N_b}\sigma^2_{\min}\boldsymbol{R}.\end{aligned}$$

Each of the previous steps is justified by the independence assumptions, and the last approximation comes from considering the error $E\left[e_k^2\right]$ when $\boldsymbol{w}_n$ approaches $\boldsymbol{w}^*$ equal to that of the optimum Wiener filter $\sigma^2_{\min}$. Substituting the weight-vector noise $\boldsymbol{v}_n = \boldsymbol{w}_n - \boldsymbol{w}^*$ in (3.32) and developing

$$\begin{aligned}&\boldsymbol{w}_{n+1} = \boldsymbol{w}_n + \frac{\mu N_b}{2}\left(-\boldsymbol{\nabla}_n - \boldsymbol{N}_n\right)\\ \Rightarrow\ &\boldsymbol{v}_{n+1} = \boldsymbol{v}_n + \frac{\mu N_b}{2}\left(-2\boldsymbol{R}\cdot\boldsymbol{v}_n - \boldsymbol{N}_n\right) = \boldsymbol{v}_n\left(\boldsymbol{I} - \mu N_b\boldsymbol{R}\right) - \frac{\mu N_b}{2}\boldsymbol{N}_n.\end{aligned}$$

In steady-state regime, the mean of $\boldsymbol{v}_n$ is zero and its covariance is

$$\begin{aligned}\operatorname{cov}(\boldsymbol{v}_n) &= \left(\boldsymbol{I} - \mu N_b\boldsymbol{R}\right)^2\operatorname{cov}(\boldsymbol{v}_n) + \frac{\mu^2 N_b^2}{4}\operatorname{cov}(\boldsymbol{N}_n)\\ &= \left(\boldsymbol{I} - \mu N_b\boldsymbol{R}\right)^2\operatorname{cov}(\boldsymbol{v}_n) + \mu^2 N_b\sigma^2_{\min}\boldsymbol{R}\end{aligned}$$

$$\begin{aligned}\Rightarrow \left(2\mu N_b\boldsymbol{R} - \mu^2 N_b^2\boldsymbol{R}\boldsymbol{R}\right)\operatorname{cov}(\boldsymbol{v}_n) &= \mu^2 N_b\sigma^2_{\min}\boldsymbol{R}\\ \Rightarrow \left(2\boldsymbol{I} - \mu N_b\boldsymbol{R}\right)\operatorname{cov}(\boldsymbol{v}_n) &= \mu\sigma^2_{\min}\boldsymbol{I}\\ \Rightarrow \operatorname{cov}(\boldsymbol{v}_n) &\approx \frac{\mu}{2}\sigma^2_{\min}\boldsymbol{I}.\end{aligned}$$

The last step neglects $\frac{\mu N_b}{2}\sigma_u^2 \ll 1$. Finally,

$$\begin{aligned}E[(\boldsymbol{w}_n^T\boldsymbol{\alpha}_k)^2] &= E[\boldsymbol{w}_n^T\boldsymbol{w}_n]\sigma^2 = E[(\boldsymbol{w}_n - \boldsymbol{w}^* + \boldsymbol{w}^*)^T(\boldsymbol{w}_n - \boldsymbol{w}^* + \boldsymbol{w}^*)]\sigma^2\\ &= \left(||\boldsymbol{w}^*||^2 + E[\boldsymbol{v}_n^T\boldsymbol{v}_n]\right)\sigma^2 = \left(||\boldsymbol{w}^*||^2 + \operatorname{tr}\left(\operatorname{cov}(\boldsymbol{v}_n)\right)\right)\sigma^2 =\\ &\left(||\boldsymbol{w}^*||^2 + \frac{\mu N_E}{2}\sigma^2_{\min}\right)\sigma^2.\end{aligned}$$

$E[(\boldsymbol{\rho}_n^T\boldsymbol{u}_k)^2]$: For this term, we have that

$$E[(\boldsymbol{\rho}_n^T\boldsymbol{u}_k)^2] = \operatorname{tr}\left\{E\left[\boldsymbol{\rho}_n^T\boldsymbol{\rho}_n\right]\boldsymbol{R}\right\}. \tag{3.33}$$

[5] For the sake of clarity, we will omit the ranges of the indices from now on where there is no ambiguity.

The update equation with finite precision is

$$\begin{aligned}
\boldsymbol{w}'_{n+1} &= \boldsymbol{w}'_n + \mu \sum_k \boldsymbol{u}'_k e'_k + \boldsymbol{\varsigma}_n \\
&= \boldsymbol{w}'_n + \mu \sum_k \boldsymbol{u}_k e_k - \mu \sum_k \boldsymbol{u}_k \boldsymbol{u}_k^T \boldsymbol{\rho}_n - \mu \sum_k \boldsymbol{u}_k \boldsymbol{w}_n^T \boldsymbol{\alpha}_k \\
&+ \mu \sum_k \boldsymbol{u}_k (\beta_k - \eta_k) + \mu \sum_k \boldsymbol{\alpha}_k e_k + \boldsymbol{\varsigma}_n,
\end{aligned} \tag{3.34}$$

where $\boldsymbol{\varsigma}_n$ is the error produced by the quantization in the sum $\mu \sum_k \boldsymbol{u}'_k e'_k$. In the case of the Block LMS, the quantization is only performed after the sum: $Q\left(\mu \sum_k \boldsymbol{u}'_k e'_k\right)$, producing an error of power σ_w^2; for completeness and to cover all the practical cases, we will preserve the same parameter d used for the LMS, with a slightly changed meaning:

$$d = \begin{cases} 1, & \text{if each product of the sum } \mu \sum_k e_k \boldsymbol{u}_k \text{ is individually quantized in (3.4)} \\ 0, & \text{if there is no intermediate quantization in } \mu \sum_k e_k \boldsymbol{u}_k \text{ in (3.4).} \end{cases}$$

From (3.29) and (3.34), the coefficients error vector $\boldsymbol{\rho}$ has the following update equation

$$\boldsymbol{\rho}_{n+1} = \boldsymbol{F}_n \boldsymbol{\rho}_n + \boldsymbol{b}_n, \tag{3.35}$$

with

$$\begin{aligned}
\boldsymbol{F}_n &= \boldsymbol{I} - \mu \sum_k \boldsymbol{u}_k \boldsymbol{u}_k^T, \\
\boldsymbol{b}_n &= \mu \left(\sum_k \boldsymbol{u}_k \boldsymbol{w}_n^T \boldsymbol{\alpha}_k + \sum_k \boldsymbol{u}_k (\beta_k - \eta_k) + \sum_k \boldsymbol{\alpha}_k e_k \right) + \boldsymbol{\varsigma}_k.
\end{aligned}$$

After operating, we obtain

$$\begin{aligned}
E\left[\boldsymbol{\rho}_{n+1} \boldsymbol{\rho}_{n+1}^T\right] &= E\left[\boldsymbol{F}_n \boldsymbol{\rho}_n \boldsymbol{\rho}_n^T \boldsymbol{F}_n^T\right] + \mu^2 \left(E\left[\boldsymbol{w}_n \boldsymbol{w}_n^T\right] N_b \sigma^2 \boldsymbol{R} + N_b \left(\sigma^2 + c\sigma_I^2\right) \boldsymbol{R} + N_b \sigma^2 E[e_n^2] \boldsymbol{I} \right) + \sigma_w^2 \boldsymbol{I} \\
&+ d\left((N_b - 1)\sigma_w^2 \boldsymbol{I} + N_b \sigma_I^2 \boldsymbol{R}\right) \\
&\approx E\left[\boldsymbol{F}_n \rho_n \rho_n^T \boldsymbol{F}_n^T\right] \\
&+ \underbrace{\mu^2 \left(||\boldsymbol{w}^*||^2 N_b \sigma^2 \boldsymbol{R} + N_b \left(\sigma^2 + c\sigma_I^2\right) \boldsymbol{R} + N_b \sigma^2 \sigma_{\min}^2 \boldsymbol{I}\right) + \sigma_w^2 \boldsymbol{I} + d\left((N_b - 1)\sigma_w^2 \boldsymbol{I} + N_b \sigma_I^2 \boldsymbol{R}\right)}_{\boldsymbol{Q}_n},
\end{aligned} \tag{3.36}$$

where the last approximation comes from the steady-state regime assumption. Using the same approximation for the first term as in [54, (A14)] (neglect $\boldsymbol{PRP}$ w.r.t. $\boldsymbol{R}\mathrm{tr}(\boldsymbol{RP}_n)$), and denoting $\boldsymbol{P}_n = E[\boldsymbol{\rho}_n \boldsymbol{\rho}_n^T]$, we get

$$\boldsymbol{P}_{n+1} \approx \boldsymbol{P}_n - \mu N_b \left(\boldsymbol{RP}_n + \boldsymbol{P}_n \boldsymbol{R}\right) + \mu^2 N_b^2 \boldsymbol{R} \mathrm{tr}\left(\boldsymbol{RP}_n\right) + \boldsymbol{Q}_n.$$

In steady-state $\boldsymbol{P}_{n+1} = \boldsymbol{P}_n$, and

$$\mathrm{tr}\left(\boldsymbol{RP}_n\right) = \frac{\mathrm{tr}(\boldsymbol{Q}_n)}{2\mu N_b - \mu^2 N_b^2 \mathrm{tr}(\boldsymbol{R})}. \tag{3.37}$$

Substituting in (3.37) the definition of $\boldsymbol{Q}_n$ (3.36), and the result in (3.33), we obtain

$$E[(\boldsymbol{\rho}_n^T \boldsymbol{u}_k)^2] = \frac{\mu^2 N_b \sigma^2 \left(\left(1 + c\frac{\sigma_I^2}{\sigma^2} + ||\boldsymbol{w}^*||^2 \right) \mathrm{tr}(\boldsymbol{R}) + N_E \sigma_{\min}^2 \right) + N_E \sigma_w^2 + d\left(N_E(N_b-1)\sigma_w^2 + N_b \sigma_I^2 \mathrm{tr}(\boldsymbol{R}) \right)}{2\mu N_b - \mu^2 N_b^2 \mathrm{tr}\boldsymbol{R}}.$$

Due to the independence of $\boldsymbol{\rho}_n$ and data at time n and due to (3.35), the term $-2E[e_n \boldsymbol{\rho}_n^T \boldsymbol{u}_n]$ is zero. Substituting back each of the terms in (3.31), the final expression shown in (3.18) for the MSE in the Block LMS implementation is obtained.

3.C.1. Transient Deviation due to Finite Precision

Following a similar derivation to that in [30], we have extended the theoretical adaptation curve to the BLMS algorithm. The target is to calculate the evolution of the weight vector misadjustment, defined as $\mathcal{M}_n = E[\boldsymbol{\rho}_n^T \boldsymbol{\rho}_n] = \mathrm{tr}(E[\boldsymbol{\rho}_n \boldsymbol{\rho}_n^T])$. Using the same notation as in the previous section, and the same independence assumptions, the quantization error propagated to the prediction error signal is

$$e_n' - e_n = \sum_k \left(\beta_k - \eta_k - \left(\boldsymbol{\rho}_n \boldsymbol{u}_k + \boldsymbol{w}_n^T \boldsymbol{\alpha}_k + \boldsymbol{\rho}_n^T \boldsymbol{\alpha}_k \right) \right).$$

Operating on (3.34) and including all the second order terms, we get

$$\begin{aligned} \boldsymbol{w}_{n+1} + \boldsymbol{\rho}_{n+1} &= \overbrace{\boldsymbol{w}_n + \mu \sum_k \boldsymbol{u}_k e_k}^{\boldsymbol{w}_{n+1}} + \boldsymbol{z}_n + \boldsymbol{B}_n \boldsymbol{\rho}_n \\ \Rightarrow \boldsymbol{\rho}_{n+1} &= \boldsymbol{b}_n + \boldsymbol{F}_n \boldsymbol{\rho}_n, \end{aligned}$$

we redefine $\boldsymbol{F}_n$ and $\boldsymbol{b}_n$ to incorporate the neglected terms in the previous formulation

$$\begin{aligned} \boldsymbol{F}_n &= \boldsymbol{I} - \mu \sum_k \left(\boldsymbol{u}_k \boldsymbol{u}_k^T + \boldsymbol{u}_k \boldsymbol{\alpha}_k^T + \boldsymbol{\alpha}_k \boldsymbol{u}_k^T + \boldsymbol{\alpha}_k \boldsymbol{\alpha}_k^T \right) \\ \boldsymbol{b}_n &= \varsigma_n + \mu \sum_k \left(\left(\beta_k - \eta_k - \boldsymbol{w}_n^T \boldsymbol{\alpha}_k \right) \cdot (\boldsymbol{u}_k + \boldsymbol{\alpha}_k) + e_k \boldsymbol{\alpha}_k \right). \end{aligned} \tag{3.38}$$

Since the errors are assumed to be uncorrelated,

$$\boldsymbol{P}_{n+1} = E[\boldsymbol{\rho}_{n+1} \boldsymbol{\rho}_{n+1}^T] = E[\boldsymbol{F}_n \boldsymbol{\rho}_n \boldsymbol{\rho}_n^T \boldsymbol{F}_n] + E[\boldsymbol{b}_n \boldsymbol{b}_n^T]. \tag{3.39}$$

For the first term, splitting $\boldsymbol{F}_n = \underbrace{\boldsymbol{I} - \mu \sum_k \boldsymbol{u}_k \boldsymbol{u}_k^T}_{\boldsymbol{F}_n^{(1)}} \underbrace{- \mu \sum_k \left(\boldsymbol{u}_k \boldsymbol{\alpha}_k^T + \boldsymbol{\alpha}_k \boldsymbol{u}_k^T + \boldsymbol{\alpha}_k \boldsymbol{\alpha}_k^T \right)}_{\boldsymbol{F}_n^{(2)}}$, and developing each of the terms of the product

$E\left[\left(\boldsymbol{F}_n^{(1)}+\boldsymbol{F}_n^{(2)}\right)\boldsymbol{\rho}_n\boldsymbol{\rho}_n^T\left(\boldsymbol{F}_n^{(1)}+\boldsymbol{F}_n^{(2)}\right)\right]$, assuming i.i.d. $\boldsymbol{u}_n$ (i.e., the autocorrelation matrix is diagonal and its eigenvalues matrix is $\boldsymbol{\Lambda}=\boldsymbol{R}$), neglecting $\sigma^4 \ll \sigma^2$, we obtain

$$E[\boldsymbol{F}_n\boldsymbol{\rho}_n\boldsymbol{\rho}_n^T\boldsymbol{F}_n] = \left((\boldsymbol{I}-\mu N_b\boldsymbol{\Lambda})^2 - 2\mu N_b\sigma^2\left(\boldsymbol{I}-\mu N_b\boldsymbol{\Lambda}\right) + 4\mu^2 N_b\sigma^2\boldsymbol{\Lambda}\right)\boldsymbol{P}_n. \quad (3.40)$$

For the second term, $\boldsymbol{b}_n$ can also be split into two uncorrelated terms (due to the errors being uncorrelated and zero-mean)

$$\boldsymbol{b}_n = \overbrace{\mu\sum_k\left(e_k\boldsymbol{\alpha}_k + (\beta_k-\eta_k)\,\boldsymbol{u}_k - \boldsymbol{u}_k\boldsymbol{w}_n^T\boldsymbol{\alpha}_k - \boldsymbol{\alpha}_k\boldsymbol{\alpha}_k^T\boldsymbol{w}_n\right)}^{\boldsymbol{b}_n^{(1)}} + \overbrace{\boldsymbol{\varsigma}_n + \mu\sum_k\left((\beta_k-\eta_k)\,\boldsymbol{\alpha}_k\right)}^{\boldsymbol{b}_n^{(2)}}.$$

Hence, the second term, neglecting $\sigma^2(\sigma^2+\sigma_I^2)$, is

$$E\left[\boldsymbol{b}_n\boldsymbol{b}_n^T\right] \approx \left(\mu^2 N_b\sigma^2 E[e_n^2] + \sigma_w^2\right)\boldsymbol{I} + \mu^2 N_b\left(\sigma^2||\boldsymbol{w}_n^2|| + \sigma^2 + \sigma_I^2\right)\boldsymbol{\Lambda}. \quad (3.41)$$

Substituting (3.40) and (3.41) in (3.39),

$$\begin{aligned}\boldsymbol{P}_{n+1} = &\left((\boldsymbol{I}-\mu N_b\boldsymbol{\Lambda})^2 - 2\mu N_b\sigma^2\left(\boldsymbol{I}-\mu N_b\boldsymbol{\Lambda}\right) + 4\mu^2 N_b\sigma^2\boldsymbol{\Lambda}\right)\boldsymbol{P}_n \\ &+ \left(\mu^2 N_b\sigma^2 E[e_n^2] + \sigma_w^2\right)\boldsymbol{I} + \mu^2 N_b\left(\sigma^2||\boldsymbol{w}_n||^2 + \sigma^2 + \sigma_I^2\right)\boldsymbol{\Lambda}. \quad (3.42)\end{aligned}$$

Neglecting the second order effects of the gradient noise, and taking into account that for BLMS, the update matrix for the error given by the direct-averaging method [118] is $E[\boldsymbol{I}-\mu\sum_k\boldsymbol{u}_k\boldsymbol{u}_k^T] = \boldsymbol{I}-\mu N_b\boldsymbol{R}$, the functions $E[e_n^2]$ and $||\boldsymbol{w}_n||^2$ can be respectively approximated by

$$E[e_n^2] \approx \sigma_{\min}^2 + \sum_{k=0}^{N_E-1}\lambda_k w_k^{*2}\cdot(1-\mu N_b\lambda_k)^{2n} \quad (3.43)$$

$$||\boldsymbol{w}_n||^2 \approx \sum_{k=0}^{N_E-1} w_k^{*2}\cdot\left(1-(1-\mu N_b\lambda_k)^n\right)^2, \quad (3.44)$$

being w_k^* the kth component of the optimum Wiener filter, and λ_k the kth eigenvalue of $\boldsymbol{R}$; for an i.i.d. $\boldsymbol{u}_n$, $\lambda_k=\sigma_u^2$, $k=\{0,\ldots,N_E-1\}$. Taking this into account, substituting (3.43) and (3.44) in (3.42), and neglecting $\mu\sigma^2 \ll 1$, we have

$$\mathcal{M}_{n+1} = \gamma^2\mathcal{M}_n + \mu^2 N_b N_E\left(\overbrace{2\sigma^2\sigma_u^2||\boldsymbol{w}^*||^2}^{A}\left(\gamma^{2n}-\gamma^n\right) + \overbrace{\sigma_u^2\left(\sigma^2(1+||\boldsymbol{w}^*||^2) + c\sigma_I^2\right) + \sigma^2\sigma_{\min}^2}^{B}\right) + N_E\sigma_w^2, \quad (3.45)$$

with $\gamma = 1-\mu N_b\sigma_u^2$. Finally, solving the difference equation, Expression (3.19) follows.

3.D. Secure Comparison Protocol

For our purpose of comparing two numbers in the asymmetric case where $\mathcal{A}$ can both encrypt and decrypt with an additively homomorphic cryptosystem and $\mathcal{B}$ can only encrypt, both parties have their respective integer quantities x_A and x_B in the clear. In a generic case, we will have to resort to the protocol in [77] or the one in [173]. But for the cases we deal with, we have assured that $0 \leq x_A, x_B < 2^l$ (x_A and x_B are l-bit numbers). Both parties want to compute an encryption of $[x_A > x_B]$ (the other three possible comparisons, $<, \geq, \leq$ can be straightforwardly obtained with trivial changes to the presented protocol).

We are in the case where both parties can obtain the binary representation of their respective numbers, so $\mathcal{A}$ obtains the encryption of $[\![-x_A]\!]_b$ with l bits (in two's complement), and sends the l encryptions to $\mathcal{B}$. $\mathcal{B}$ has to execute a binary adder circuit with the binary representation of x_B and the received encrypted binary representation of $[\![-x_A]\!]_b$. The carry bit of this circuit gives the result of the comparison. At the end of the protocol, $\mathcal{B}$ has the encryption of $[\![x_A > x_B]\!]$.

We avoid the use of garbled circuits, that would increase the complexity of the protocol, and we rely on an additive homomorphism of the used cryptosystem (or secret sharing scheme), so that this circuit can be implemented in l rounds with the following complexity:

$$\begin{aligned}
\mathrm{Cpx}_{cm,COMP}(l) =& l|E| + (l-1)\mathrm{Cpx}_{cm,MULT},\\
\mathrm{Cpx}_{cp,COMP,\mathcal{A}}(l) =& (l-1)\mathrm{Cpx}_{cp,MULT,\mathcal{A}} + l\mathrm{Cpx}_{EncBit},\\
\mathrm{Cpx}_{cp,COMP,\mathcal{B}}(l) =& 2\mathrm{Cpx}_{EP} + 2\mathrm{Cpx}_{EA} + (l-1)(\mathrm{Cpx}_{cp,MULT,\mathcal{B}} + 3\mathrm{Cpx}_{EP}\\
&+ 5\mathrm{Cpx}_{EA}).
\end{aligned}$$

$|E|$ represents the number of bits of an encryption (or share); the subindex *cm* stands for communication complexity, and *cp* for computational complexity for party $\mathcal{A}$ or $\mathcal{B}$, being $\mathrm{Cpx}_{xx,MULT}$ the corresponding complexity of the interactive multiplication protocol; Cpx_{EA}, Cpx_{EP} respectively denote the computational complexity of a homomorphic addition and product (by a known scalar) for the used cryptosystem (or secret sharing scheme), and Cpx_{EncBit} is the computational complexity for encrypting (sharing) a bit.

We are not imposing any limitation to the number of rounds, as we are looking for the lowest computation complexity, but in case of constant round protocols, we would have to resort to the Prefix-OR protocol in [77], obtaining a larger computation complexity which, in any case, is in the same order $O(l)$ as the previously exposed protocol.

Chapter 4

Applications

There are many application scenarios where SPED can be used to protect the privacy of the users whose sensitive signals are being processed. This chapter is focused on a set of application scenarios for which privacy-preserving solutions based on SPED are presented. These scenarios are Secure Watermark Detection with symmetric key, Cloud Computing and, in particular, Medical Clouds, and, as a specific medical application, private approximate searches on DNA (Desoxyribo-Nucleic Acid) strings, that convey the most sensitive information about an individual.

The work shown in this chapter has been partially presented at ACM MMSEC'06 [222], SPIE'07 [223], EURASIP Journal on Information Security [224], CISE 2010 [225], CLOSER 2011 [192], VPH 2010 [226] and CCS 2007 [221]; some of the technical developments have been filed as international patent applications (Patent pending, PCT Application No. PCT/IB2008/051771).

4.1. Introduction

There are multiple privacy-sensitive signal processing applications where the primitives presented in Chapter 2 and other SPED primitives may be used to fulfill the privacy requirements. This chapter firstly explores several of these applications, that are briefly sketched in the following classification, exemplifying the use of the solutions from Chapter 2.

- *Biometrics*: The most evident application of biometrics is authentication. Here, the server has information regarding the biometrics of a person. Due to its fuzziness, the region of acceptance may be modeled as a convex polytope in the features space. The user presents her features as a feature vector, and both parties may run the presented secure point inclusion protocol (cf. Section 2.3) for determining the correctness of the user's claimed identity. In this process, the biometric features of the client are protected from the server, and the region of acceptance is not disclosed to the user. Furthermore, the whole interaction consists of encrypted values, thereby protecting the information against an eavesdropper.

Comparing this method with the typical *Helper Data Systems* [232] employed in biometric authentication, the complexity of the proposed protocol is higher, but its main advantage is its flexibility, as it allows to perform fine-grained adjustments of the detection boundary.

- *Classification*: The point inclusion problem with a convex polytope can be regarded as a classification problem. In this case, the spatial region is interpreted as a fusion of linear classifiers, each one represented by one of the hyperplanes that form the polytope boundary. Thus, the protocol of Section 2.3 implements a secure classifier. The case of hyperellipsoids corresponds to a one-layer RBF (Radial Basis Function) network with threshold activation function.

- *Database queries*: The developed point inclusion protocol can also find an application in non-orthogonal database queries, where a query, represented as a convex region in the measurable terms space, is matched with an entry, represented by a vector of terms. In this case, the query is not revealed to the database server, and the server can keep the entries secret until they match a query.

- *Positioning*: If the point inclusion protocol is restricted to two or three dimensions, it can be applied to the problem of secure positioning. Here, a party wants to check whether one particular location is inside a region whose definition is owned by another party, but neither of them want to disclose their own data to the other party. The work in [197] is a typical example of a secure positioning application in a pervasive sensor network, where a user wants to know if his current position is being sensed, but the monitoring party does not want to disclose the sensing area.

- *Watermarking/Fingerprinting*: Classic symmetric watermarking and fingerprinting schemes require disclosure of the embedding key during detection. In case the party performing watermark detection is malicious, it can use the key to remove a watermark [181]. Thus, traditional symmetric watermark detectors are not applicable in this case. The secure point inclusion protocol can be applied in a secure watermark detector, where the detection region is a convex polytope in a multidimensional space. This makes it possible to run the detection protocol without disclosing either the detection region to the party that presents the possibly watermarked work, nor this work to the party that owns the description of the detection region. Next in the chapter, a novel solution for secure watermark detection without the use of the point inclusion protocol is explored.

- *Secure Approximate Searching and Matching*: This problem can be related to several error metrics, but it is commonly associated with the Edit or Levenshtein distance [144], that is the used metric in the developed oblivious automaton execution protocol (cf. Section 2.5). This distance measure accounts for three types of errors, namely symbol substitutions, deletions and insertions. Given two strings $\boldsymbol{x}$ and $\boldsymbol{y}$, the Edit distance is defined as the minimum number of Edit errors that $\boldsymbol{x}$ must undergo in order to be transformed into $\boldsymbol{y}$. If this number is below a given threshold, both sequences are said to approximately match; in case of a match, a sequence alignment can be computed, which associates the symbols of $\boldsymbol{x}$ and $\boldsymbol{y}$, up to insertions and deletions. *Approximate string searching* deals with a short sequence $\boldsymbol{x}$ (the pattern) searched in a longer sequence $\boldsymbol{y}$, while tolerating Edit errors. This is also applicable to DNA Sequences, as will be shown in this chapter.

- *Regular expression matching*: As the protocol presented in Section 2.5 allows the efficient privacy-preserving execution of an automaton, it can be also applied to any problem with a need of privacy preservation that can be stated in terms of a regular expression. There are plenty of applications where regular expressions are commonly used, like *password format validation* or *data parsing.* In general, a regular expression can indicate the format that a given text must conform to in order to be considered valid, and this is normally the first step of a validation process that protects the validator from entries that are out of domain and would likely cause errors. Whatever the validated information is, the need for privacy in the validator extends also to the need of privacy for the format checker.

- *Secure file parsing*: another application of regular expressions, where some text is erased, substituted or inserted in some parts of the file; this can be done through the application of a finite automaton with output. When security is a concern the input text has to be protected, and the presented protocol may be applied. A specific case of the above is word or pattern finding in a document, a commonly used technique in spam checkers for electronic mail or virus analyzers. When dealing with confidential mails or private software, they must be protected from the party that runs the checker or analyzer. The application of the protocol for these scenarios is straightforward.

- As for sequential transducers, they represent an efficient approach for large-scale dictionaries [164, 165], used for *computational linguistics*, in *lexical analysis*, *morphology* and *phonology*, *syntax*, *text-to-speech synthesis*, or *speech recognition*. All these applications can also be handled by the secure automata execution protocol presented in Section 2.5, when there is the need of protecting the recognized sequence.

- *Linear systems*: The resolution of systems of linear equations is the nucleus of many signal processing applications, from common optimization problems to many other more complex *data mining* systems that perform statistical calculation on private data, like channel equalization, maximum likelihood detection, beamforming, systems and control, etc.

In the next sections, we develop several specific applications with detailed privacy-preserving solutions described in depth, namely:

- *Digital Watermarking Security* (Section 4.2): we present an efficient secure watermark detector that does not reveal the secret key to the party that runs it, achieving an improved robustness against sensitivity attacks.
- *Cloud Computing* (Section 4.3): we provide a conceptual approach to a high-level architecture for private processing in Cloud Computing, exemplified in Medical Clouds.
- *Privacy in Biological Signals* (Section 4.4): as a specific medical application, we present a privacy-preserving system for detecting DNA diseases using the oblivious automata execution protocol of Section 2.5.

4.2. Digital Watermarking Security

This section reviews the topic of watermark security and the basics of key protection and sensitivity attacks in blind watermark detection; two previous detectors and one novel proposed detector are compared in terms of robustness against sensitivity attacks, showing how Zero-Knowledge detection fits in this scenario. Subsection 4.2.5 details the novel detection protocol and an improved version, giving also a complete security (Subsection 4.2.6) and complexity (Subsection 4.2.7) analysis.

Watermarking technology [37, 88] has emerged as a solution for authorship proofs or dispute resolving. In these applications, there are several requirements that watermarking schemes must fulfill, like imperceptibility, robustness to attacks that try to erase a legally inserted watermark or to embed an illegal watermark in some asset, and they must also be secure to the disclosure of information that could allow the breakage of the whole system by unauthorized parties. These three factors and their interplay have been a matter of discussion and research for many years within the watermarking community. In this section we are concerned with watermarking security; good surveys on this topic can be found in [181, 97]. There are numerous alternative countermeasures for protecting the security of watermarking systems, or, more specifically, for protecting the security of the secret key used to generate and embed the watermark on the host assets. While some of these countermeasures are targeted at making more difficult the estimation of this key from several observations of marked (and/or unmarked) assets, there is another important group of countermeasures whose primary objective is to protect the secret key.

Up to now it has been shown that the most sensitive part of watermarking schemes is the embedding key; once this key is disclosed, the whole system is compromised, so the less information about this key the watermarking scheme leaks, the better for security. Nevertheless, symmetric schemes [37] use the same embedding key also for detection/decoding of the inserted watermark, and this represents a security hole. There are two approaches for protecting the embedding key during the detection/decoding process, namely *asymmetric watermarking* and *zero-knowledge watermarking.*

4.2.1. Asymmetric Watermarking

The goal of **asymmetric schemes** is to make the process of detection/decoding independent of the embedding, by using different keys in these two steps. Although sometimes the terms public-key and asymmetric watermarking are used indistinctly, they have a different meaning, pointed out in most of the works in this area

- *Asymmetric watermarking*: The keys used for embedding and for extraction are different.
- *Public-key watermarking*: The key used for extraction (public key) holds enough information to accomplish the detection/decoding, while not allowing to remove the watermark or forge illegal contents if the key used for embedding (private key) is kept secret.

Currently, there is no truly public-key watermarking method, although many efforts have been done in order to achieve an asymmetric scheme that fulfils also the requirements of public-key watermarking. In [163], Miller states that key asymmetry is not sufficient to achieve a valid public-key scheme, and he wonders whether it would even be necessary if some scheme applicable in an open-cards scenario existed. In fact, the presented asymmetric schemes up to date are not really public-key, while their improved security when not publishing any keys comes from the higher complexity of the watermarking regions [38], leading to better security when increasing the order of the detection function [98, 126].

4.2.2. Zero-Knowledge Watermarking

Zero-knowledge watermark has arisen as a solution to conceal all the security parameters needed for detection/decoding in symmetric schemes. This way, when using a zero-knowledge watermarking protocol between two parties (Prover and Verifier), only the fact that a watermark is present or absent is disclosed to the Verifier, but all the security parameters remain secret. This solves the problem posed by tampering attacks (cf. [181]), and provides a better protection against sensitivity attacks (cf. [181]), as only blind attacks may succeed.

The concept of zero-knowledge was introduced by Goldwasser *et al.* [112] in 1985. It basically consists in convincing an adversary of an assertion without giving him any knowledge but the assertion whose validity is proven. Zero-knowledge protocols are widely used in cryptography, generally to force a malicious adversary to behave as stated by a determined protocol.

These protocols are based on interactive proofs [112, 113] and arguments [48], and especially on proofs of knowledge [162]. All of them are based on the intuitive notion that it is easier to prove a statement through an interaction between both parties (Prover and Verifier), than to write a proof that can be verified by any party without interaction. The concealment of data involved in this interaction is measured in terms of knowledge complexity [110], related to the similarity between random sequences and the sequences produced by the interaction. Zero-knowledge is the result of the indistinguishability of both types of sequences.

The first attempt of application of zero-knowledge to watermark detection was undertaken by Gopalakrishnan [116]; it consists in a protocol that allows to detect an encrypted watermark in an encrypted image, through the use of RSA [191]. Later, Craver [72] proposed several schemes of watermark detection with minimal disclosure, based on permutations using Pitas's scheme [185], or ambiguity attacks to generate a set of watermarks indistinguishable from the real one.

Adelsbach *et al.* [22] proved afterwards that all the preceding works had some flaws that made them non zero-knowledge, as they give information about the embedded watermark when using the detector as an oracle.

The formalization of zero-knowledge watermark detection was given by Adelsbach and Sadeghi [26]; they proposed the use of commitment schemes [76, 198] for concealing the secret parameters of the detector; also in this work, they presented a truly zero-knowledge detection protocol for Cox's additive spread spectrum watermarking algorithm [68], as a high level protocol that uses existing zero-knowledge proofs as subblocks; it benefits from the homomorphic properties of some commitment schemes [96, 78] for alleviating the communication complexity. Following the same philosophy, Piva *et al.* [187] also presented a zero-knowledge detection protocol for ST-DM.

Nevertheless, there are some security issues that must be taken into account when developing zero-knowledge watermarking protocols; they have been pointed out by Katzenbeisser in [137], and are mainly related to the correct concealing of protocol inputs and the problem of guaranteeing the correct generation of a concealed watermark. To overcome the latter issue, Adelsbach *et al.* [23] proposed several new zero-knowledge protocols that can be used to prove that a given sequence follows a determined probability distribution.

Although zero-knowledge protocols could seem an utopical solution to many security problems, they have advantages and also drawbacks [159]. Their main advantages are their null security degradation when used several times, and their resistance against clear-text attacks; their main drawback is their efficiency, as they commonly produce communication and complexity overheads that are much bigger than those presented by public-key protocols; as an example, a complete complexity study of the zero-knowledge version of Cox's non-blind detection scheme [68] is developed in [25]. Moreover, many techniques that are based on zero-knowledge lack a formal proof of zero-knowledge or even validity, due to the choices of parameters to improve efficiency; actually, many of the concepts related to zero-knowledge are asymptotic and cannot be directly applied to practical protocols.

4.2.3. Sensitivity Attacks

The watermarking schemes that have been used up to now are symmetric, as they employ the same key for watermark embedding and watermark detection; as pointed out in the previous section, the fact that the secret key must be given to the party that runs the detector, which in most cases is not trusted, constitutes a security hole that can be tackled through the use of Zero Knowledge watermark detection, like in [26], where a Prover $\mathcal{P}$ tries to demonstrate to a Verifier $\mathcal{V}$ the presence of a watermark in a given asset.

Nevertheless, such minimum disclosure of information still allows for blind sensitivity attacks [64], that have arisen as very harmful attacks for methods that present simple detection boundaries. The ZK detection protocols presented to date—Adelsbach and Sadeghi [26] and Piva *et al.* [186]—are based on correlation detectors, for which blind sensitivity attacks are especially efficient.

Hence, this section presents a novel zero-knowledge blind watermark detection protocol based on the spread spectrum detector by Hernández *et al.* [121], which is optimal for additive watermarking in generalized Gaussian distributed host features (e.g. AC DCT coefficients of images). The robustness to sensitivity attacks comes from the complexity of the detection boundary for certain shape factors. Thus, when combined with zero-knowledge, it becomes secure and robust. This protocol will be compared in terms of performance and efficiency with the previous ZK protocols based on additive spread-spectrum and Spread-Transform Dither Modulation (ST-DM), and rewritten in a form that greatly improves its communication and computation complexity.

4.2.4. Blind Watermark Detection

Some of the concepts involved in Blind Digital Watermarking Detection, needed for the development of the studied protocols, are briefly introduced before describing the protocols in the next subsection.

Given a host signal $\mathbf{x}$, a watermark $\mathbf{w}$ and a pair of keys $\{K_{\text{emb}}, K_{\text{det}}\}$ for embedding and detection (they are the same key in symmetric schemes), a digital blind watermark detection scheme consists of an *embedder* that outputs the watermarked signal $\mathbf{y} = \text{Embed}(\mathbf{x}, \mathbf{w}, K_{\text{emb}})$; and a *detector*, that given a possibly attacked signal $\mathbf{z} = \mathbf{y} + \mathbf{n}$, where $\mathbf{n}$ represents added noise, the watermark $\mathbf{w}$ and the detection key K_{det}, outputs a Boolean value indicating whether the signal $\mathbf{z}$ contains or not the watermark $\mathbf{w}$, without using the original host data $\mathbf{x}$.

Three detection algorithms will be compared in terms of their Receiver Operating Characteristic (ROC), namely Additive Spread Spectrum with a correlation based detector (SS), Spread-Transform Dither Modulation without distortion compensation (ST-DM), and Additive Spread Spectrum with a Generalized Gaussian maximum likelihood (ML) detector (GG). In all of them, the host features $\mathbf{x}$ are considered i.i.d. with variance σ_X^2, the watermarked features are denoted by $\mathbf{y} = \mathbf{x} + \mathbf{w}$, and $\mathbf{z}$ represents the input to the receiver, which may be corrupted with AWGN noise $\mathbf{n}$, that is considered also i.i.d with variance σ_N^2. The binary hypothesis test that must be solved at the detector is:

$$\mathcal{H}_0 : \mathbf{z} = \mathbf{x} + \mathbf{n}, \quad \mathcal{H}_1 : \mathbf{z} = \mathbf{x} + \mathbf{w} + \mathbf{n}.$$

Table 4.1 summarizes the Probabilities of false alarm (P_f) and missed detection (P_m) for the three detectors [37, 180, 182].

Table 4.1: Probabilities of false alarm (P_f) and missed detection (P_m) for the three studied detectors.

	AddSS	**ST-DM**	**GG**
P_f	$Q\left(\frac{\sqrt{L}\eta}{\sqrt{\sigma_X^2+\sigma_N^2}}\right)$	$\sum_{i=-\infty}^{\infty}\left[Q\left(\frac{\Delta(i+1/2)-\eta}{\sqrt{L(\sigma_X^2+\sigma_N^2)}}\right) - Q\left(\frac{\Delta(i+1/2)+\eta}{\sqrt{L(\sigma_X^2+\sigma_N^2)}}\right)\right]$	$Q\left(\frac{\eta+m_1}{\sigma_1}\right)$
P_m	$Q\left(\frac{\sqrt{L}(\alpha-\eta)}{\sqrt{\sigma_X^2+\sigma_N^2}}\right)$	$1-\sum_{i=-\infty}^{\infty}\left[Q\left(\frac{i\Delta-\eta}{\sqrt{L}\sigma_N}\right) - Q\left(\frac{i\Delta+\eta}{\sqrt{L}\sigma_N}\right)\right]$	$1-Q\left(\frac{\eta-m_1}{\sigma_1}\right)$

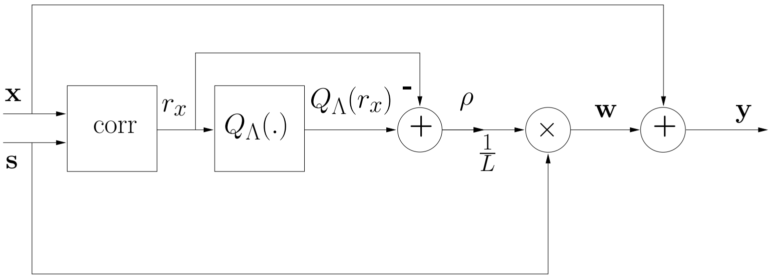


Figure 4.1: Block Diagram of the watermark embedding process for ST-DM

4.2.4.1. Additive Spread Spectrum with correlation-based Detector

In SS, the watermark is generated as the product of a pseudorandom vector $\mathbf{s}$, that we will consider a binary sequence with values $\{\pm 1\}$ (with norm $||s||^2 = L$) and a perceptual mask α (that is assumed to be constant to simplify the analysis), that controls the trade-off between imperceptibility and distortion ($D_w = \frac{1}{L}\sum_{k=1}^{L} E\{w_k^2\} = E\{\alpha_k^2\} = \alpha^2$).

The maximum-likelihood detector for Gaussian distributed host features is a correlation-based detector:

$$r_z = \frac{1}{L}\sum_{k=1}^{L} z_k s_k \overset{\mathcal{H}_1}{\underset{\mathcal{H}_0}{\gtrless}} \eta,$$

where η is a threshold that depends on the probabilities of false alarm (P_f) and missed detection (P_m), as indicated in Table 4.1.

4.2.4.2. Spread Transform Dither Modulation

Given the host features $\mathbf{x}$ and the secret spreading sequence $\mathbf{s}$, which will be considered here binary with values $\{\pm 1\}$, the embedding of the Watermark in ST-DM [59] (similar to Quantized Projection QP [182, 180]) is done as indicated in Figure 4.1.

The host features $\mathbf{x}$ are correlated with the projection signal $\mathbf{s}$, and the result (r_x) is quantized with an Euclidean scalar quantizer $Q_\Lambda(.)$ of step Δ, that controls the distortion, and with centroids defined by the shifted lattice $\Lambda \triangleq \Delta\mathbb{Z} + \Delta/2$. Let $\rho = (Q_\Lambda(r_x) - r_x)$; then, the watermarked vector is given by

$$\mathbf{y} = \mathbf{x} + \mathbf{w} = \mathbf{x} + \frac{1}{L}\rho\mathbf{s}.$$

In order to detect the watermark, the host features, possibly degraded by AWGN noise $\mathbf{n}$, are correlated with the spreading sequence $\mathbf{s}$, and the resulting

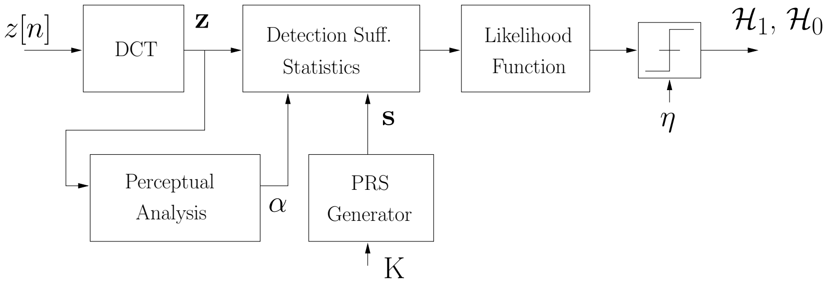


Figure 4.2: Block Diagram of the watermark detection process for the GG detector.

value $r_z = \sum_{k=1}^{L} z_k s_k$ is quantized and compared to a threshold η to determine whether the watermark is present:

$$|Q_\Lambda(r_z) - r_z| \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\lesseqgtr}} \eta.$$

Due to the Central Limit Theorem (CLT), the computed correlations can be accurately modeled by a Gaussian pdf.

4.2.4.3. Additive Spread Spectrum with Generalized-Gaussian Features

Figure 4.2 shows the detection scheme for this case. The host features are assumed to be the DCT coefficients of an image, what justifies the Generalized Gaussian model with the following pdf:

$$f_X(x) = Ae^{-|\beta x|^c}, \quad \beta = \frac{1}{\sigma}\left(\frac{\Gamma(3/c)}{\Gamma(1/c)}\right)^{1/2}, \quad A = \frac{\beta c}{2\Gamma(1/c)}.$$

The embedding procedure is the same as the one described for SS. For detection, a preliminary perceptual analysis provides the estimation of the perceptual mask α that modulates the inserted secret sequence $\mathbf{s}$. The parameters c and β are also estimated from the received features. The likelihood function for detection is

$$l(\mathbf{y}) = \sum_k \beta^c \left(|Y_k|^c - |Y_k - \alpha_k s_k|^c\right) \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \eta, \tag{4.1}$$

where η represents the threshold value used to make the decision.

As shown in [121], the pdf's of $l(Y)$ conditioned to hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$ are approximately Gaussian with the same variance σ_1^2, and respective means $-m_1$ and m_1, that can be estimated from the watermarked image [121].

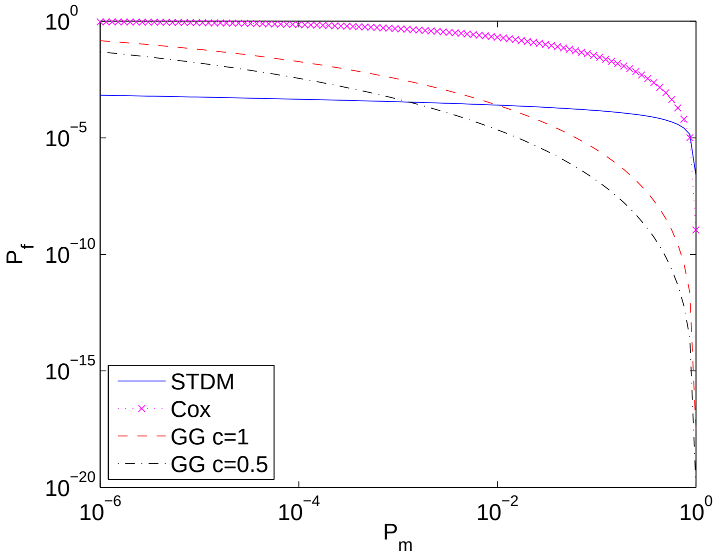


Figure 4.3: Theoretical ROC curves for the studied detectors under AWGN attacks, with DWR=20 dB, WNR=0 dB, L=1000, and Generalized Gaussian distributed host features with c=0.8.

4.2.4.4. Comparison

The three detectors can be compared in terms of robustness through their *Receiver Operating Characteristic* (ROC), taken from the formulas in Table 4.1. The correlation-based detector is only optimum when $c = 2$, and when $c \neq 2$ the Generalized Gaussian detector outperforms it; ST-DM can outperform both for a sufficiently high DWR (Data to Watermark Ratio, DWR $= 10\log_{10}(\sigma_X^2/\sigma_W^2)$), due to its host rejection capabilities. However, the performance of the Generalized Gaussian detector and the ST-DM one are not much far apart when c is near 1 and the DWR in the projected domain (DWR$_p$ = DWR$-10\log_{10} L$) is low. Figure 4.3 shows a plot of the ROC for fixed DWR and WNR (Watermark to Noise Ratio, WNR $= 10\log_{10}(\sigma_W^2/\sigma_N^2)$), with a features shape parameter of $c = 0.8$, that has been chosen as an example of a relatively common value for the distribution of AC DCT coefficients of most images. It is remarkable that even when the exact c is not used, and it is below 1, the performance of the GG detector with $c = 0.5$ is much better than that of the correlation-based one, and its ROC remains near the ST-DM ROC.

Regarding the resilience against sensitivity attacks, it can be shown that the correlation-based detector and the ST-DM one make the watermarking scheme very easy to break when the attacker has access to the output of the detector, as the detection boundaries for both methods are just hyperplanes; Figure 4.4 shows the two-dimensional detection regions for each of the three methods. On the other hand, the detection function in the GG detector when $c < 1$ (Figure 4.4c) presents

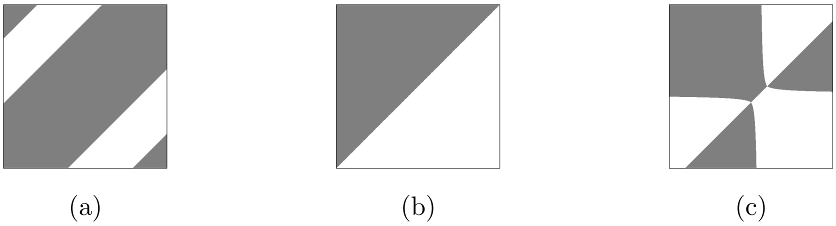


Figure 4.4: Two-dimensional detection boundaries for ST-DM (a), correlation-based detector (b) and GG detector (c).

the property that component-wise modifications produce bounded increments; that is, when modifying one component of the host signal Y, the increment produced in the likelihood function (Eq. (4.1)) is bounded by $|\alpha_k s_k|^c$ independently of the component $|Y_k|$ if $c < 1$:

$$||Y_k|^c - |Y_k - \alpha_k s_k|^c| \leq |\alpha_k s_k|^c.$$

This means that it is not possible to get a signal in the boundary by modifying a single component (or a number N of components such that $\sum_N |\alpha_k s_k|^c$ is less than the gap to η), opposed to a correlation detector, in which just making one component big (or small) enough can get the signal out of the detection region. This property can make very difficult the task of finding a vector in the boundary given only one marked signal.

In order to quantitatively compare the resilience of the three detectors against sensitivity attacks, we will take as robustness criterion the number of calls to the detector needed for reaching an attack distortion equal to that of the watermark (NWR=0 dB). This choice is supported by the fact that for an initially non-marked host $\boldsymbol{x}$ in which a watermark $\boldsymbol{w}$ has been inserted, yielding $\boldsymbol{y}$, it is always possible to find a vector $\boldsymbol{z}$ in the boundary whose distortion with respect to $\boldsymbol{y}$ is less than the power of the watermark (e.g., taking the intersection between the detection boundary and the line that connects $\boldsymbol{x}$ and $\boldsymbol{y}$). Thus, a sensitivity attack can always reach a point with NWR=0 dB. In general, it is not guaranteed that an attack can reach a lower NWR. Furthermore, given that for a blind detection the original non-marked host is not known, imposing a more restrictive fidelity criterion for the attacker than for the embedder makes no sense. In light of the previous discussion, we can consider that a watermark has been effectively erased when a point $\boldsymbol{z}$ is found, whose distortion with respect to $\boldsymbol{y}$ is equal to the power of the embedded watermark $\boldsymbol{w}$; the number of iterations that a sensitivity attack needs to reach this point can thus be used for determining the robustness of the detector against the attack.

We have taken BNSA (Blind Newton Sensitivity Attack [64]; an *RRP*-compliant description of BNSA can be found in [65]) as a powerful representative

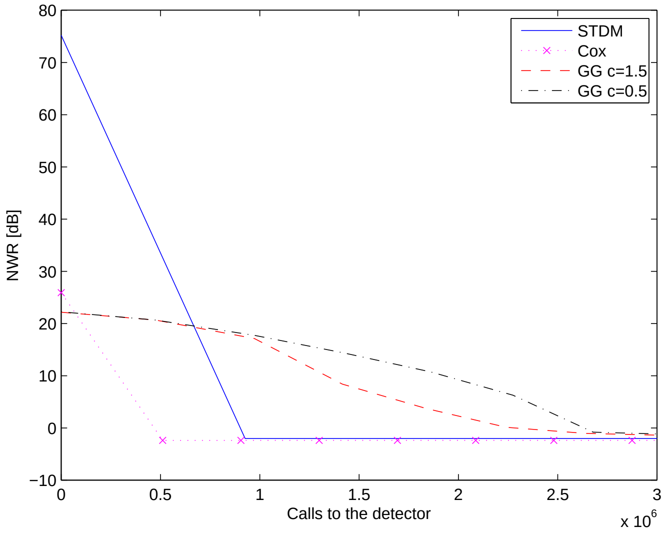


Figure 4.5: NWR for a sensitivity attack (BNSA) as a function of number of calls to the detector for Correlation Detector (Cox), ST-DM and Generalized Gaussian (GG) with $c = 0.5$ and $c = 1.5$ for DWR= 16 dB, $P_f = 10^{-4}$ and $L = 8192$.

of sensitivity attacks, and simulated its execution against the three studied detectors. Each iteration of this algorithm calls the detector a number of times proportional to the number of dimensions of the involved signals. The results show that both ST-DM and the correlation detector are completely broken in just one iteration of the algorithm, independently of the dimensionality of the signals, so the attack needs $\mathcal{O}(L)$ calls to the detector in order to succeed (achieving not only a point with NWR<0 dB, but also convergence to the nearest point in the boundary). This is due to their simple detection boundaries, that have a constant gradient. Figure 4.5 shows the NWR of the attack as a function of the number of calls to the detector, for the three detectors, using DWR=16 dB and $P_f = 10^{-4}$, as a result of averaging 100 random executions. The GG detector is used with two different shape factors, $c = 0.5$ and $c = 1.5$; the number of iterations needed to break the detector in both cases is bigger than for the correlation detectors, due to the more involved detection boundary, but this effect is more evident when $c < 1$, case in which the detector has the aforementioned property of bounded increments for component-wise modifications at the input.

The involved detection boundary of the Generalized Gaussian ML detector makes the number of iterations needed for achieving convergence grow also with the dimensionality of the host. This means that the number of calls to the detector needed to get a certain target distortion is not only higher for the GG detector, but it also grows faster than for the other detectors with the dimensionality of the host (Figure 4.6) for fixed WNR and P_f. We have found empirically that the number of calls needed for reaching NWR=0 dB is approximately $\mathcal{O}(L^{1.5})$. Furthermore, if we took as robustness criterion the absolute convergence of the algorithm (not

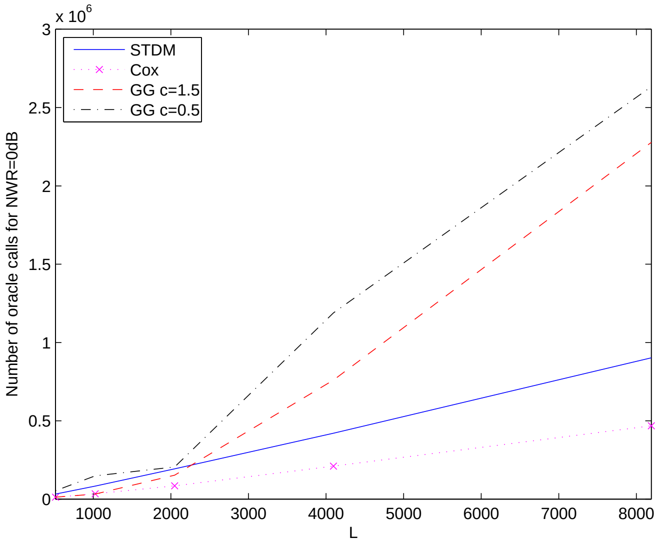


Figure 4.6: Number of calls to the detector for a sensitivity attack (BNSA) for reaching NWR=0 dB as a function of the dimensionality of the watermark for Correlation Detector (Cox), ST-DM and Generalized Gaussian (GG) with $c = 0.5$ and $c = 1.5$ for DWR= 16 dB and $P_f = 10^{-4}$.

only achieving NWR=0 dB), the advantage of the GG detector is even better both in number of iterations and in number of calls to the detector; that is, while for the GG detector convergence is slowly achieved several iterations after reaching NWR= 0 dB, for correlation detectors BNSA achieves both NWR<0 dB and convergence in just one iteration.

4.2.4.5. Zero-Knowledge Watermark Detection for blind watermarking

The formal definition of a zero-knowledge watermark detection scheme concreted for a blind detection mechanism [26, 22] can be stated as follows:

Definition 3 (Zero-Knowledge Watermark Detection) *Given a secure commitment scheme with the operations* Com() *and* Open()*, and a blind watermarking scheme with the operations* Embed() *and* Detect()*, the watermarked host data* $\mathbf{z}$*, and the commitments on the watermark* $C_{\mathbf{w}}$ *and key* C_{K_w} *(for a keyed scheme), with their respective public parameters* $par_{com} = (par_{com}^{\mathbf{w}}, par_{com}^{K_w})$*, a zero-knowledge blind watermark detection protocol for this watermarking scheme is a zero-knowledge proof of knowledge between a Prover* $\mathcal{P}$ *and a Verifier* $\mathcal{V}$ *where on common input* $x := (\mathbf{z}, C_{\mathbf{w}}, C_{K_w}, par_{com})$*,* $\mathcal{P}$ *proves knowledge of a* *tuple* $aux = (\mathbf{w}, K_w, r_{com}^{\mathbf{w}}, r_{com}^{K_w})$ *such that:*

$$\begin{aligned}[(Open(C_{\mathbf{w}}, \mathbf{w}, r_{com}^{\mathbf{w}}, par_{com}^{\mathbf{w}}) = true)\wedge \\ (Open(C_{K_w}, K_w, r_{com}^{K_w}, par_{com}^{K_w}) = true)\wedge \\ (Detect(\mathbf{z}, \mathbf{w}, K_w) = true)]\end{aligned}$$

The zero-knowledge watermark detection protocol developed by Adelsbach and Sadeghi [26] has a blind and a non-blind version. The communication complexity of the non-blind version is studied in [25]; it is much more inefficient than the blind-version due to the higher number of committed operations that must be undertaken. Piva *et al.* ZK detection protocol for ST-DM [186] is based on a blind correlation detector.

4.2.4.6. Zero-Knowledge Subproofs

The proofs that are employed in the previous zero-knowledge detectors and in the presented Generalized Gaussian one are shown in Table 4.2 with their respective communication complexity, which has been calculated when applied to the Damgård-Fujisaki commitment scheme [78] as a function of the security parameters F, B, τ and k, defined in Section 1.1.1.3.

Table 4.2: Zero-knowledge subproofs and their communication complexity.

Proof	Cpx_{cm} (bits)
$PK_{\mathrm{op}}[m, r : C_m = g^m h^r \mod n]$	$3\lvert F\rvert + \lvert\tau\rvert + 2B + 3k + 2$
$PK_{\mathrm{eq}}[m, r_1, r_2 : C_m^{(1)} = g_1^m h_1^{r_1} \mod n \wedge C_m^{(2)} = g_2^m h_2^{r_2} \mod n]$	$4\lvert F\rvert + \lvert\tau\rvert + 2B + 5k + 3$
$PK_{\mathrm{sq}}[m, r_1, r_2 : C_m = g_1^m h_1^{r_1} \mod n \wedge g_2^{m^2} h_2^{r_2} \mod n]$	$4\lvert F\rvert + \lvert\tau\rvert + 3B + 5k + 3$
$PK_{\mathrm{int}}[m, r : C_m = g^m h^r \mod n \wedge m \in [a, b]]$	$25\lvert F\rvert + 5\lvert\tau\rvert + 10B + 27k + 2\lvert n\rvert + 20$
$PK_{\geq 0}[m, r : C_m = g^m h^r \mod n \wedge m \geq 0]$	$11\lvert F\rvert + 4\lvert\tau\rvert + 12B + 14k + 9$
$PK_{\mathrm{sqrt}}[m, r_1, r_2 : C_m = g^m h^{r_1} \mod n \wedge C_{n\sqrt{m}} = g^{n\sqrt{m}} h^{r_2} \mod n]$	$48\lvert F\rvert + 9\lvert\tau\rvert + 18B + 53k + 6\lvert n\rvert + 39$
$PK_{\mathrm{abs}}[m, r_1, r_2 : C_m = g^m h^{r_1} \mod n \wedge C_{\lvert m\rvert} = g^{\lvert m\rvert} h^{r_2} \mod n]$	$19\lvert F\rvert + 6\lvert\tau\rvert + 16B + 24k + 15.$

The first five proofs are already existing zero-knowledge proofs for the opening of a commitment [78] (PK_{op}), the equality of two commitments [47] (PK_{eq}), the square of a commitment [47] (PK_{sq}), a commitment is inside an interval [47] (PK_{int}) and non-negativity of a commitment [146] ($PK_{\geq 0}$).

All these proofs are just simple operations, but the lack of some operations like the computation of the absolute value or the square root, both necessary for the first implementation of the GG ML detector, led us to the development of the last two zero-knowledge proofs; PK_{sqrt} represents a proof that a committed integer is the rounded square root of another committed integer, and it is based on a mapping of quantized square roots into integers. PK_{abs} allows the application of the absolute value operator to a committed number, without disclosing the magnitude nor the sign of that number. Both proofs are described in Appendix 2.D.

4.2.5. Zero-Knowledge GG Watermark Detector

Our zero-knowledge version of the Generalized Gaussian detector conceals the secret pseudorandom signal s_k using the Damgård-Fujisaki scheme [78] C_{s_k}. The supposedly watermarked image Y_k is publicly available, so the perceptual analysis (α_k) and the extraction of the parameters β_k and c_k can be done in the public domain, as well as the estimation of the threshold η for a given point in the ROC. In this first implementation, only shape factors $c = 1$ or $c = 0.5$ are allowed, so the employed c_k will be the nearest to the estimated shape factor. The target is to perform the calculation of the likelihood function

$$D = \sum_k \beta_k^{c_k} \left(|Y_k|^{c_k} - \underbrace{|\overbrace{Y_k - \alpha_k s_k}^{A_k}|^{c_k}}_{B_k} \right),$$

and the comparison with the threshold η, without disclosing s_k.

The protocol executed by Prover and Verifier so as to prove that the given image Y_k is watermarked with the sequence hidden in C_{s_k} is the following:

1. Prover and Verifier calculate the commitment to $A_k = Y_k - \alpha_k s_k$ applying the homomorphic property of the Damgård-Fujisaki scheme:

$$C_{A_k} = \frac{g^{Y_k}}{C_{s_k}^{\alpha_k}}.$$

2. Next, the Prover generates a commitment $C_{|A_k|}$ to the absolute value of A_k, sends it to the Verifier, and proves in zero-knowledge that it hides the absolute value of the commitment C_{A_k}, through the developed proof PK_{abs} (Section 2.D.2).

3. If $c = 1$ (Laplacian features) then the operation $|A_k|^c$ is not needed, so, just for the sake of notation, $C_{B_k} = C_{|A_k|}$.

   If $c = 0.5$, the rounded square root of $|A_k|$ must be calculated by the Prover; then he generates the commitment $C_{B_k} = C_{\sqrt{|A_k|}}$, sends it to the Verifier and proves in zero-knowledge the validity of the square root calculation, through the proof PK_{sqrt} (Section 2.D.1).

4. Both Prover and Verifier can independently calculate the values $\beta_k^{c_k}$ and $|Y_k|^{c_k}$, and complete the committed calculation of the sum $D =$

$\sum_k \beta_k^{c_k} \left(|Y_k|^{c_k} - B_k\right)$, thanks to the homomorphic property of the used commitment scheme:

$$C_D = \prod_k \left(\frac{g^{|Y_k|^{c_k}}}{C_{B_k}} \right)^{\beta_k^{c_k}}.$$

5. Finally, the Prover must demonstrate in zero-knowledge that $D > \eta$, or equivalently, that $D - \eta > 0$, which can be done by running the proof of knowledge by Lipmaa [146] on $C_{th} = C_D g^{-\eta}$.

4.2.5.1. Improved GG Detector with Binary Antipodal Spreading Sequence (GGBA)

When the spreading sequence s_k is a binary antipodal sequence, so it takes only values $\{\pm s\}$, we can apply a trivial transformation to the detection function of the GG detector (Eq. (4.1)):

$$\begin{aligned}
D &= \sum_k \beta^c \left(|Y_k|^c - |Y_k - \alpha_k s_k|^c\right) \\
&= \sum_k \beta^c \big(|Y_k|^c - (|Y_k - \alpha_k s|^c \cdot \mathbf{1}_{\{s\}}(s_k) + |Y_k + \alpha_k s|^c \cdot \mathbf{1}_{\{-s\}}(s_k))\big) \\
&= \sum_k \beta^c \left(|Y_k|^c - \left(|Y_k - \alpha_k s|^c \cdot \frac{1}{2s}(s + s_k) + |Y_k + \alpha_k s|^c \cdot \frac{1}{2s}(s - s_k)\right)\right) \qquad (4.2) \\
&= \underbrace{\sum_k \beta_k^{c_k} \left(|Y_k|^{c_k} - \frac{1}{2}\left(|Y_k - s\alpha_k|^{c_k} + |Y_k + s\alpha_k|^{c_k}\right)\right)}_{G} \\
&\quad - \sum_k \underbrace{\frac{\beta_k^{c_k}}{2s}\left(|Y_k - s\alpha_k|^{c_k} - |Y_k + s\alpha_k|^{c_k}\right)}_{H_k} s_k. \qquad (4.3)
\end{aligned}$$

In (4.2) we use the fact that s_k can only be given a value s or $-s$ in order to substitute the indicator function $\mathbf{1}_{\{s\}}(s_k) = \frac{1}{2s}(s + s_k)$ and $\mathbf{1}_{\{-s\}}(s_k) = \frac{1}{2s}(s - s_k)$.

The factors termed as G and H_k in (4.3) can be computed in the clear-text domain, working with floating-point precision arithmetic, and then have their commitments generated. This implies that all the non-linear operations are transferred to the clear-text domain, greatly reducing the communication overhead, as will be shown in Section 4.2.7; only additions and multiplications must be performed in the encrypted domain, and they can be undertaken through the homomorphic properties of the commitment scheme. This transference also diminishes the computational load, as clear-text operations are much more efficient than modular operations in a large ring.

The zero-knowledge protocol can be reduced to the following two steps:

1. Prover and Verifier homomorphically compute $th = D - \eta$

$$C_{th} = \frac{g^{G-\eta}}{\prod_k C_{s_k}^{H_k}}.$$

2. The Prover demonstrates the presence of the watermark by running the zero-knowledge proof that $D - \eta > 0$.

The number of needed proofs during the protocol is reduced to only one, what propitiates the aforementioned reduction in computation and communication complexity, with the additional advantage that this scheme can be applied to any value of the shape parameter c_k, so it will be preferred to the previous one unless s_k is not binary antipodal.

4.2.6. Security Analysis for the GG Detection Protocols

After presenting the protocols for the zero-knowledge implementation of the Generalized Gaussian ML detector, we can state the following theorem:

Theorem 1 *The developed detection protocols for the Generalized Gaussian detector are computationally sound and statistically zero-knowledge.*

A sketch of the proof for this theorem can be found in Appendix 4.A.

The reformulation of the generalized Gaussian protocol deserves two comments concerning security. The first one involves the non-linear operations that were performed under encryption in Section 4.2.5, which are now transferred to the public clear-text domain. Although this could seem at first sight a knowledge leakage, currently it is not; all those operations can be performed with the same public parameters as in Section 4.2.5 in a feasible time, so the parameters G and H_k that are publicly calculated in this protocol could also be obtained in the previous version, and their disclosure gives no *extra* knowledge.

The second comment deals with the correlation form of the reformulation, and its resilience to blind sensitivity attacks. Even when the operation performed in the encrypted domain is a correlation, the additive term (G) is what preserves the bounded-increment property, by virtue of which component-wise modifications of the input signal only produce bounded increments on the likelihood function

$$-\alpha^c \leq |Y_k|^c - |Y_k - \alpha s_k|^c \leq \alpha^c, \qquad c < 1.$$

The result of the addition is not disclosed during the protocol; thus, the correlation cannot be known even when the term G is public, and both terms cannot be decoupled, so no extra knowledge is learned from G, and the difficulty for finding points in the detection boundary, that is a necessary step for sensitivity attacks, remains, as well as the shape of the detection regions, unaltered.

4.2.7. Efficiency and Practical Implementation

We will measure the efficiency of the developed protocols in terms of their communication complexity, as this parameter is what entails the bottleneck of the system, and it is easily quantifiable given the complexity measures calculated in the previous sections for each of the subprotocols.

Taking into account the plot of the raw protocol (Section 4.2.5), a total of $2L$ commitments (with a length $|n|$) are interchanged, namely the L commitments that correspond to the secret pseudorandom sequence $\mathbf{s}$ and the L commitments to $|A_k|$, while in the GGBA detector (Section 4.2.5.1) only the L commitments to $\mathbf{s}$ are sent; the rest of the commitments are either calculated using homomorphic computation or are already included in the complexity of the subprotocols.

Thus, the total communication complexity for the detector applied to Laplacian distributed features and $c = 0.5$ in the first scheme, as well as the complexity for the improved $GGBA$ detector can be expressed as

$$\begin{aligned}
\mathrm{Cpx}_{cm,ZKWD_{\mathrm{GG}(c\,=\,1)}} &= 2L|n| + L\cdot\left(\mathrm{Cpx}_{cm,PK_{\mathrm{abs}}} + \mathrm{Cpx}_{cm,PK_{\mathrm{op}}}\right) + \mathrm{Cpx}_{cm,PK_{\geq 0}},\\
\mathrm{Cpx}_{cm,ZKWD_{\mathrm{GG}(c\,=\,0.5)}} &= 2L|n| + L\cdot\left(\mathrm{Cpx}_{cm,PK_{\mathrm{abs}}} + \mathrm{Cpx}_{cm,PK_{\mathrm{op}}} + \mathrm{Cpx}_{cm,PK_{\mathrm{sqrt}}}\right)\\
&\quad + \mathrm{Cpx}_{cm,PK_{\geq 0}},\\
\mathrm{Cpx}_{cm,ZKWD_{\mathrm{GGBA}}} &= (L+1)|n| + L\cdot \mathrm{Cpx}_{cm,PK_{\mathrm{op}}} + \mathrm{Cpx}_{cm,PK_{\geq 0}}.
\end{aligned}$$

In every calculation, L proofs of knowledge of the opening of the initial commitments have been added, as even when they are not explicitly mentioned in the sketch of the protocols, they are needed to protect the Verifier.

In order to reduce the total time spent during the interaction, it is possible to convert the whole protocol in a non-interactive one, following the procedure described in [41], keeping the condition that the parameters for the commitment scheme must not be chosen by the Prover, or he would be able to fake all the proofs. In addition to the reduction in interaction time, the use of this technique also overcomes the necessity of a honest Verifier that some subprotocols impose.

The calculated complexity for Piva *et al*'s ST-DM detector and Adelsbach

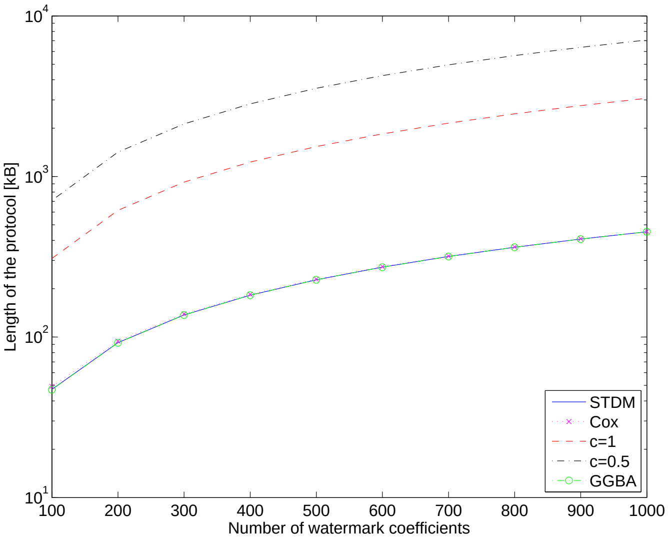


Figure 4.7: Communication complexity in kB for the studied protocols.

and Sadeghi's blind correlation-based detector is the following:

$$\begin{aligned}\mathrm{Cpx}_{cm,ZKWD_{\mathrm{STDM}}} &= (L+1)|n| + L\cdot \mathrm{Cpx}_{cm,PK_{\mathrm{op}}} + \mathrm{Cpx}_{cm,PK_{int}},\\ \mathrm{Cpx}_{cm,ZKWD_{\mathrm{SS}}} &= (L+1)|n| + L\cdot \mathrm{Cpx}_{cm,PK_{\mathrm{op}}} + 2\mathrm{Cpx}_{cm,PK_{\geq 0}} + \mathrm{Cpx}_{cm,PK_{sq}}.\end{aligned}$$

As a numeric example, in Figure 4.7 the evolution of the communication complexity for every protocol is compared using $|F| = 80$, $|n| = 1024$, $B = 1024$, $\tau = 2^{256}$ and $k = 40$, for growing L. All the protocols have complexity $\mathcal{O}(L)$. The two protocols for Generalized Gaussian host features with $c = 1$ and $c = 0.5$ have a higher complexity, due to the operations that cannot be computed by making use of the homomorphic property of the commitment scheme (absolute value and square root). Nevertheless, their complexity is comparable to that of the zero-knowledge non-blind detection protocol developed by Adelsbach *et al.* [25].

On the other hand, the zero-knowledge *GGBA* detector achieves the lowest communication complexity of all the studied protocols, even lower than the previous correlation-based protocols, with the increased protection against blind sensitivity attacks when $c < 1$ is used, being this the first benefit of the reformulated algorithm.

Furthermore, the communication complexity of the protocol is constant if we discard the initial transmission of the commitments for the spreading sequence and their corresponding proofs of opening; once this step is performed, the protocol can be applied to several watermarked works for proving the presence of the same watermark with a (small) constant communication complexity.

Regarding computation complexity, the original detection algorithm (without the addition of the zero-knowledge protocol) for the generalized Gaussian is more expensive than ST-DM or Cox's (normalized) linear correlator, due to its non-linear operations. The use of zero-knowledge produces an increase in computation complexity, as, additionally to the calculation and verification of the proofs, homomorphic computation involves modular products and exponentiations in a large ring, so clear-text operations have almost negligible complexity in comparison with encrypted operations.

The second benefit of the presented GGBA zero-knowledge protocol is that all the non-linear operations are transferred from the encrypted domain (where they must be performed using proofs of knowledge) to the clear-text public domain; thus, all the operations that made the symmetric protocol more expensive than the correlation-based detectors can be neglected in comparison with the encrypted operations, so the computation complexity of the zero-knowledge GGBA protocol will be roughly the same as the one for the correlation-based zero-knowledge detectors.

4.3. Secure Cloud Computing with application to Medical Clouds

The second application shown in this chapter consists in secure cloud computing and its particularization to medical clouds, in which very sensitive information is managed.

In recent years, the paradigm of Cloud computing has gained an increasing interest from the academic community as well as from the commercial point of view. Cloud is a very appealing concept both for the providers–that can benefit from hiring out their extra computation and storage resources–and for the users–that can avoid the initial investment on resources by outsourcing their processes and data to a Cloud–.

From a technological point of view, there are currently some challenges that Cloud still needs to tackle in order to be fully operational; they are mainly related to scalability, manageability, interoperability and multi-tenancy. But the most important issues that can hold back the widespread adoption of Cloud are security and privacy. Both concepts are very close to each other in Cloud, as there can be no privacy without security. Nevertheless, privacy is a more specific requirement, and it is related only to sensitive data and/or processes. In this section, we focus on privacy for signal processing.

While many research efforts are devoted nowadays to guaranteeing security [133] in Clouds, dealing with aspects such as authentication through federated identities or basic encryption of the managed data [195, 99], the issue of preserving data privacy and addressing the different data protection legislations remains open. The privacy problem in Cloud is a very severe concern [128, 91, 100, 189, 210], mainly because data can be distributed among different servers and even different countries with their own data protection legislation. Furthermore, the fuzzy nature of data processing and location in Clouds can negatively affect the trust that users put on these systems, as they face the risk of losing control over their data and processes when they are outsourced to a Cloud; this fact can constitute a severe barrier for Cloud adoption [20].

Cloud Privacy in general is a very broad subject that would be out of the scope of this chapter; hence, we narrow the problem and devote our efforts to the framework of Cloud Privacy for Signal Processing and we firstly define, as a very coarse classification, two main kinds of low-level privacy that can be required in a signal processing Cloud application, namely signals privacy and processes privacy, exemplified as follows:

- On the one hand, users might want to outsource the storage or processing of some sensitive signals to a Cloud, or input these signals to some service provided in a Cloud; this is the case when only one level of privacy protection (*signals privacy*) is needed.
- On the other hand, an enterprise might have developed a private algorithm, and they want to act as Cloud vendors, offering its functionality through a Cloud; the sensitiveness of the algorithm itself stems from its commercial value; thus, the process must remain concealed; this case implies a second level of privacy protection (*processes privacy*).

Of course, both kinds of privacy can be required at the same time for a given application. If we add the Cloud paradigm to the most relevant use-cases in SPED, we are left with a significant number of use-cases that would greatly benefit from a privacy-preserving Cloud solution. We have chosen two of these cases for this section, that are representative enough to show the potential of such a solution: secure biometric recognition, and secure medical analysis/diagnosis.

4.3.1. Secure Biometric Recognition

Figure 4.8 shows this scenario, where a face, iris, fingerprint or other biometric information is contrasted against the templates stored in a biometric database located at a server (or distributed among several collaborative servers) in order to determine whether the individual whose biometric is presented is recognized by the system (see also Chapter 6 for a specific privacy-preserving solution in this scenario). We can exemplify it with a CCTV system where the faces of the recorded citizens are matched against a database of potential criminals.

The sensitive information in this scenario comes from two sources: on the one hand, the biometric signals that are presented to the system for recognition, and on the other hand, the templates stored at the database. In this case, the server that holds the database and the server that processes the biometrics and checks for a match are the untrusted agents from which the privacy of the signals must be protected.

The two levels of protection are also present in this scenario, not only the *signals privacy*, but also the *processes privacy*, related to the recognition algorithm used for finding a match. That is, the presented biometric sample must be kept encrypted while it is processed in the server, and the database templates may be also kept encrypted within the servers. Examples of secure face recognition systems with encrypted samples can be found in [89, 196] and in Chapter 6.

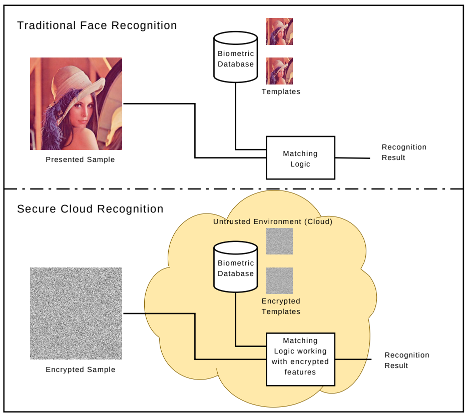


Figure 4.8: Secure Biometric Recognition scenario.

This scenario is very amenable to a Cloud implementation, due to the presence of a large database of biometric samples, that can be distributed among several servers, and due to the fact that the matching against these servers can be easily parallelized. In this sense, the Cloud would manage the storage of the database, and also the execution of the recognition algorithm when a biometric sample is presented. The cloud would then represent the untrusted environment in which privacy of the signals and algorithms must be preserved.

4.3.2. Secure medical analysis/diagnosis

In this scenario some biomedical signals (DNA, ElectroCardioGrams - ECGs, Magnetic Resonance Images - MRIs,...) from a patient or group of patients are presented to an expert system that must complete some given analysis and/or report a diagnostic from these signals. An example of a secure DNA diagnostic

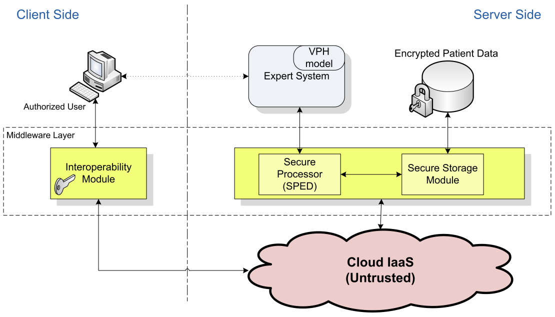


Figure 4.9: Proposed architecture for Secure Medical Cloud using VPH models: the storage and processing are performed in the Cloud, but patient data are protected, as they are always encrypted. The Expert System works with encrypted data through the use of the secure processor.

system can be found in [221] (cf. Section 4.4), and an example of a secure ECG classification system, in [39].

The sensitive information comprises the biomedical signals, while the analysis system represents the untrusted party, from which the client/patient may want to protect her information. The performed analysis or the diagnosis algorithm can also be subject of protection, at the level of processes privacy.

The medical database that holds patients' signals and records can be stored in a Cloud, provided that the access to these records is adequately controlled; on the other hand, the processing of medical signals has already been shown to benefit from the use of Grids (HealthGrids [4]). Again, the Cloud/Grid represents the untrusted environment that must implement some mechanism for preserving the privacy of the signals and the analysis algorithms. Following this direction, we have also proposed a secure system for the processing of medical data through VPH (*Virtual Physiological Human*) models executed in a Cloud environment using SPED primitives [226]; the proposed architecture is shown in Figure 4.9.

4.3.3. Related work

The idea of using homomorphic encryption in cloud environments is quite novel. There is some work on this topic, but a complete solution for the Cloud is still missing. In [62] some common cloud security problems are characterized, and the proposed solutions include empowering data with intelligence to protect itself using trusted computing or privacy-enhanced business intelligence based on homomorphic encryption. In [179] a privacy manager for cloud computing, an appliance for secure cloud access, was presented. To achieve this goal an obfuscation mechanism based on homomorphic encryption is used. The paper describes some application scenarios like SQL queries or photo tagging. The solution only encrypts part of the information that is sent to the Cloud in order to keep efficiency at reasonable levels. As a conclusion, previous related work focuses on particular applications or assumes that the service provider is honest, tackling a different privacy problem as the one that concerns us in this chapter, namely the processing of signals in the Cloud as an untrusted environments.

4.3.4. CryptoDSPs for Privacy in Cloudified Signal Processing

In order to deal with privacy issues in Signal Processing performed in a Cloud scenario, we propose that Cloud services and infrastructures dealing with these applications adopt some of the efficient secure processing techniques from Signal Processing in the Encrypted Domain and Secure Function Evaluation. We present the architecture of a privacy-preserving Cloud computing system for the outsourcing of Signal Processing by using SPED techniques and materializing them as a virtual DSP (Digital Signal Processor) that performs the needed operations in the encrypted domain; we denote this processor *Virtual CryptoDSP*.

Figure 4.10 presents the proposed conceptual architecture, which is transparent to the final user, and adds three main blocks to the classical Cloud architecture, namely the *Virtualized Coded Storage*, the virtualized *CryptoDSP core*, and the *Client Plug-In*. In order to provide the greatest versatility, the new elements are implemented as middleware on top of a Cloud infrastructure (IaaS). A specifically developed API, presented at the PaaS level and comprising Secured Signal Processing operations that can be interpreted by the Virtual CryptoDSP, guarantees that any privacy-aware signal processing application can be built up on top of this secure middleware in order to be endowed with the required level of privacy. We describe the functionality and the elements that compose each of these blocks:

Virtualized CryptoDSP core

This element holds the server-side implementation (parallelized, to take advantage of the Cloud infrastructure) of the secure signal processing primitives, using SPED technologies, and implementing also a communication module for interacting with the client-side plug-in during the execution of the corresponding interactive protocols. The logic implemented by the CryptoDSP core includes only the on-line computation. The off-line computation is assumed by the Virtualized Coded Storage Module.

The implemented primitives should be designed to work with encrypted signals, but it is also possible, and desirable, that, through the corresponding API, the users can also provide a signal processing circuit that gets compiled to primitives to be interpreted by the CryptoDSP.

Client Plug-In

The client plug-in is the client-side module that must present a transparent interface from the client view-point. It comprises a cryptographic module for data encryption, key generation and management, and a communication module for the on-line interaction with the cryptoDSP at the server-side.

Virtualized Coded Storage Module

This module performs three complementary functions: 1) server-side data encryption, 2) data pre-processing and off-line processing for the secure protocols, and 3) Management of the (possibly distributed) storage of the encrypted and preprocessed data.

The main target of this module is the optimization of the computational load of the secure protocols for reducing their on-line time and/or the needed communication bandwidth.

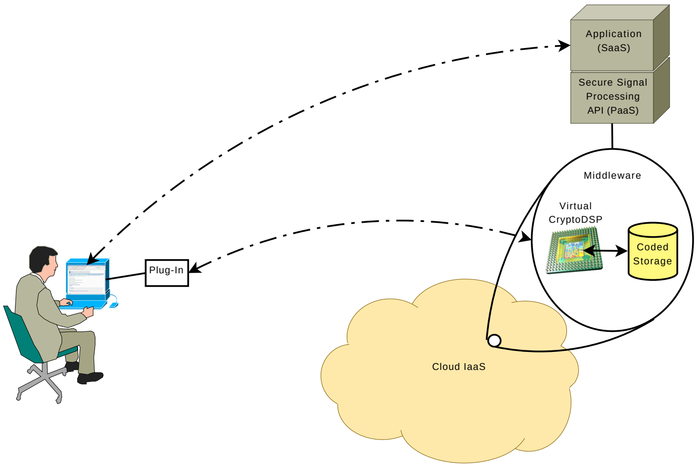


Figure 4.10: Architecture of a Cloud Computing system supporting Private Signal Processing outsourcing through a Virtual CryptoDSP.

4.3.5. Practical Considerations

It must be taken into account that the proposed one is a conceptual architecture, and most of the current SPED primitives are still too restrictive to produce a practical solution for the CryptoDSPs paradigm, taking into account their two most important limiting factors:

- *Computational load*: The implemented primitives have to be adapted to the Cloud infrastructure, and present a highly asymmetric load balance: the client must be as lightweight as possible, while the server, in the Cloud, can handle a much heavier processing load.
- *Bandwidth*: It is a very limiting factor, and it can become the bottleneck of the system for SPED primitives. The Cloud architecture allows to deal with the bandwidth problem distributing it among the nodes of the Cloud, instead of focusing it at the link between the client plug-in and the CryptoDSP.

For these shortcomings to be overcome, it is necessary to develop noninteractive efficient solutions; this is one of the current hot topics in SPED, materialized in the research in practical fully homomorphic cryptosystems and noninteractive privacy-preserving protocols (cf. Chapter 6). Nevertheless, there are some preliminary proposals of simple proof-of-concept privacy-preserving systems executed in Cloud environments [192], still limited to linear operations and with an important bandwidth bottleneck.

4.4. Privacy and DNA Sequences

As the last application presented in this chapter, we have chosen one dealing with the most sensitive signal a privacy-aware system can handle: DNA sequences.

The Human Genome Project [5] took nearly 13 years and required more than US-$3 billion to sequence a ‘prototypical’ human genome. Nonetheless, biomedical technology is advancing at a rapid pace and the costs for sequencing an individual’s genome are dropping. The goal set by the U.S. National Institute of Health is to reduce sequencing costs for a human genome to a hundred thousand dollars in 2009 and to less than a thousand dollars by 2014 [117]. This target is also known as the $1000 genome.[1] At that cost, it is anticipated that by 2015 genomic information will be ubiquitously used by healthcare providers and that patients will be able to acquire a digital record of their genome.

The human genome contains a wealth of information about a person’s body; broad access to the genome is likely to revolutionize medical diagnosis and treatment. Doctors can, for example, use genomic information to test whether a person has a pre-disposition towards developing a specific disease, even years before the first symptoms appear. In treatment, genomic data may be used to predict

[1] Earlier in 2012 this target was fulfilled by a commercial system by Life Technologies [21], that can provide a solution to sequence the full DNA in less than one day and with a cost of US$1,000.

whether a patient will react positively against a specific therapy or whether the treatment will likely fail, thereby reducing the overall costs and increasing the effectiveness of the therapy. Finally, it may be possible to create an individualized drug therapy for each patient by analyzing his genetic profile and predicting his response to different medications.

Broad access to and storage of personal Desoxyribo-Nucleic Acid (DNA) sequences involves significant risks to personal privacy and may open the door for discrimination based on genomics. For instance, a person carrying a gene known to increase the likelihood of a particular cancer may be denied coverage by the health insurance company; an employee may be rejected for a permanent work contract due to his pre-disposition towards a disabilitating disease; or the discovery of parental relationships via DNA profiling may have undesirable consequences for the person's private life. These are only some of the risks we can foresee at this time; once the functionality of the human genome is fully uncovered there may be even more significant risks to privacy. As we move forward, there is a clear and emerging need for privacy-preserving mechanisms for the protection of genomic data.

Privacy concerns about DNA information have traditionally been addressed through laws and procedures: Healthcare professionals are required to keep sensitive data confidential and make it available only with explicit consent of the patient. So far this traditional approach has worked reasonably well, mostly due to the limited availability and use of genomic profiles in established medical centers. Nonetheless, as genomic profiles become ubiquitous, this traditional form of protection may be insufficient to prevent sensitive information leakage. We believe that cryptographic privacy-preserving protocols will become invaluable components that complement the procedural approach.

The problem setup considered here is as follows: A patient has a digital record of her DNA sequence and wants to give another party (such as her healthcare provider) selective access to run a query on this record, for instance, to find out whether she has a pre-disposition to a particular disease. As she is concerned about her privacy, she does not want to disclose her DNA profile to the health-care provider in the clear. On the other hand, her health-care provider may like to keep the details of the query confidential as it is commercially valuable. In the next subsection, we identify the differentiating properties of queries run on DNA data and their implications on the design of privacy-preserving protocols.

4.4.1. Queries on DNA data

We denote a DNA sequence as a finite string over the alphabet $\Sigma = \{A, C, T, G\}$, representing the four different nucleotides Adenine, Cytosine, Thymine and Guanine (also known as bases). How this sequence regulates human physiology is under investigation. However, one of the main regulation mechanisms is through encoding of proteins. Triplets of nucleotides in particular sections of a DNA sequence, known as coding regions, encode different amino-acids. In turn, a sequence of amino-acids forms a protein, which regulates various functions in the body.

In the following, we discuss some properties of typical queries to DNA data that need to be considered when designing privacy-preserving protocols.

- **Mutations:** A mutation is a deviation on the DNA sequence that may affect one single nucleotide or a sequence of subsequent nucleotides. It may involve substitutions (one nucleotide is converted into another), deletions and insertions (missing or extra nucleotides due to imperfections in the replication process). Specific mutations in the coding regions are known to be indicative of some diseases. The location of these mutations can be fixed and known in advance; alternatively, mutations can occur at a relative distance from a fixed marker. In order to query for the presence of a specific mutation, one usually checks whether a certain string $\boldsymbol{x} \in \Sigma^*$ appears in the DNA sequence.

  A mutation may also appear in a non-coding region of a DNA sequence, where it is clinically irrelevant. In that case the mutation becomes an error, which the query mechanism should handle gracefully.

- **Sequencing Errors:** Today, even the best DNA sequencing methods cannot guarantee 100% accuracy. Due to the imperfections of the chemical sequencing process, three different types of errors occur: symbol substitutions (an incorrect base is recorded), insertions (a base that is not present in the genome is reported in the digital record) and deletions (the sequencing process fails to report a base, even though it is present in the analyzed genome). Queries on DNA sequences should thus be able to cope with infrequent errors of these types. In the literature, these errors are usually called *Edit* errors, since they frequently occur when transcribing a text or when using an Optical Character Recognition (OCR) tool. There is a known distance measure, called Levenshtein or Edit distance [144], allowing to quantify the number of substitutions, insertions and deletions that a sequence has suffered with respect to a reference.

- **Many-to-one Mappings:** While each triplet of DNA bases encodes one amino-acid, this encoding is not unique: there exist different base triplets that encode the same amino-acid. As only the latter is relevant in diagnosis, DNA queries should be able to handle this ambiguity.

- **Incomplete Specifications:** In the existing medical genomic databases, there are many DNA sequences that are not completely specified, as the

> exact effect of punctual mutations has not yet been completely determined, even though there is evidence of the relationship between these mutations and a known disease. In this case, the possible queries that can be applied in order to detect those mutations must offer a flexibility to handle incomplete specifications, apart from the already mentioned error resilience.

A natural representation for a query that is able to cope with the aforementioned properties are regular expressions, implemented as finite automata. Note that regular expressions not only allow to handle incomplete specifications and ambiguity, they can also be used to cope with Edit errors due to sequencing problems or clinically irrelevant mutations (cf. Section 4.4.2.1).

4.4.2. Secure Approximate Searching and Matching

The problem of *approximate string matching* (briefly presented in Section 2.2.1) can be related to several error metrics, but it is commonly associated with the Edit or Levenshtein distance [144], that is the same metric that can account for the same types of errors to which DNA is subjected, namely symbol substitutions, deletions and insertions. Given two strings $\boldsymbol{x}$ and $\boldsymbol{y}$, the Edit distance is defined as the minimum number of Edit errors that $\boldsymbol{x}$ must undergo in order to be transformed into $\boldsymbol{y}$. If this number is below a given threshold, both sequences are said to approximately match; in case of a match, a sequence alignment can be computed, which associates the symbols of $\boldsymbol{x}$ and $\boldsymbol{y}$, up to insertions and deletions.

The commonly used algorithm to compute sequence alignments is a dynamic programming algorithm developed by Needleman and Wunsch [170], even though similar algorithms are also used for speech recognition [235] and spell checking. Besides computing an alignment, the algorithm also determines the Edit distance between two sequences. However, in many applications it is not necessary to obtain an alignment; it suffices to know whether the Edit distance is below a given threshold. This decision problem is a special case of the *approximate string searching* problem, in which a pattern string $\boldsymbol{x}$ is searched in a longer sequence $\boldsymbol{y}$, tolerating Edit errors; both problems can be solved efficiently by running a finite automaton, as we show in the following subsection; hence, the privacy-preserving approximate DNA searching and matching can be solved using the protocol for oblivious automata execution of Section 2.5.

4.4.2.1. Searching and Matching by FSMs

Given a string $\boldsymbol{x}_A$, we use the method in [206] for computing a finite automaton $\mathcal{LEV}_d(\boldsymbol{x}_A)$ that accepts all strings that have at most Levenshtein distance

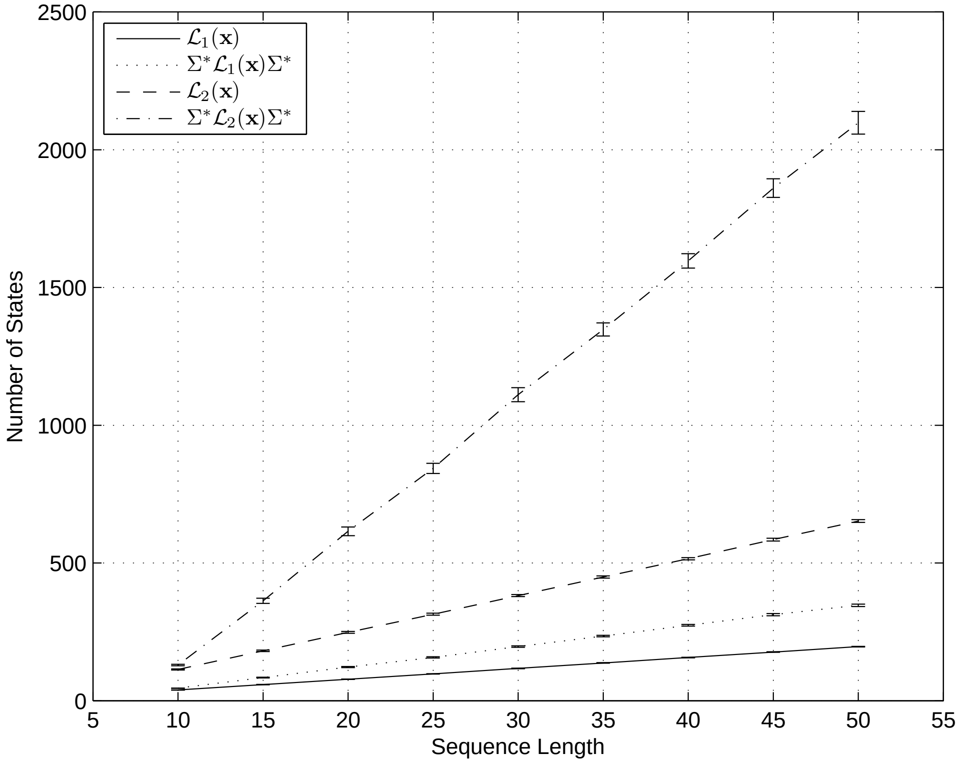


Figure 4.11: Number of states of the Levenshtein Automaton and its extension, as a function of the sequence length ($|\boldsymbol{x}_A|$).

d from $\boldsymbol{x}_A$. The resulting minimal automaton is denoted *degree d Levenshtein automaton*. By construction, $\mathcal{LEV}_d(\boldsymbol{x}_A)$ is always *acyclic*. We will denote the language accepted by this automaton as $\mathcal{L}_d(\boldsymbol{x}_A)$. For a fixed d, the algorithm for generating $\mathcal{LEV}_d(\boldsymbol{x}_A)$ given $\boldsymbol{x}_A$ is linear in time and space in the length of the string $\boldsymbol{x}_A$. The dependency on d can be at worst exponential; however, d is usually a small parameter compared to the length of $\boldsymbol{x}_A$ for practical applications (like DNA searching). In this way, the problem of calculating the Levenshtein distance between two sequences $\boldsymbol{x}_A$ and $\boldsymbol{x}_B$ and comparing it to a given threshold d gets reduced to the execution of the computed automaton $\mathcal{LEV}_d(\boldsymbol{x}_A)$ on input $\boldsymbol{x}_B$. This gives a solution to the approximate matching problem.

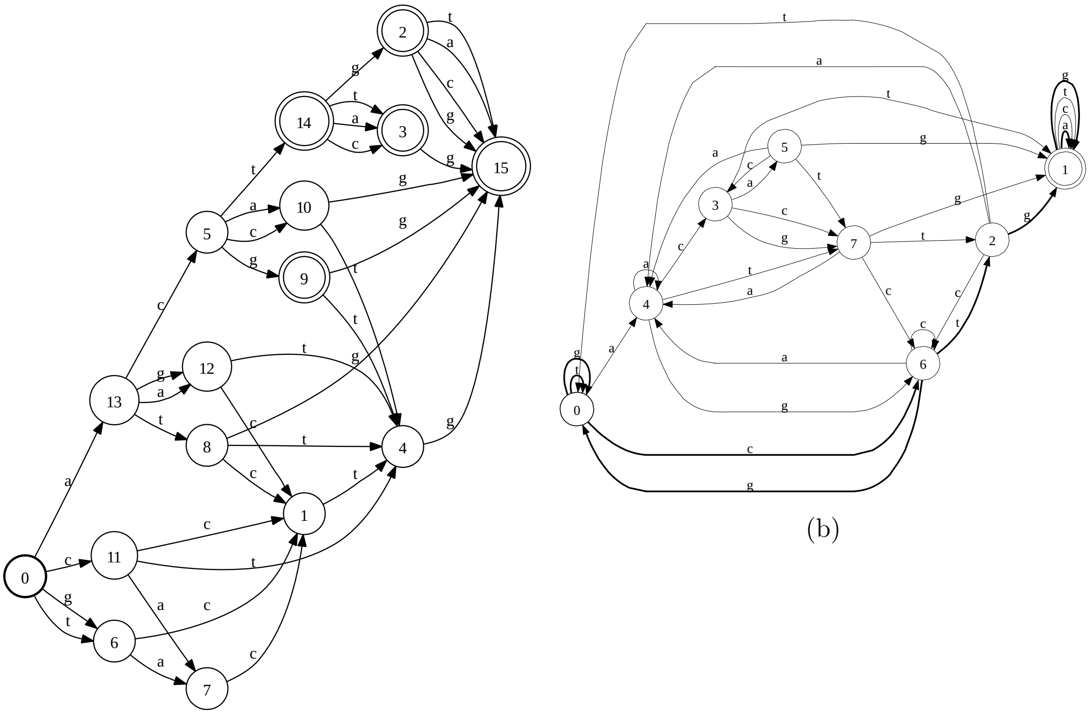
Once the Levenshtein automaton for a given sequence is generated, we extend it to accept the language $\Sigma^*\mathcal{L}_d(\boldsymbol{x}_A)\Sigma^*$. Thus, the resulting automaton accepts any string that contains as substring any of the sequences accepted by the Levenshtein automaton, thus solving the problem of approximate string searching, when the automaton is run on $\boldsymbol{x}_B$.

The advantage of using an automaton instead of a dynamic programming algorithm resides in the fact that an automaton has predefined transitions, and it does not need any comparisons while traversing the input sequence. Comparisons are one of the most expensive operations under encryption (as they reduce to instances of the Millionaire's problem). By using a finite automaton, all the comparisons can be avoided, because they are all *hard-wired* in the automaton itself. Furthermore, using an automaton allows the implementation of any matching problem represented in the form of a regular expression, endowing our privacy-preserving solution with a strong generality.

Even though the construction of the Levenshtein automaton assures that the number of states of $\mathcal{LEV}_d(\boldsymbol{x}_A)$ is linear in the length of the sequence $\boldsymbol{x}_A$, computing the extended automaton $\Sigma^*\mathcal{L}_d(\boldsymbol{x}_A)\Sigma^*$ will increase its number of states. Extending the Levenshtein automaton comprises two concatenations with Σ^*, the right one being trivial, as it only involves adding self-loops in all the final states. This right concatenation cannot increase the number of states of the automaton. Let us suppose that the Levenshtein automaton has n states, t of them being acceptance states; by construction, the automaton is acyclic. Applying the right concatenation, all of the t acceptance states collapse to only one *sink* acceptance state, and the rest of the states remain unaltered. Thus, the resulting automaton after the right concatenation has $n - t + 1$ states, one of them being the unique sink acceptance state, and the only cycles that the automaton has are the self loops in this state.

Applying a known bound on the state complexity of the concatenation of regular languages [246], the left concatenation could increase the number of states of the automaton by at most 2^{n-t}. Nevertheless, this bound is a worst case bound. We have found experimentally that the number of states usually grows linearly even after performing the left concatenation, resulting the number of states of the extended Levenshtein automaton being linear in the length of the input sequence $\boldsymbol{x}_A$. As an example, Figure 4.11 shows the evolution of the number of states of the Levenshtein automaton $\mathcal{LEV}_d(\boldsymbol{x}_A)$ and its extension to the language $\Sigma^*\mathcal{L}_d(\boldsymbol{x}_A)\Sigma^*$, as a function of the length of the sequence $\boldsymbol{x}_A$, for threshold Levenshtein distances of 1 and 2 errors. The plot was obtained using 100 random DNA sequences $\boldsymbol{x}$ for each length and averaging the number of states of the obtained automata; it also shows the 95% confidence intervals. From this figure, it is clear that the state complexity usually is linear in the length of the input sequence.

As a toy example, for the sequence $\boldsymbol{x}_A = [actg]$, the Levenshtein automaton for distance $d = 1$ is shown in Figure 4.12a, while the extension to cope with arbitrary length sequences is shown in Figure 4.12b. It is clear from this example that the extension to arbitrary length sequences does not necessarily imply an increase in the number of states of the automaton; in this case, it even supposes a reduction, due to the short length of the used pattern. Figure 4.12b also shows thicker arcs for the transitions that the automaton makes when inputting the sequence [*ttcggcg**ctg**ga*], where the pattern is present with one deletion, resulting in acceptance.



(a)

(b)

Figure 4.12: State diagram for the Levenshtein automaton accepting all the sequences at distance ≤ 1 of $[actg]$ (a) and its extension to arbitrary length sequences (b).

4.4.3. Secure Approximate Searching and Matching for DNA Sequences

Let us recall the scenario of DNA searching: Two parties $\mathcal{A}$ and $\mathcal{B}$ want to check if the DNA pattern $\boldsymbol{x}_A$ (owned by $\mathcal{A}$) is approximately present in $\mathcal{B}$'s DNA sequence $\boldsymbol{x}_B$, where $|\boldsymbol{x}_B| \gg |\boldsymbol{x}_A|$. Approximate presence means that the Edit distance between $\boldsymbol{x}_A$ and some substring of $\boldsymbol{x}_B$ is less than a given threshold d. The case of matching is similar, except that $|\boldsymbol{x}_B| \approx |\boldsymbol{x}_A|$.

To perform either matching or searching in a privacy-preserving manner, both parties execute the following steps:

1. $\mathcal{A}$ builds the Levenshtein automaton $\mathcal{LEV}_d(\boldsymbol{x}_A)$ corresponding to his sequence $\boldsymbol{x}_A$, given a maximum allowable distance d, following the procedure in [206], and minimizes it. If needed, the number of states can be partially concealed by adding a random number of dummy states.
2. In the case of a search, $\mathcal{A}$ extends the Levenshtein automaton by concatenating the Kleene closure of the alphabet Σ^* at the left and at the right

(see Section 4.4.2.1). The resulting automaton is then minimized. Again, a random number of dummy states can be added in order to partially conceal the number of states of the minimal automaton.

3. Both parties run the protocol presented in Section 2.5.1 with $\mathcal{A}$'s automaton and $\mathcal{B}$'s sequence $\boldsymbol{x}_B$ as inputs, in order to get a binary answer to the approximate matching or searching problem.

Regarding the complexity of the resulting protocol, we can combine the results obtained in Sections 2.5.2 and 4.4.2.1. By virtue of the latter, the extended Levenshtein automaton usually has a state complexity linear $\mathcal{O}(n)$ in the length n of the sequence $\boldsymbol{x}_A$; the former shows that the private evaluation of an automaton with $|Q|$ states and an input alphabet Σ on an input sequence of length N, has a communication complexity of $\mathcal{O}\left(N \cdot (|Q| + |\Sigma|)\right)$. Finally, the application of the developed protocol for the approximate search of a sequence of length n in another sequence of length N incurs in a communication complexity of $\mathcal{O}\left(N \cdot n\right)$. Concerning computational complexity, taking into account that the automaton transformation can be precomputed, and applying the same reasoning as for communication overhead, the total amortized computational complexity for the owner of the *query* n-length sequence $\boldsymbol{x}_A$ is $\mathcal{O}(N \cdot n)$, and for the owner of the *long* N-length sequence $\boldsymbol{x}_B$, it is $\mathcal{O}(N)$. This means that for the party that makes the query, the privacy-preserving protocol has a computational complexity in the same order as the one of the non-privacy preserving protocol, while the complexity for the other party is linear in her sequence's length, and does not depend on the length of the query string.

We can also make one final remark about round complexity. In Section 2.5.2 we have stated that the round complexity of the privacy-preserving protocol for the automaton evaluation is linear in the length of the input sequence ($\boldsymbol{x}_B$). In this particular case, it is known that $\mathcal{A}'s$ Levenshtein automaton will accept only sequences of length smaller than $|\boldsymbol{x}_A| + d$. Thus, if round complexity is a concern and the value $|\boldsymbol{x}_A| + d$ does not have to be kept secret, we can partition the input sequence $\boldsymbol{x}_B$ into several consecutive blocks with an overlap of $|\boldsymbol{x}_A| + d - 1$ symbols, and run in parallel one instance of the oblivious automaton protocol per block. Then, a logical OR can be straightforwardly applied to the obtained (concealed) outputs. Taking the maximal number of blocks, the number of rounds of the resulting protocol does not depend on the length of the input sequence; as a counterpart, the overlaps produce an increase in communication complexity, which is quadratic in the number of states of the automaton. Between the two extreme cases, a tradeoff can be found, with a sublinear round complexity in the length of the input sequence $\boldsymbol{x}_B$ and a subquadratic communication complexity in the number of states of the automaton.

4.5. Conclusions and Further Work

This chapter has presented several application scenarios for privacy-preserving solutions based on SPED primitives, and described in detail specific approaches for three cases, namely zero-knowledge watermark detection, private cloudified signal processing and private queries on DNA sequences.

The presented zero-knowledge watermark detection protocol based on Generalized Gaussian ML detector solves the problem of private detection with a symmetric key watermark scheme, while outperforming the previous correlation-based zero-knowledge detectors implemented to date in terms of robustness against blind sensitivity attacks, and improving on the ROC of the correlation-based spread-spectrum detector with a performance that is near that of ST-DM.

We have also provided a conceptual high-level architecture for implementing SPED primitives in Cloud environments, with application to medical Clouds; most SPED primitives are not yet mature to be implemented with such architecture, but further research may open the door to fully non-interactive and efficient encrypted domain processing in a Cloud environment.

As for the privacy-preserving DNA searching solution based on the oblivious automata protocol of Section 2.5, it constitutes the first efficient privacy-preserving solution for error-resilient DNA searching. Furthermore, due to the versatility of finite state machines, the presented protocol can also be used for privately solving any problem that involves matching a string against a regular expression, such as searching a DNA database with incomplete definitions, oblivious spam checkers and virus analyzers. This work on privacy-preserving DNA queries opened a research line that has been followed by numerous subsequent works, like [134], [136], [45], [103], [119], or [36].

4.A. Sketch of the proof for Theorem 1

Proof *Completeness:* Let us assume that both parties behave according to the protocol. The values C_{A_k} calculated by the correct Prover and the correct Verifier coincide. For correctly produced $C_{|A_k|}$, the completeness of the absolute value subproof guarantees the acceptance of the Verifier; equally, the completeness of the rounded square root subproof guarantees the acceptance for a correctly calculated C_{B_k}. Next, the values of C_D computed by both parties coincide, and, finally, due to the completeness of the non-negativity proof, the Verifier will accept the whole proof in case the signal $\{Y_k\}$ is inside the detection region. For the case of a binary antipodal spreading sequence (Section 4.2.5.1), if the values G, H_k and C_{th} are correctly calculated, the completeness of the non-negativity proof guarantees the acceptance when $\{Y_k\}$ is inside the detection region. This concludes the completeness proof.

Soundness: The binding property of the commitments assures that the Prover will not be able to open the commitments that he calculates (C_{A_k}, $C_{|A_k|}$, C_{B_k}, C_D, C_{th}) to wrong values. Furthermore, the statistical soundness of the used subproofs (absolute value, rounded square root and non-negativity) guarantees that an incorrect input in any of them will only succeed with negligible probability. This fact, together with the homomorphic properties of the commitments, that makes impossible for the Prover to fake the arithmetic operations performed in parallel by the Verifier, propitiates that the probability that a signal $\{Y_k^*\}$ that is not inside the detection region succeeds the proof be negligible.

Zero-Knowledge: We can construct a simulator S^{V^*} such that the real interactions have a probability distribution indistinguishable from that of the outputs of the simulator. The statistical zero-knowledge property of the absolute value, rounded square root and non-negativity subproofs guarantee the existence of simulators for their outputs; thus, S^{V^*} can generate C_{A_k}, C_D and C_{th} as in a real execution of the protocol, thanks to the homomorphic properties of the commitment scheme. On the other hand, it must generate $C_{|A_k|}$ and C_{B_k} as commitments to random numbers; the statistical hiding property of the commitments guarantees that the distribution of these random commitments be indistinguishable from the true commitments. Furthermore, these generated values will not affect the indistinguishability of the simulators for the subproofs, as these simulators do not need knowledge of the committed values in order to succeed. Thus, the output of S^{V^*} is indistinguishable from true interactions of an accepting protocol, and the whole protocol is statistically zero-knowledge.

∎

Chapter 5

Other approaches: Videosurveillance and Multimedia Privacy

Privacy and security have always been key concerns for individuals. They have also been closely related concepts: in order to increase their perception of security, people sacrifice a part of their privacy by accepting to be surveilled by others. The tradeoff between both is usually reasonable and commonly accepted; however, the case of videosurveillance systems has been particularly controversial since their inception, as their benefits are not perceived to compensate for the privacy loss in many cases. The situation has become even worse during the last years with the massive deployment of these systems, which often do not provide satisfactory guarantees for the citizens. This chapter proposes a DRM-based framework for videosurveillance to achieve a better balance between both concepts: it protects privacy of the surveilled individuals, whilst giving support to efficient automated surveillance. This chapter takes advantage of the delicate tradeoff between privacy and security in these videosurveillance scenarios to provide a different solution to those shown in the previous chapters, based on a peculiar use of DRM tools instead of encrypted processing and SPED primitives.

The work shown in this chapter has been partially presented at ACM DRM 2009 [219].

5.1. Introduction

Since the beginning of time, any species that makes its way through evolution must have developed mechanisms to ensure its own security and protection. Human kind is no exception to this rule, and thus, security has always been a concern to our species. Our characteristic advantage for achieving this goal resides in technology. When it reached the required maturity level, videosurveillance was a natural step in this direction. The problem comes when security collides with privacy. Probably the best definition of privacy so far was given by Westin in 1970 [239]: *"Privacy is the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others."* Individuals accept to give away a part of their privacy in exchange for greater security, but up to what extent? In terms of privacy, the situation has become worse in the last years due to the rise of IP videosurveillance.

Figure 5.1 shows the main components of a modern IP videosurveillance network: basically it is composed of a number of IP cameras which may be in very disparate locations but all of them are connected to a control center via an IP network (either local or through the Internet). In the control center, the incoming streams are managed by one or more processing servers, and they can be stored in hard disks for *a posteriori* access. Typically, there is one or more human operators that supervise in a video console the recordings in search for incidents and other relevant events. Moreover, modern videosurveillance systems are beginning to feature functionalities for automated image analysis which make easier the task of the human operators whilst increasing security. However, this increased efficiency in detecting events and collecting information is raising even more privacy concerns. This motivates the need for a joint framework addressing the tradeoff between privacy and security. In this chapter, such a framework is proposed, and it is proven that, with slight modifications, the paradigm of DRM can yield a solution that covers both aspects of videosurveillance, namely privacy rights of surveilled people and automation targeted towards security: the two sides of the mirror.

The rest of the chapter is organized as follows. Section 5.2 introduces the general framework of privacy in videosurveillance from the point of view of the current European legislation. Section 5.3 reviews the past works on privacy management and protection, paying special attention to the videosurveillance scenario, and also states the current situation of automated videosurveillance. Section 5.4 presents our DRM-based proposal, and Section 5.5 offers a high level perspective on its implementation. The application to a real scenario is illustrated with a use case, described in Section 5.6. Finally, some concluding remarks are given in Section 5.7.

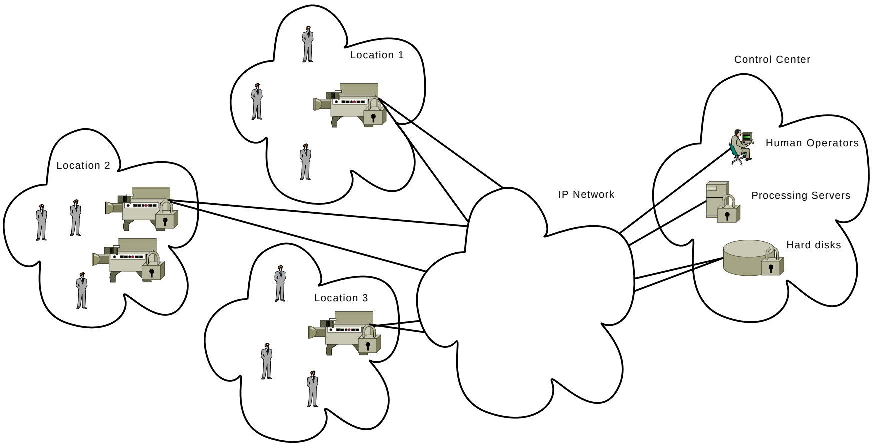


Figure 5.1: Simplified architecture of an IP videosurveillance network

5.2. The legal framework of videosurveillance

In the last years, the exposition of people to videosurveillance systems has increased considerably. Although these systems are usually perceived as a good means to improve individuals security, as they help to prevent, investigate, detect and prosecute criminal offences, an increasing concern exists about the subsequent reduction of individuals' fundamental rights and freedoms, specially in those aspects related to privacy. This concern was materialized in the International Conference of Data Protection Authorities, held in London during 2006, where *"the need to adapt videosurveillance to the demands of the fundamental right to data protection"* was addressed. In this regard, many different states all over the world with a prior legal framework in privacy are implementing a series of measures aimed at legislating the framework where the activity of the aforementioned videosurveillance systems can be performed, and specifying the necessary conditions such that their activity can be carried out whilst protecting citizens' privacy rights. This section will be focused on the description of European and Spanish legislations concerning videosurveillance systems, as the European Union (EU) is one of the few domains in the world that currently has a comprehensive set of privacy legislations, with specific legislation for the handling of personal data [40]. As for the Spanish legislation, it represents one of the implementations of the recommendations set forth in the EU. Nevertheless, it must be noted that similar laws rule in most countries.

From a European point of view, Directive 95/46/EC [10] deals with the *"protection of individuals with regard to the processing of personal data and on the free movement of such data"*. This right to privacy is recognized in Article 8 of the European Convention for the Protection of Human Rights and Fundamental Freedoms, promoted by the Council of Europe Convention the 28th January 1981, for the Protection of Individuals with regard to Automatic Processing of

Personal Data, and in the general principles of Community Law. It is worth pointing out that, according to this directive, processing of personal data covers *"any operation or set of operations which is performed upon personal data, whether or not by automatic means, such as collection, recording, organisation, storage, adaptation or alteration, retrieval, consultation, use, disclosure by transmission, dissemination or otherwise making available, alignment or combination, blocking, erasure or destruction"*. Therefore, given this definition of *"processing of personal data"*, it is straightforward to see that videosurveillance systems are one of the many scenarios covered by the broad scope of this directive. Hence, all videosurveillance systems in the EU should be compliant with it.

According to the principles of protection derived from this Directive, the person/body responsible for processing personal data (the *controller*) must provide to the citizens information about its own identity, the pursued purposes with the processing of their data, and who is the recipient of such data. Furthermore, the citizens are entitled to: 1) know whether or not data relating to them are being processed; 2) know the logic involved in any automatic processing of data concerning them; 3) rectify, erase or block data whose processing is not compliant with the legislation in force. These principles of protection must be applied to any information concerning an identified or identifiable person. In addition, it must be also considered that *"Member States may, in the interest of the data subject or so as to protect the rights and freedoms of others, restrict rights of access and information"*. Furthermore, the Directive states that *"any processing of personal data must be lawful and fair to the individuals concerned whereas, in particular, the data must be adequate, relevant and not excessive in relation to the purposes for which they are processed,"* limiting in this way the personal data that can be manipulated by a videosurveillance system.

Nevertheless, all the previous provisions are subject to exemptions or derogation. For instance, the directive is not applicable when processing data is intended for journalistic purposes, or when concerning exclusively personal or domestic data, as well as in those activities regarding *"public safety, defence, State security or the activities of the State in the criminal laws."*

Last, but not least, the aforementioned Directive also indicates that Member States should establish supervisory authorities, which must help to ensure transparency of processing in their corresponding countries. In the Spanish case, this role is played by the Spanish Data Protection Agency (AEPD, Agencia Española de Protección de Datos), that in November 2006 proclaimed the Instruction 1/2006 [27], *"on processing personal data for surveillance purposes through camera or video-camera systems,"* which is aimed at ensuring that videosurveillance systems are compliant with the principles of the Organic Act 15/1999 *on Personal Data Protection* [11] (LOPD, according to its Spanish acronym), the national law that materializes the Directive 95/46/EC [10]. Most aspects considered in the LOPD are a direct translation of the principles established in the European Directive to the Spanish videosurveillance framework. Nevertheless, the Instruction specifies in some cases a further detailed implementation. Some of the additional constraints are:

- The controller's information duty binds to *"place at least one informative sign in the areas under videosurveillance, in a sufficiently visible location, in open as well as enclosed spaces."*
- *"The data will be cancelled within the maximum term of 1 month from being gathered."*
- *"The controller must take the measures of technical and organizational nature required to guarantee the security of the data and avoid their alteration, loss and unauthorised processing or access."*

Even when we have centered our attention on the European and Spanish regulations, they are indeed a representative example, and it must be noted that in most countries the legislation in force sets the basis for privacy protection in the videosurveillance scenario.

5.3. Prior art

Since the late 90's, privacy in the digital world has become a major concern. This section reviews the main initiatives regarding privacy management at a global level and in the particular case of the videosurveillance scenario.

5.3.1. Privacy management

So far, most of the work in this area has been put in the context of web services. Whereas a bunch of bundled commercial solutions for managing user privacy at the client side are available, solutions addressing privacy issues at a global level are not widely deployed yet.

Kenny and Korba described in [138] a global approach to privacy management that consists in adapting traditional DRM for building an architecture capable of satisfying the European Directive 95/46/EC for privacy protection. The proposed Privacy Rights Management (PRM) architecture is motivated by the inexistence of proper technological means for fulfilling the Directive requirements in web transactions, where users provide private data which is processed by another entity. The authors of [138] propose a client-server architecture with three key elements: 1) the data subject, which is the originator and owner of the private data; 2) the data controller, which is in charge of managing the collection, storage and processing of the private data, and is the ultimate responsible for the misuse of such data; 3) the data processors, which either may depend directly on the data controller or may be third entities.

The data controller manages a system comprising a web server for interfacing with the users, and a PRM server for managing the transactions with data processors and a set of databases. These databases contain the (cryptographically protected) private data of the users, logs on data use, information about the data processors and data subjects, and the rights database which defines the operations that can be carried out on the private data (in other words, the privacy rules). These rights can be conveniently expressed, as for any DRM system, in standard machine readable language using XrML [2], REL [237], etc. Thus, the private data in a PRM system plays the role of "asset" in classical DRM, which is protected according to the legislation in force (the rights). In addition, the data subject could issue a license by which he/she grants certain rights to the data processor in case certain conditions apply.

The authors show in [138] that the proposed PRM architecture fits well the privacy requirements of the Directive 94/45/EC. However, the proposal is not absent of drawbacks. One of the problems is the trustworthiness of the data controller, which processes the private data in the clear. Yet another serious issue, due to its client-server architecture, is scalability, important if a large scale deployment is to be made.

A more recent and popular initiative for privacy management is the Transparent Accountable Datamining Initiative (TAMI) [238], promoted by the MIT, which advances on the basis set by the P3P project [8]. TAMI advocates for transparency and accountability of the private data use, instead of restricting the access to private data. The rationale is that the World Wide Web is making more and more difficult to restrict access to decentralized data, and it is making it easier to aggregate data from multiple information sources, specially for data mining purposes. Hence, it is reasonable to devote efforts to the enforcement of fair data use. However, the problem in videosurveillance is different, as the scenario is bounded and well defined, thus making it easier to implement a DRM-like approach. Moreover, transparency and accountability can be ensured by means of secure event reporting. The TAMI project is particularly focused on large scale data mining. It proposes a "policy aware" architecture comprising rule languages that are able to express policy constraints, and reasoning engines able to produce inferences and proofs for private data use being compliant with the relevant rules.

5.3.2. Privacy in videosurveillance

Even if societal and ethical privacy concerns due to the rise of videosurveillance systems have been widely discussed during the last years [213],[156],[155],[14],[18], the problem has not been frequently addressed from the technological point of view. As recently noticed in [56], progressive advances in computing power and computer vision can help to achieve the right balance between privacy and security for videosurveillance. To the best of our knowledge, the first relevant attempt in this direction is due to the IBM Research Division, as described in a technical report in 2003 [200]. Prior to its publication, only a few works had proposed technological solutions for privacy protection in the videosurveillance scenario (see [200], Sect. 3.3).

In [200], the authors propose the use of computer vision in order to understand the captured video and hide the sensitive data at different levels, according to the privileges defined for different users of the system. These privileges are to be defined considering the different classes of users of the system, which is dependent on each particular scenario. In an illustrative use case, three different classes of users are envisaged: anonymous (have access only to statistics collected by video analysis engines), privileged users (can watch video but with certain sensitive information hidden), and superusers, such as law enforcement officers (can watch the whole video without restrictions).

The privacy-preserving viewing of the video is ensured by means of a secure video console that re-renders the raw video, producing a new video stream where private data is hidden or obfuscated, whilst keeping the necessary information such that a human operator can evaluate the scene. Proper access control mechanisms guarantee that each user class has access only to the apropriate video stream. Thus, a security guard for instance could watch the video but with the faces of the recorded people erased or downsampled in order to make idenfication of the individuals impossible or at least very difficult. "Smart" video engines can detect different classes of objects (e.g. car plates), situations (e.g. accidents, fights, etc.) and individuals. In case the video engine is equipped with facial recognition capabilities, it is even possible to define different actions for the different individuals, provided the latter have been previously enrolled in the system.

The work described above gave rise to a significantly large number of papers, but mostly (or exclusively) dealing with implementations of video engines for detecting private data in the video stream and/or re-rendering raw video in a privacy-preserving manner. Some of these works will be briefly reviewed in Section 5.5.1.

5.3.3. Automated event control

Albeit automatic scene understading has been proposed as a tool to enable privacy-enhanced videosurveillance [200], up to now it has been raising more concern than relief, as it simplifies the collection and analysis of sensitive information on individuals.

Currently, a large number of tools for automating event control in videosurveillance, based on computer vision and image understanding, are being developed or already commercialized. Despite the large availability of tools of this kind, their massive deployment is still being hindered, not by the aforementioned privacy concerns, but rather by interoperability issues. In this context, a consortium originally formed by Axis Communications, Bosch Security Systems and Sony Corporation has began to promote the Open Network Video Interface Forum (ONVIF) standard [19]. More than 60 partners, including major actors in the videosurveillance industry, have already joined the initiative. ONVIF is aimed at developing an open standard for the communication between network video clients and video transmitter devices, giving a solution to the interoperability problem. The current ONVIF specification covers aspects such as device management, audio and video streaming, event management and video analytics, with all the interfaces described as web services by means of well known standards like XML, SOAP, and the Web Service Description Language (WSDL).

ONVIF defines two main architectural elements: the Network Video Transmitter (NVT), and the Network Video Client (NVC). The NVT is a device that sends video over an IP network to an NVC, so it plays the role of "service provider." On the other hand, the NVC is a controller device that communicates with an NVT, thus playing the role of "service requester." ONVIF gives support to JPEG, MPEG-4, H.264, G.711, G.726 and AAC codecs for video and audio streaming. The architecture of a video analytics application is composed of two main modules:

1. A video analytics engine which receives a video stream and produces a Scene Description, i.e. an abstract representation of the observed scene in terms of the objects present and their behavior.

2. A rule engine which contains the set of rules that govern the allowed actions in the observed scene (for instance, a certain virtual perimeter must not be crossed by pedestrians), the allowed intra-object relations (e.g. a person who lifts his/her bagage in the airport), and the allowed object behaviors (such as a maximum speed limit). The comparison between the Scene Description and the rules produces an event.

Clearly, ONVIF can have a great impact in the development of incoming privacy-

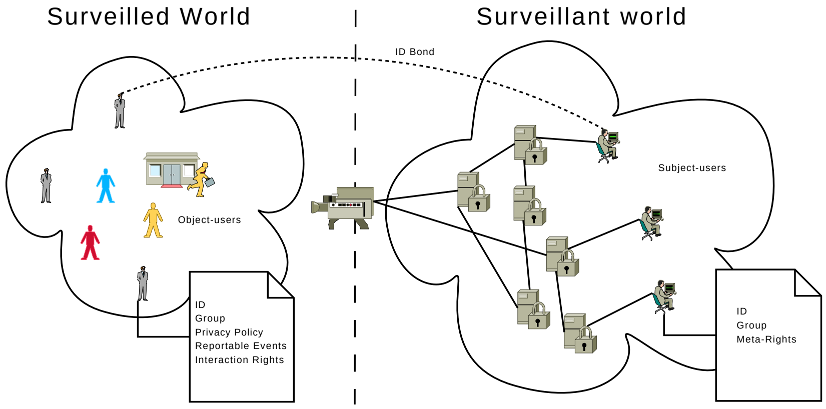


Figure 5.2: Architecture of a videosurveillance system using DRM

preserving videosurveillance systems, as it opens the door to standard and interoperable management of sensitive events and privacy rules.

5.4. Rights management: a global solution for videosurveillance

DRM has traditionally been used for protecting the rights of content creators. As pointed out in [138], due to the duality between rights management and privacy protection, DRM can also be applied, with some considerations, to privacy protection. The proposed solution goes even further. We have already anticipated that in a videosurveillance system, automated surveillance and privacy are closely related; thus, both aspects should not be considered independently, but jointly in a complete approach like the presented one. With a slight redefinition of some concepts, DRM can cope with all the aspects of videosurveillance, ranging from automatic event reporting to security of transmissions and storage, hierarchical access control for authorized users, and different levels of privacy protection for the surveilled users. At the same time, our proposal circumvents the problems presented by other individual privacy or automated surveillance systems (cf. Section 5.3) in terms of scalability, trust support or flexibility, always complying with the current regulations.

In the following we describe the proposed architecture, depicted in Figure 5.2, of a videosurveillance system using DRM for providing both privacy and automated surveillance.

5.4.1. Object-users and Subject-Users

A videosurveillance system conceptually divides its users in two categories:

- **Subject-users**: We will call subject-users to those agents that have access to data generated by the system, and can perform actions on these data. In order to access the system, they must authenticate themselves. In a DRM system, these users are represented as content consumers and adapters, that have certain access rights to the contents in the system, specified by the licenses of those contents. The information that these users produce is limited to event reports generated by the actions they perform, and contents adapted from existing contents in the system (when they have the right to produce them), but these users cannot generate new contents by themselves.

- **Object-users**: We will call object-users to those watched items (objects, people or regions), whose actions are surveilled and generate data. The data generation is initiated automatically when one of these users is recognized by the system in the contents generated by a camera; there is no authentication for these users besides the automatic recognition performed by the system; thus, there must be at least one object-user or group of object-users to which the detected but unrecognized users are mapped; it is also possible to automatically enroll every unrecognized user with a generic *unidentified* profile, in order to allow for relative ID recognition.

  Every object-user has an associated *virtual* content, that represents the object-user as an element of the DRM system with which other object-users may interact. It contains no resources, but only license and reporting information, (the *interaction rights* defined in Section 5.4.2). In the DRM system, these users are identified with content creators, and their associated *virtual* content is considered just as one ordinary content.

Informally, subject-users are those who can operate some part of the system, including employees of the enterprise, security guards, law authorities, etc. On the other hand, object-users are those in front of the cameras, that are being surveilled. Both categories of users must be uniquely identifiable by the system, and even when there is no a priori direct relation between them, the system may keep unique bonds between a determined subject-user and an object-user, that link both users under a unique system identity. For example, a security guard can be a subject-user, for he can have access to data in the system, but when appearing in front of the camera, the guard becomes also an object-user, although both users are bond to the same identity. These links allow for the assignment of access rights of a subject-user to the data generated by the linked object-user; thus, it solves the problem of automatically providing a recognized object-user access to his/her own data without the need of an external process that determines the relationship between the data and the authorized subject-user; this would effectively implement the data access right required by Directive 95/46/EC. Nevertheless, these links constitute only identity links, and access rights can be defined elsewhere, as will be shown later on. This is another example of the need of taking into account the dependency between privacy and automatic surveillance.

5.4.2. Roles and privileges

As we are dealing with a global solution that covers all the aspects of videosurveillance through a DRM system, we must take into account two complementary questions:

- **Privacy**: When taking into account the privacy concerns related to videosurveillance, it is customary to protect the information of the object-users that are being recorded, and allow access to this information only to authorized subject-users. This implementation through DRM would improve on the typical access control lists commonly proposed for privacy protection, providing a more flexible access system and different access levels. Regarding privacy, it is also desirable to implement different privacy profiles, such that each user has relative freedom to choose how, when and against whom his/her data must be protected, and whether or not the object-user wants to be informed whenever some subject-user accesses his/her data. On the other hand, subject-users generate only reports on their activity on data previously created by object-users; thus, the privacy of subject-users is taken into account by granting them privileges in order not to report certain actions; for example, a representative of the law authorities may have visioning access to some recorded scenes, and this event should not originate reports to the involved object-users.
- **Event control**: An automated surveillance system must be able to detect certain events and inform to the appropriate agent(s) of the system about their occurrence. In this way, the system must define which actions of object-users must be subject to event control, and which subject-users must be informed about those actions; this should be either a generic policy, affecting all object-users, or a specific policy, affecting a limited group of users (or only one user). E.g., an enterprise may want that several members of the "security" group be informed about determined events happening between object-users; this would define a policy affecting all the involved object-users.

This two-fold approach can be materialized through the application of DRM with the following mapping:

- **Object-users**: For every object-user, the following elements will be defined:

  - **Privacy policy**: The object-user, as a content generator of the DRM system, must have a data creation policy (a template for content generation and for the associated licenses), that defines:
    - The access rights that each group of subject-users has over each piece of information generated by the object-user in question.
    - A selection of tools used to secure the generated contents. This selection will represent the way the data will be protected, and therefore, the privacy level assigned to the user; they can comprise, but are not limited to: distortion, masquerading, substitution, encryption or even total elimination; this last choice would represent the level of total privacy, for which unauthorized users would not have access to the data and also to knowing whether some data is present or not. More about total privacy is discussed in Section 5.5.
  - **Reportable events**: A list of actions (record, play, copy, encapsulate,...) on the data generated by the object-user that must generate event reports, and to which subject-users these reports are sent.
  - **Interaction Rights**: The surveillance system may be able to detect interactions between object-users (i.e., a person entering a restricted access zone, a person leaving a case on the floor, a fight,...). Each object-user or group of object-users may have a list of allowed/forbidden interactions, that will define which interactions must be recorded as an event report, which the level of those reports is, and to which subject-user(s) those reports will be sent. This rights are defined in the *virtual* content associated to each object-user.
  - **Group**: Every object-user should belong to one of the defined groups, as Interaction Rights are more efficiently described when specifying a group and an object than when specifying pairs of objects. Additionally, groups of object-users may have default privacy policies. Grouping is a critical aspect to allow for scalability of the system.

- **Subject-users**: Subject-users have their privacy taken into account through restrictions on the reported events that their actions generate, granting them privileges in order not to report certain actions.

  As well as object-users, subject-users can also be grouped, as there may be some generic classes of users (like representatives of the law, security professionals, regular employees,...), with default profiles and different privileges. Additionally, subject-users may also have meta-rights over virtual contents, being able to issue or revoke the access rights of some object-user or group of object-users to certain virtual contents.

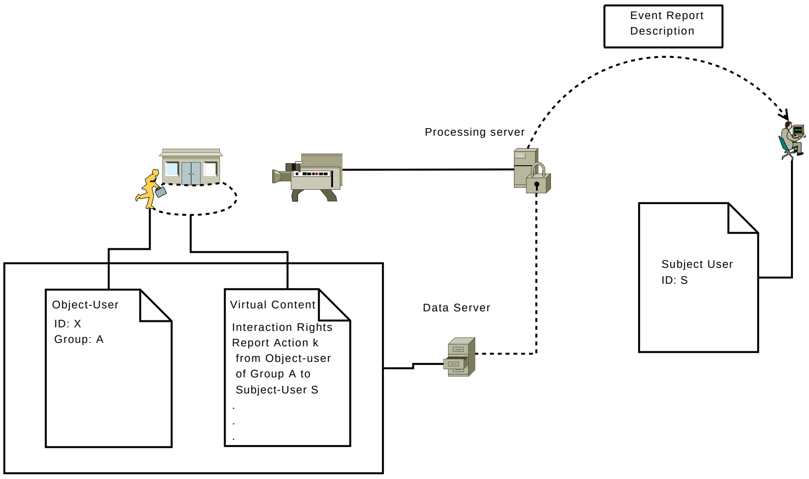


Figure 5.3: Architecture of event reports using DRM

5.4.3. Event Management

As highlighted in the previous sections, there are two types of events: those produced by the interaction between object-users, and those produced by the actions performed by subject-users on system data. The former will be called *surveilled events*, while the latter will be called *privacy events*. The reports that these events generate will always have one or several subject-users as recipients. Nevertheless, all of them can be handled in a unified way by a DRM system supporting event reporting or metering functions.

Events are actions that the system can identify. *Privacy events* are defined by typical access rights of subject-users to ordinary contents of the system. This is directly interpreted as traditionally done in a DRM system. On the other hand, *surveilled events* are defined by access rights of object-users to *virtual* contents. Thus, the surveillance system must identify *objects* in the surveilled scene, map them to their corresponding object-users and virtual contents in the DRM system, and map the interaction among objects to actions performed by object-users on virtual contents; then, the virtual contents will indicate which of these actions must be reported as events.

5.4.4. Surveillance Ontology

As shown in the previous sections, the proposed framework can cope with both sides of a videosurveillance system: the world behind the cameras, and the world in front of them. Thus, the ontologies typically associated to DRM systems are not enough to describe all the agents and relationships that are present in our framework, such as the interaction rights. Therefore, an extended ontology is needed to cope with the concepts and the bonds that are present in our framework.

5.5. Implementation guidelines: a high level perspective

5.5.1. Prior works

Most works on videosurveillance privacy are very recent, and most of them are focused on particular modules of the global privacy management architecture, especially on: 1) engines for automated detection of objects of interest; and 2) methods for protection (including blurring and encryption) of the private data.

As for the first module, the objects of interest are usually chosen to be persons and moving objects such as vehicles. The main reason is simplicity, since the considered scenarios usually have static videosurveillance cameras: hence, moving object detection can be accomplished by means of simple background segmentation techniques [57], [87], [247], [212]. More sophisticated approaches for object detection are based on trained classifiers, such as [152], [157], [60]. These approaches allow for better control and decision making, but at the cost of increased computational complexity. Nevertheless, a bunch of efficient detectors are described in the literature and are successfully used in real scenarios, like the method by Gavrila and Philomin [102], the well-known Viola-Jones method [236], or the HOG method [75].

Regarding the protection of private data, we can find works such as [171], [60], [212], [247], where a variety of non-invertible transforms are applied to the sensitive data: obscuring, pixelization, blurring, silohuetting, face masking, etc. In other works, the application of standard encryption methods (e.g. AES, permutation-based encryption) is proposed in order to ensure recoverability of the private data if the viewer has access to the appropriate key. Some works simply apply encryption to the raw bits comprising the detected objects of interest [57], but others resort to layered encryption techniques combined with perceptual coding, in order to provide different privacy levels in a natural manner. Two examples can be found in [87], [151] for Motion JPEG-2000 video. Other works on layered encryption can be found for H.264 [178], DCT-based [93], Hierarchical MPEG [122], and scalable codecs in general [94]. However, the four latter works are not directly applicable to the videosurveillance scenario, as no objects of interest are considered (the whole image is encrypted).

In general, the usability of the proposed encryption algorithms is strongly dependent on the video codec used in the videosurveillance system (especially in layered encryption techniques). In [55], permutation-based image encryption is proposed in order to achieve codec independence. Furthermore, this encryption method allows for a certain transcoding of the video without completely destroying the encrypted information if afterwards recovered using the proper secret key.

The system performance has been evaluated with several video codecs, including MPEG-2 and H.264.

In view of the existing work, it appears that no global framework for a practical implementation of privacy management solutions exists for the videosurveillance scenario. Our proposal in this direction is introduced below.

5.5.2. A generic, standards-compliant implementation proposal

At a high-level, our implementation proposal can be viewed as a generic video analysis system coupled with a DRM system. It is basically a combination of MPEG-4, MPEG-21 and ONVIF standards, in such a way that all aspects of the proposed architecture and functionalities are satisfactorily covered:

- MPEG-4 covers aspects related to object management and cryptographic protection of the sensitive data (IPMP);
- MPEG-21 provides standard means for defining licenses on data use and a language for expressing rights (REL), as well as the format for event reports and their requests.
- ONVIF provides event management and transmission capabilities.

The different components of our proposal are described below. It must not be understood in any way as a restrictive implementation, but just as an illustrative implementation which is generic enough and covers all aspects of the proposed architecture using standard technology.

5.5.2.1. Video segmentation, encryption and encoding using MPEG-4

Unlike MPEG-2 and MPEG-1, MPEG-4 strongly relies on the concept of object. This is one of the characteristics that makes it specially amenable to the implementation of our proposal. Whereas an MPEG-2 program is typically formed by two audiovisual elements (one full-screen video stream, and one audio stream), MPEG-4 content may be built of an arbitrary number of audiovisual elements, called *objects*, that belong to a wide range of defined object types, such as rectangular video, video with shape, synthetic face or body, speech, synthetic audio, text, or graphics. The basics of the object encoding mechanisms available in MPEG-4 are briefly explained in the following.

As illustrated in Figure 5.4, the access to MPEG-4 contents starts with an Initial Object Descriptor (IOD). This IOD points to at least two basic streams:

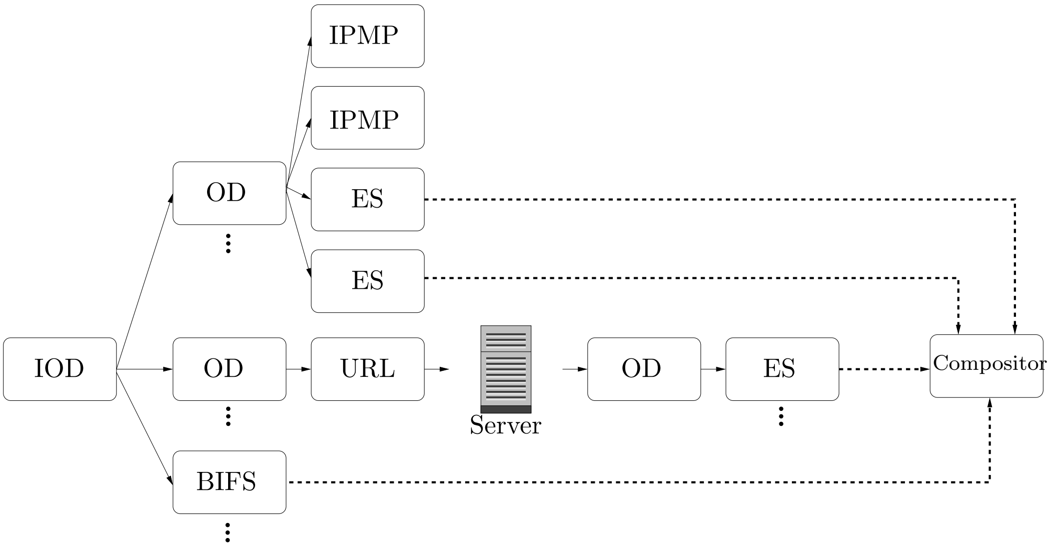


Figure 5.4: Schematic description of MPEG-4 contents based on objects.

a scene description (BInary Format For Scenes, BIFS), and an Object Descriptor (OD) stream. The OD is a kind of container aggregating all the useful information about the corresponding object. An OD can contain a URL pointing to a media stream, or a series of subdescriptors. These subdescriptors contain pointers to individual Elementary Streams (ESs), semantic information about an object, and pointers to contents access management information (Intellectual Property Management and Protection, IPMP). IPMP descriptors and IPMP streams specify a means for decrypting ciphered ESs, or for checking authorization or entitlement information. It is worth pointing out that a single visual or audio object can be coded into one or more ESs.

OD streams are usually associated with a scene description (BIFS) stream; the scene description conveys the spatio-temporal layout of the media objects in the scene, i.e., it indicates how to assemble the various media streams described within the OD stream. ODs and BIFS are associated through another OD. Based on BIFS, a visual scene in MPEG-4 is described as a composition of Video Objects (VOs) characterized by their shape (not just rectangular, but arbitrary shapes can be considered), motion, and texture. Each VO can consist of one or more layers (VOL) which can be used to enhance the temporal or spatial resolution of a VO. An instance of a VOL at a given time instant is called a Video Object Plane (VOP).

Due to its hierarchical structure, its modular nature and the diversity of its encoding tools, MPEG-4 indeed provides many degrees of freedom for producing a video sequence. This high number of possibilities can be effectively exploited in our framework for videosurveillance.

As explained before, the main interest is in hiding certain parts of the video stream, depending on the rights of the viewer and the licenses associated to the objects in the image.[1] MPEG-4 allows to define a bottom-to-top video structure, where it is possible to separately encode and protect the objects of interest, such that they can be a posteriori selectively extracted and reproduced, according to the relevant rights and licenses. An illustrative example of these capabilities is the following.

1. In the videosurveillance scenario, the first object of interest to be defined is the background image, usually static, being captured by the camera. For still cameras, this background image can be easily defined. For motorized remotely controlled cameras, a reference background image can be also defined for each possible positioning vector (including azimuth and elevation parameters) and configuration parameters (focus, shutter time, resolution, etc.)

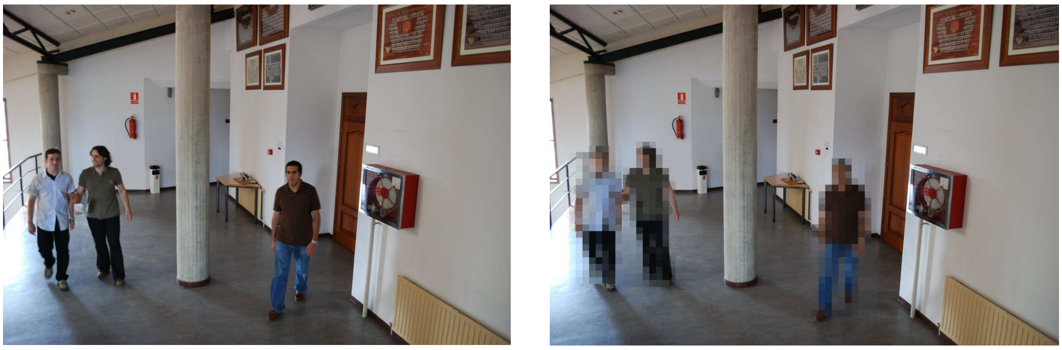
2. The background image plays the role of canvas over which the moving objects (e.g. persons, vehicles) are superimposed in order to compose the complete scene consisting of a set of layers (VOL).

3. The viewer (subject-user) can check in the IPMP descriptors whether he/she has the rights to visualize the objects in the scene (in the clear). If that is the case, the corresponding VOs and VOLs will be conveniently decoded, decrypted and placed in the scene thanks to the BIFS capabilities. On the contrary, for the subject-user lacking the necessary rights, the system would render a processed version of the protected objects over the background image. This processing may encompass partial decryption, blurring, transparency (even total erasure), substitution by a synthetic face/body, etc., according to the degree of privacy required and/or the rights of the subject-user. An example of the output presented to a subject-user is shown in Figure 5.5.

This conditional access problem can be solved by means of traditional DRM tools applied on the media streams needed for the composition of the final video. The application of DRM can be performed at different points of the MPEG-4 structure.

- A simple approach would consist of encrypting the ESs corresponding to the media objects representing the object-users of interest, and sending

[1] Note that the MPEG-4 objects of interest in our scenario are content media generated by object-users (i.e., multimedia information containing the image/voice of the object-user), so the fact that a given subject-user has the appropriate rights to access the data will not just depend on the subject-user group and the corresponding rights, but it will also depend on the object-user privacy policy.



(a) Video frame in the clear (b) Privacy-protected video frame

Figure 5.5: Example of privacy-preserving videosurveillance by means of person detection and pixelization.

them by default in a stream available to all the subject-users. One obvious drawback is the significant communications overhead (the encrypted data is useless for unauthorized viewers), and the possible privacy leaks because of the mere awareness that certain information is being hidden.

- A more sophisticated approach would be the use of encrypted URLs in the ODs pointing to the DRM-protected streams, that could be even stored in a different video server. This way, a twofold objective would be achieved: 1) unauthorized users would not download data they would not be able to process, and 2) privacy would be improved to a larger extent, as unauthorized users will know that some encrypted URLs are present, but they will not be able to know exactly how many there are. Furthermore, strategies could be designed where a media object were partitioned into multiple ESs, each of them DRM-protected and stored in a different server; doing so, not even hacking a server would be enough for accessing the protected data, as the set of all the servers containing data of a multimedia object should be cracked. Therefore, the privacy and security levels provided by this approach would be significantly higher than for the privacy-enhancing videosurveillance systems proposed so far.

5.5.2.2. Licensing, rights and event management implementation via MPEG-21 and ONVIF

MPEG-21 provides a rich and extensible language for defining user rights and allowed uses of digital contents, specified as a Rights Expression Language (REL). These rights are introduced in licenses that, when associated to a Digital Item (DI), specify the actions that determined users can perform on the resource linked to the DI. The rights defined in MPEG-21 REL can be directly mapped to the actions (play, record, copy,...) performed by subject-users on the contents generated by the videosurveillance system; nevertheless, the REL must be extended in order to include the actions that object-users perform on virtual contents, e.g., enter, leave, take,... (cf. Section 5.4). MPEG-21 REL allows to easily define these extensions.

On the other hand, the ONVIF standard provides a set of security operations for configuring NVTs including, among others, setting access security policies, and handling user credentials and settings. The security model in ONVIF ([16]) defines a standard set of SOAP extensions, and includes a Token Profile based on REL ([15]), considering four different user levels: Administrator, Operator, Media user, and Anonymous. The access security policy of each of these groups can be defined through REL. Thus, both standards can complement each other within an implementation of our framework: while MPEG-21 REL provides the standard language for defining user rights and access privileges, ONVIF would provide standardized protocols for the protection of these rights and privileges.

Event handling in ONVIF is based on the OASIS WS-BaseNotification and WS-Topics specifications [7], that define event handling principles, basic formats and communication patterns; however, the standard does not require particular notification topics, and it defines a set of basic notification topics that an NVT is recommended to support [19]. Likewise, MPEG-21 Event Report (ER) defines the format of the Event Report Request (ERR) and Event Report (ER) messages. It must be noted that both standards do not collide also in the specification of event handling, as MPEG-21 ER deals with report formats, while ONVIF defines the layer for communicating generic reports (Real-Time and non Real-Time). Thus, turning again to our framework, a natural implementation of event reporting would consist in using MPEG-21 formats for ER and ERR, embedding the latter into the corresponding DIs, and encapsulating the former inside ONVIF messages, in order to be simultaneously compliant with both standards.

5.6. Use case description

This section describes by example the application of the videosurveillance model presented in Sect. 5.4 to a real scenario. It is focused on a videosurveillance system for airports because such scenario encompasses a wide range of users and situations that illustrate well the capabilities of the proposed approach.

5.6.1. Users

According to Sect. 5.4.1, we distinguish between object-users and subject-users. Object-users are those individuals/entities in front of the cameras, and some examples are

1. public security forces (police and similar),
2. private security agents,
3. airport assistants,
4. airlines desk staff, airlines on-board crew,
5. duty free, bars and restaurants or cleaning staff,
6. airport users (e.g. passengers),
7. baggage items and trolleys,
8. security control trays,
9. restricted areas.

On the other hand, subject-users are those who operate the videosurveillance system. In this case, it would correspond to user types 1 and 2. Only user types 1-6 may define how the information generated by them is processed, while user types 7-9 will be assigned a generic or specific policy.

The classification of a given object-user in one of the aforementioned categories requires a previous enrollment and/or system training. In the case of items or areas (user types 7-9), the videosurveillance system needs a training phase to understand what kind of object it has to look for, and sometimes intervention of a subject-user (to define the perimeter of a restricted area, for instance). For many human object-users (types 1-5), enrollment is already usually required in current airport security systems. It is not necessarily the case for passengers and other airport users, who can anonymously move around many areas inside the airport. Thus, they would belong in general to the class of *unidentified* object-users. Nevertheless, the fact that a passenger does not follow a typical enrollment process does not imply that he/she can not be tracked all over the airport. The videosurveillance system could have a database containing information about the observed passengers in the last, say, 24 hours. This way, a high level of security can be achieved whilst complying with the proportionality principle.

Notice that human object-users of types 1-5 are usually required to bear an identity card, which can be equipped with an RFID device, and hence used for verifying unobtrusively their identity against a biometric recognition system integrated in the videosurveillance network.

5.6.2. Privacy policies

According to Sect. 5.4.2, object-users can define through their privacy policies how their information is managed by the system and accessed by others. Some examples are given here:

1. A shop assistant in a duty free store can request to make his/her image unidentifiable to private security agents operating the videosurveillance system while he/she is at work inside the area of the store facilities.
2. A police officer can request his/her image to be completely removed for any subject-user, unless it is another police officer. In addition, he wants his private data to be protected with RSA encryption.
3. Default rules can be applied to unidentified object-users, such as passengers and other airport users not enrolled in the security system.

These privacy policies will be eventually reflected on the access rights that are granted to subject-users, depending on their specific function at the airport. For example, subject-users of types 1-2 have access to most of the information that is processed/recorded by the videosurveillance system. On the other hand, subject-users corresponding to object-user types 3-6 (or even user types 1-2 not in charge of operating the videosurveillance system) should not have access to these data.

5.6.3. Interaction rights and automated event control

According to Sect. 5.4.2, object-users have predefined rights that control the allowed interactions between them. The automated videosurveillance system can detect whether these interaction rights are being violated in order to appropriately generate event reports. Some examples are:

- A piece of baggage (type 7 user) remains unattended for a long time. The system tags this item as “unattended baggage” and sends the correspoding report to a security officer.
- A type 6 user picks up a piece of baggage that had been tagged as “unattended baggage”. This could indicate a theft, so the appropriate report is generated and sent again to the security officer.
- A type 6 user enters a restricted area (type 9 user). The system checks whether the former has the right to access this area; if not, the action is notified via an event report.

5.7. Conclusions and Further Work

Individuals demand for technical ways of improving their personal security that do not hinder their right to privacy, which is granted by the European legislation in force. As far as videosurveillance is concerned, the DRM-based architecture that has been proposed in this chapter comes to provide a good balance between security and individuals' privacy. In fact, as presented, DRM can solve the critical issues of a current videosurveillance system, considering their twofold nature: covering and standardizing the automation of the surveillance activity, while putting in the hands of the users the appropriate technical means to control the access to their private information. Thus, the way DRM is used in our framework can lead to an increased acceptance of videosurveillance, as its target is the protection of the final user.

Note that this chapter has dealt only with conceptual elements of a videosurveillance network, without taking into account how they can be mapped to physical devices. The most straightforward solution is to include in the network dedicated processing nodes, which can provide the necessary analysis and automated decision functionalities. On the other hand, the computation power of IP "smart" cameras is rapidly increasing, so it is foreseeable that videosurveillance cameras will soon become autonomous devices capable of performing complex video processing operations. This will ensure that many of the functionalities envisaged by the presented architecture will be directly realizable in the cameras. Additionally, the more powerful the analysis engines run by the system, the more granular and reliable it will be.

Finally, it has been shown that the presented architecture can be implemented through the use and adaptation of current standards, like ONVIF, MPEG-21 and MPEG-4. This constitutes a clear advantage, in the sense that a standards-compliant solution provides more generality, transparency and availability.

Chapter 6

Fully Private Noninteractive Biometric Authentication

Face recognition is one of the foremost applications of image processing, that often deals with sensitive signals; privacy concerns have been lately raised and tackled in several recent papers dealing with privacy-preserving face recognition systems. Nevertheless, the presented systems either use the knowledge of some information derived from the database templates in order to perform the recognition or require several interaction rounds between client and server.

In this chapter, we present a private system that can cope with a simple verification algorithm executed in the server without interaction, in which both the templates and the queried face are encrypted. In order to achieve this, we make two significant contributions which must be combined to reach a fully non-interactive solution: on the one hand, we use a feature model based on circularly symmetric and Generalized-Gaussian marginally distributed real and imaginary parts of Gabor coefficients driving an efficient Lloyd-Max quantization of Gabor coefficients magnitude, combined with an SVM classifier; the goodness of fit of the probabilistic model for Gabor magnitudes is assessed through Kullback-Leibler divergence. This allows for a great reduction of both storage and plaintext cardinality.

On the other hand, we also provide an extension of a quasi-fully homomorphic encryption, that combined with the small cardinality plaintext of quantized indices of Gabor coefficients, is able to compute the SVM's soft scores operating on all the input parameters, features and templates in encrypted form, and without interaction with any of the clients. We show its performance in terms of time complexity and size of transferred encryptions, as well as in verification accuracy with respect to the non-private system. The combination of these two contributions opens the door to completely private and noninteractive outsourcing of face recognition.

The work in this chapter has been partially presented at IEEE ICASSP 2010 [215] and submitted to IEEE ICIP 2012 [217] and IEEE TIFS 2012 [216]; some of the technical developments have been filed as patent applications (Patent pending, Application No. 61/596151)

6.1. Introduction

Face recognition is an important and active area of research [250], whose interest has increased in recent years because of theoretical and application-driven motivations. Nevertheless, it is also a prototypical image processing application where privacy constraints come into play, due to the sensitivity of the involved biometric signals. In a common privacy-aware face recognition scenario, a user presents his/her face for matching against a database of enrolled clients, to verify a given identity; the database must not be disclosed to the new user, as this would harm the security of the system and the privacy of the enrolled users, while the face presented by the query user must not be disclosed to the recognition system, for preserving the user's privacy. There have been several recent proposals of efficient privacy-preserving solutions for this scenario, combining additive homomorphic encryption and garbled circuits, like Erkin *et al.* [89] or Sadeghi *et al.* [196], both focused on private face identification using a simple but effective recognition system, called Eigenfaces [231]; the latter is based on applying a PCA projection matrix to the presented face.

However, this *traditional* scenario does not protect the privacy of the enrolled users, as the recognition system must have clear-text access to the templates stored in the database and to the projection matrix. More involved scenarios, like outsourced ones, where Clouds or other *untrusted* environments are used not only for storing the databases but for performing certain operations, are becoming increasingly ubiquitous. If the matching database is stored in an untrusted third party together with the detection logic, enrolled users' privacy must also be protected, and that party must have access neither to the database contents nor to the fresh faces presented against the system for recognition. Additionally, it is desirable that the system can run autonomously without interaction rounds with the client, requiring the lowest computational effort from the client-side, that is usually executed on an embedded or mobile device.

This chapter tackles this privacy-aware scenario, where we aim at face verification in an outsourced system that works with a fully encrypted template database and query faces (total privacy) and provides a verification result without interaction with the client. That purpose can only be achieved by combining the two essential elements that we provide: a quasi-fully homomorphic extension of Gentry's fully homomorphic cryptosystem [105], and an efficient quantization system for Gabor features that allows for a great plaintext cardinality reduction; these two elements joined together enable the implementation of the noninteractive private system, whose performance is evaluated in the envisaged biometric scenario. This opens up a wide new set of applications, and provides a first stone for a fully private noninteractive outsourced processing in untrusted environments.

The rest of the chapter is organized as follows: Section 6.2 reviews the GG distribution used to model real and imaginary parts of Gabor coefficients; Section 6.3 introduces and evaluates the used statistical model for Gabor coefficients magnitude, as well as describing coefficients' quantization using this model. Section 6.4 reviews Gentry's Fully Homomorphic cryptosystem; and presents the proposed extension with a lower bound on the number of achievable sequential homomorphic multiplications. Section 6.5 presents the application to a fully-private noninteractive face verification scenario joining together the efficient quantization system and the extended encryption, and evaluates its performance figures in widely known test databases (XM2VTS [160] and LFW [125]), comparing also the clear and the fully-private system. Finally, Section 6.6 discusses the security aspects of the extended cryptosystem, and Section 6.7 draws some conclusions.

6.2. Face Features and Existing Models

Gabor filters have received great attention due to biological reasons (Gabor filtering effectively emulates that performed by the Human Visual System, HVS) and because of their optimal resolution in both frequency and spatial domains [81]. As a matter of fact, there is a large number of publications that have adopted such features for face processing, including [241, 194, 183, 184, 172, 242] (see [204, 201] for recent reviews on the use of Gabor filters for face recognition).

One of the main drawbacks [204, 114] of Gabor-based approaches is the huge amount of memory that is needed to store a representation of the image. One possible solution to this shortcoming is to quantize the input data by taking advantage of accurate statistical models. In this direction, Generalized Gaussian (GG) distributions have been proposed to model both real and imaginary parts of Gabor coefficients extracted from face images [114]. Empirical validation has confirmed that these densities provide an accurate characterization for the distribution of the input data.

In this chapter, we go one step further, trying to reduce even more the length of the representation needed for an efficient recognition using Gabor features, due to the cardinality requirements that the encryption system presented in Section 6.4.1 poses. In order to minimize the volume of data, we use a novel statistical characterization to model magnitudes of Gabor coefficients [215] (most

Gabor-based face recognition algorithms discard phase information [241]), under the assumption that both real and imaginary parts are GG distributed and the complex coefficient has circular symmetry, and we propose two different quantizations, using levels and indices (cf. Section 6.3.2), to minimize the representation length.

The fitting accuracy of the proposed model to the data is evaluated using the Kullback-Leibler divergence on two different datasets: XM2VTS [160] and Labeled Faces in the Wild (LFW) [125] databases. After assessing the quality of the fit, we apply the proposed statistical model for performing data compression via Lloyd-Max quantization, that achieves minimum mean squared error (MSE) for a given number N_L of representative levels; the use of this model greatly reduces the size of a Gabor-based face representation with a negligible impact on recognition performance, as it will be shown.

6.2.1. Generalized Gaussian Distribution

In this chapter, we are interested in zero-mean Generalized-Gaussian variables, whose pdf is given by the following expression

$$f_{GG}(x) = \frac{\beta \cdot c}{2\Gamma(\frac{1}{c})} \cdot e^{-|\beta x|^c}, \quad \beta = \frac{1}{\sigma}\sqrt{\frac{\Gamma(\frac{3}{c})}{\Gamma(\frac{1}{c})}}.$$

This characterization has two parameters:

- β represents a scale parameter inversely proportional to the standard deviation σ of the variable,
- c is the shape parameter, that controls the weight of the tails and the peakedness of the bell. Particular cases are the Laplacian distribution when $c = 1$, Gaussian distribution when $c = 2$, and Uniform distribution when $c \to \infty$; nevertheless, for the cases of interest, c will be within the interval $c \in (0, 2]$.

Usually, Generalized Gaussian distributions are employed to model peaky and heavy-tailed random variables, for which it yields good fits; examples of GG modeled variables can be found in coefficients of many transforms, like DCT or Wavelets [120, 85, 203], and, especially, the marginals of Gabor coefficients [114].

A previous approach to modeling Gabor coefficients magnitude was proposed in [115], through a generalization of the Rayleigh distribution: in the same way as the Generalized Gaussian adds a degree of freedom in the exponential decay of the tails (the shape factor c), the β-Rayleigh distribution presented in [115] generalizes the Rayleigh distribution with a shape factor β

$$f_{\beta-\mathrm{Rayl}}(x) = \frac{\beta\left(\frac{\sqrt{\Gamma\left(\frac{3}{\beta}\right)}}{\sqrt{\Gamma\left(\frac{1}{\beta}\right)}}\right)^{\frac{2+\beta}{2}}}{\sigma\Gamma\left(\frac{2+\beta}{2\beta}\right)}\left(\frac{x}{\sigma}\right)^{\beta/2} e^{-\left(\frac{x}{\sigma}\right)^{\beta}\left(\frac{\Gamma\left(\frac{3}{\beta}\right)}{\Gamma\left(\frac{1}{\beta}\right)}\right)^{\beta/2}}.$$

Unfortunately, the β-Rayleigh distribution for the magnitude cannot be obtained from GG marginals, so this model misses a connection with current GG models, that assume GG distributed real and imaginary parts.

6.3. Theoretical Model for the magnitude of Gabor Coefficients

Let $g_i \in \mathbb{C}$ be one of the Gabor coefficients extracted from a face, and $gr_i, gi_i \in \mathbb{R}$ its real and imaginary parts, respectively. As already shown in [114], both real and imaginary parts follow Generalized Gaussian marginals with the same parameters (shape factor $c < 2$ and standard deviation σ). Nevertheless, we have observed that the phase of g_i is approximately uniform, meaning that the distribution of each g_i presents circular symmetry. Actually, independent bidimensional generalized Gaussian variables are not circularly symmetric (unless they are Gaussian, $c = 2$), and consequently gr_i and gi_i are not independent. In order to assimilate this dependency, we propose a model with the following characteristics:

- The variance of each coefficient is not constant among different locations and subjects, but for the same location and subject it is the same for both real and imaginary parts.
- When conditioned to a given variance, real and imaginary parts of Gabor coefficients become locally Gaussian and independent, and thus, circularly symmetric.
- The resulting marginal distribution of the real and imaginary parts (for any location and subject) follows a GG law.

That is, for each coefficient G_i, we have $G_i = (C_i + j \cdot D_i) \cdot S_i$, where C_i and D_i are two independent Gaussian $\mathcal{N}(0, 1)$, and S_i, independent of C_i and D_i, is a non-negative random variable that models the non-constant deviation, such that C_iS_i and D_iS_i, that model respectively the real and imaginary marginals of a Gabor coefficient, are Generalized Gaussians; as S_i does not affect the phase, C_iS_i and D_iS_i preserve the circular symmetry.

This model covers all the observed properties of Gabor coefficients (circular symmetry between real and imaginary parts, and GG marginals), and allows us to calculate an accurate distribution for their magnitudes, that we present now.

Firstly, we calculate the distribution of the multiplicative factor S_i, that is determined by the Gaussian transform [29] of a Generalized Gaussian variable (GTGG):

$$f_{S_i}(s^2) = \frac{1}{s^2}\sqrt{\frac{\pi}{2s^2}}\left(\mathcal{F}^{-1}\left(f_{G_i}(\sqrt{j\omega})\right)\right)_{t=\frac{1}{2\sigma^2}},$$

where $\mathcal{F}^{-1}$ represents the inverse Fourier Transform.

Then, the modulus of G_i will be given by

$$|G_i| = \sqrt{C_i^2 + D_i^2}\cdot S_i,$$

being $R_i = \sqrt{C_i^2 + D_i^2}$ Rayleigh distributed. Finally, the density of the magnitude of a Gabor coefficient represented as the product of a Rayleigh and an independent GTGG variable can be calculated as:

$$\begin{aligned} f_{|G_i|}(x) &= \int_0^\infty f_{S_i}(\sigma^2)\frac{x}{\sigma^2}e^{-\frac{x^2}{2\sigma^2}}d\sigma^2 \\ &= \int_0^\infty \frac{1}{\sqrt{2\pi\sigma^2}}\left(\int_{-\infty}^{\infty}\frac{\beta_i\cdot c_i}{2\Gamma(\frac{1}{c_i})}e^{-\beta_i^{c_i}(j\omega)^{c_i/2}+j\frac{\omega}{2\sigma^2}}d\omega\right)\frac{x}{\sigma^2}e^{-\frac{x^2}{2\sigma^2}}d\sigma. \end{aligned}$$

Reversing the order of the integrals and after some algebra, we get

$$f_{|G_i|}(x) = \frac{c_i\beta_i}{2\cdot\Gamma(\frac{1}{c_i})\cdot x}\times\int_0^\infty\left[\frac{\cos(\frac{3}{2}\tan^{-1}(\frac{\omega}{x^2}) - \beta^{c_i}\omega^{c_i/2}\sin(\frac{\pi c_i}{4}))}{(x^4+\omega^2)^{\frac{3}{4}}}e^{-\beta^{c_i}\cos(\frac{\pi c_i}{4})\omega^{c_i/2}}\right]d\omega. \tag{6.1}$$

This integral, can be numerically evaluated for a given pair (c_i, β_i), obtaining the results exemplified in Figure 6.1 for a few shape factors. The resulting pdf is more peaky and heavy-tailed than the Rayleigh, that is also shown in Figure 6.1 for comparison.

6.3.1. Parameter Estimation and Goodness of Fit

In order to test the validity of the model and the degree of representativeness of the actual data distribution, we calculated the parameters of our model for

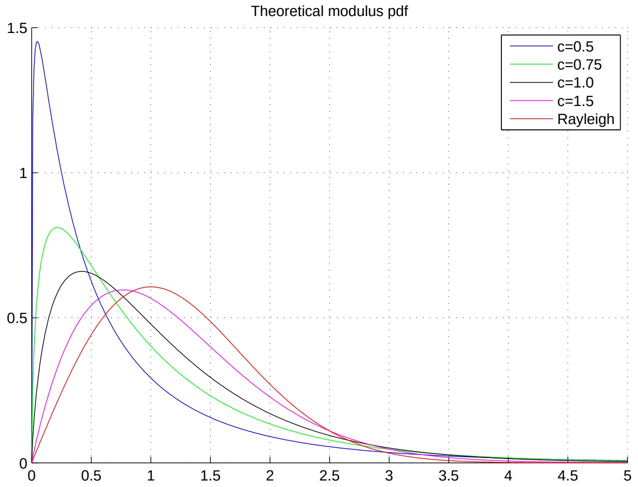
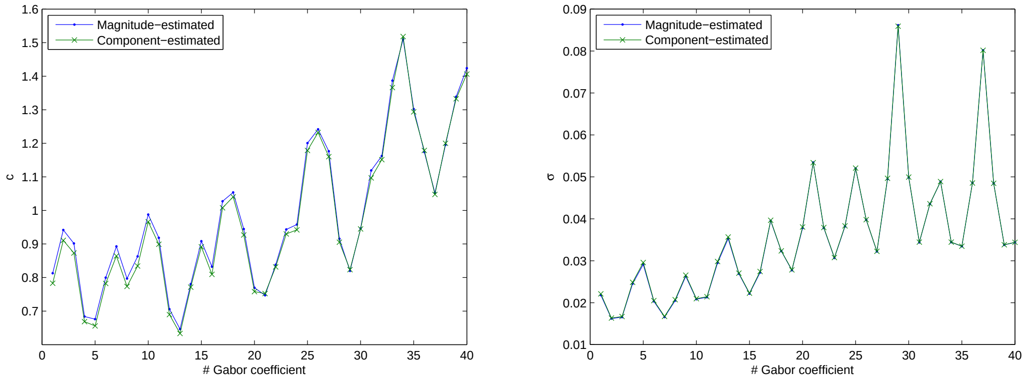


Figure 6.1: Pdfs for the presented model with varying shape factor c compared to a Rayleigh distribution

the magnitude of Gabor coefficients (extracted as stated in [114]) on two known biometric databases: XM2VTS [160] and LFW [125]. For this task, we employed ML estimation, using the numerical calculation of the pdf (6.1). Figure 6.2 shows the calculated parameters for each coefficient, that are in perfect agreement with the results obtained for the GG marginals of the real and imaginary part in [114]. This fact validates the hypothesized dependence between both real and imaginary part, and corroborates that our model perfectly agrees with the widely adopted assumption of Generalized Gaussian real and imaginary parts for Gabor coefficients.



(a) (b)

Figure 6.2: ML estimated shape factor c (a) and deviation σ (b) for the proposed magnitude model and for the GG real and imaginary components in the XM2VTS database.

For evaluating the goodness of fit, we use the Kullback-Leibler divergence [67]. This figure provides a measure of the statistical distance between two discrete distributions with probability functions P and Q, and is given by

$$KLD(P,Q) = \sum_{i=0}^{K-1} P(i) \log \left(\frac{P(i)}{Q(i)} \right),$$

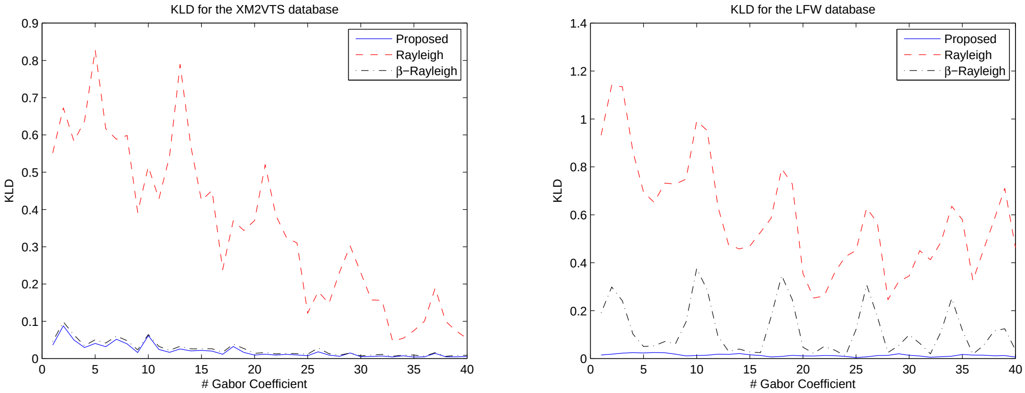
where K stands for the number of possible values of the discrete distribution. In order to apply the KLD to our case, in which we have a continuous variable but a finite sample, we discretize the theoretical pdf in K equally spaced intervals and compare it to the empirical discrete pdf given by the histogram of the actual data.

Figure 6.3 shows the KLD calculated for the XM2VTS and the LFW databases for our model compared to two distributions: Rayleigh, equivalent to considering Gaussian i.i.d. real and imaginary part of the Gabor coefficients, and β-Rayleigh, introduced in [115] as a closed-form generalization of the Rayleigh. For both databases, our model gives a much better fit than the Rayleigh, especially for the high frequency coefficients, which present a lower shape factor, and thus, cannot be modeled as Gaussian. It is also present in Figure 6.3 the pseudoperiodic effect of the shape factor when varying the orientation, that was pointed out in [114]. This effect produces the ripple in the calculated KLD with respect to the Rayleigh function, with minima in the coefficients which have shape factors closest to $c = 2$. Nevertheless, as shape factors are always in the range $(0.5, 1.5)$, the proposed model will always yield a better fit than the Rayleigh model.

On the other hand, while for the XM2VTS database our model slightly improves on the fit given by the β-Rayleigh, the improvement is much noticeable for the LFW database. These results stem from the fact that XM2VTS's samples are taken within controlled conditions, and thus present limited variation of pose and illumination, while LFW yields a richer variety of poses and illumination, producing a heavier-tailed distribution for the magnitude of the coefficients that is harder to approximate with a β-Rayleigh, but that our model fits well because the original assumptions on which our model is grounded are fulfilled by both databases. In fact, the main difference between the magnitude distribution that originates from our model and the β-Rayleigh is found in the tails: our model results in heavier tails for an equivalent shape factor.

6.3.2. Optimal Quantization of Biometric Data

The presented model has interest by itself, and there are many applications that can benefit from its use. In this chapter, the target of the model is the minimization of the plaintext cardinality of the involved magnitudes so the encrypted private system can effectively handle their processing without the need



(a) (b)

Figure 6.3: KLD results for each of the forty Gabor coefficients magnitudes modeled as Eq. (6.1), Rayleigh and β-Rayleigh [115] for the XM2VTS (a) and the LFW (b) databases.

of interaction. Hence, we reduce storage and plaintext cardinality via optimal coefficient quantization using a Lloyd-Max quantizer [150, 154], that achieves minimum mean squared error (MSE) for a given number N_L of representative levels, given an accurate distribution of the to-be-quantized variables.

A Lloyd-Max strategy was also used in [114] for quantizing independently the real and imaginary parts of Gabor coefficients. However, since most Gabor-based face recognition systems discard phase information for matching, it is more appropriate to quantize the magnitudes instead of the original complex coefficients; another desired effect stemming from this choice is that it gets a more significant storage reduction. Consequently, we expect to achieve similar performance with less representative levels.

Additionally, the quantization in [114] and [215] uses a number N_L of centroids for each coefficient, and preserves the real values of the corresponding levels as the output quantizations. This strategy allows for a storage reduction in a cleartext system, as only the (integer) indices of the corresponding quantization levels are stored, together with a mapping from the indices to the real levels; however, this mapping has to be applied to recover the corresponding quantizations before operating on them. Hence, an encrypted system that has to work with integer-valued numbers cannot translate this quantization into an actual reduction in plaintext size.

Instead, we propose the use of integer quantization indices, as a more suitable strategy for the encrypted system: i.e., all the involved variables are mapped to integer numbers with a very low cardinality (the number of quantization levels); additionally, the use of indices involves a nonlinear scaling of all the coefficients in such a way that, after scaling, the resulting centroids are arranged in equidistant

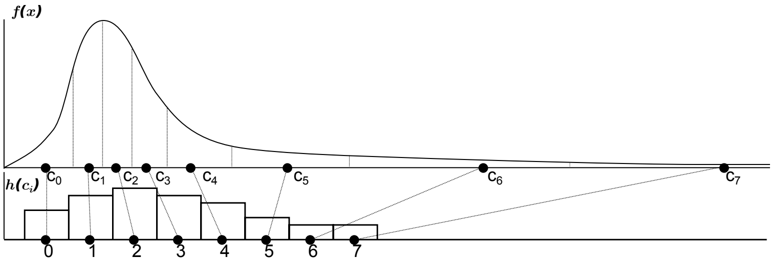


Figure 6.4: Qualitative diagram showing the inherent nonlinear scaling produced by the use of integer quantization indices (lower graph) instead of the actual real values of the quantization centroids for Lloyd-Max quantization (upper graph).

bins, as shown in Figure 6.4. This also produces an inherent normalization, reducing coefficients with high variance and amplifying those with low variance, and fixing the range for all the coefficient indices.

Section 6.5.1 shows experimental results on both XM2VTS [160] and LFW [125] databases that demonstrate that our strategy provides great storage reduction without degrading system performance, what makes such strategy an optimum choice for reducing the plaintext size in an encrypted implementation of the recognition system. Also, the nonlinear scaling will positively affect the performance of the system in some of the studied cases.

Now that we have presented our coefficient model and the optimal quantization strategy that allows for a great plaintext cardinality reduction, we introduce in the next section our extension to the homomorphic cryptosystem that, when combined with the described quantization strategy, yields the fully noninteractive face verification system presented in Section 6.5.

6.4. Extending Gentry's Fully Homomorphic Cryptosystem

We take one of the latest versions of Gentry's bootstrappable fully homomorphic cryptosystem, presented in [105]. The cryptosystem is GGH-type based on ideal lattices. Given a given lattice L with shortest nonzero vector length $\lambda_1(L)$, the rationale behind GGH cryptosystems lies in choosing two bases with different correction radii[1]:

[1] The correction radius is a property of a lattice basis; it can be defined as the norm of the shortest error vector that, added to a lattice point, cannot be corrected using that basis (as it falls outside the parallelepiped–Voronoi region–defined by the reduction modulo the basis). The correction radius of a basis is upper bounded by the inner radius of the lattice, defined as half the norm of the shortest non-zero vector of the lattice (shortest distance between two lattice points, λ_1). *Good bases* yield almost spherical Voronoi regions, with a correction radius approaching the inner radius of the lattice.

- $\boldsymbol{B}_{sk}$ constitutes the secret key; it is a *good basis*, in the sense that it allows to efficiently solve certain instances of the closest vector problem in the lattice, and it has a large enough correction radius; this basis contains short almost-orthogonal vectors.
- $\boldsymbol{B}_{pk}$ ($\boldsymbol{B}$ from now on) constitutes the public key; it is a *bad basis*, in the sense that solving the closest vector problem in L using $\boldsymbol{B}$ is algorithmically hard. $\boldsymbol{B}$ is usually chosen as the Hermite Normal Form (HNF) of the lattice, as it can be efficiently computed from any other basis, it has a very small correction radius (asymptotically zero as the dimension increases), and the LLL algorithm (the most widely known lattice reduction algorithm, by Lenstra, Lenstra and Lovasz [143]) run on the HNF is particularly slow [161].

The encryption $\boldsymbol{c}$ of a message m consists on the addition of an error vector $\boldsymbol{e}$ such that $||\boldsymbol{e}||_2 < \lambda_1(L)$, that encodes m, to a point in the lattice. For decrypting, the error vector $\boldsymbol{e}$ is recovered using the basis $\boldsymbol{B}_{sk}$ as $\boldsymbol{e}' = \boldsymbol{c} \mod \boldsymbol{B}_{sk}$.

The *somewhat homomorphic* scheme presented by Gentry in [105], following the same approach as Smart and Vercauteren [208], uses a principal-ideal lattice J, generated by a chosen polynomial $v(x)$ with t-bit signed random integer coefficients ($\boldsymbol{v}$ in its vector notation), in the ring of polynomials modulo $f_n(x) \doteq x^n + 1$; unlike in [208], where $J = (\boldsymbol{v})$ is required to have prime determinant, [105] just requires a specific structure for the HNF:

$$\boldsymbol{B}^T = HNF(J) = \begin{pmatrix} d & 0\,0 & 0 \\ -r & 1\,0 & 0 \\ -[r^2]_d & 0\,1 & 0 \\ & & \ddots & \\ -[r^{n-1}]_d & 0\,0 & 1 \end{pmatrix},$$

where d can be defined as $d = \det(J)$ or, equivalently, as the resultant of the polynomials $v(x)$ and $f_n(x)$, and r is a root of $f_n(x) \mod d$, that forms the vector $\boldsymbol{r} = [-r, -[r^2]_d, \ldots, -[r^{n-1}]_d]^T$. $\boldsymbol{B}$ is the *public-key* encryption matrix, completely determined by the pair of integers (d, r), while the private key is given by $v(x)$ and its scaled (modulo $f_n(x)$)-inverse $w(x)$ (i.e., $v(x) \times w(x) = d \mod f_n(x)$), of which only one of the coefficients of $\boldsymbol{w}$, denoted w_i, is required for the decryption procedure.

As defined, this cryptosystem is *quasi*-homomorphic under addition and multiplication, that are directly mapped from the crypto-text ring (errors w.r.t. lattice points) to the clear-text ring. There is, however, a restriction to this homomorphism, as both operations are only correctly mapped when the error lies within the same Voronoi region of the lattice L after applying the operation.

For reaching a full homomorphism, Gentry proposes to squash the decryption circuit so that it can be executed also homomorphically; this *squashing* consists in adding to the public key a big set of random elements for which a sparse subset sums up to $w_i \mod d$, and reducing the secret key to the (sparse) characteristic vector of this subset. This produces also an effective reduction in the degree of the decryption polynomial (as a function of the secret key bits), at the cost of supporting the security of the cryptosystem on the additional assumption that it is hard to determine which is the sparse subset. Therefore, the cryptosystem becomes *bootstrappable*, in the sense that it is able to homomorphically execute the decryption circuit under encryption, providing a fresh encryption from a degraded one, and this is crucial for achieving a full homomorphism. We propose to trade this full homomorphic capacity for the ability to execute polynomials of the same order as the squashed decryption circuit before the cipher gets corrupted enough to lose data, using the cryptosystem as a quasi-fully homomorphic scheme, and incrementing the allowed cardinality of the plaintext as shown in the next section.

6.4.1. Proposed Extension to Gentry's Cryptosystem

One of the limitations of Gentry's cryptosystem is that it can only deal with binary numbers in $(\mathbb{Z}_2, +, \cdot)$, being the homomorphic ring operations *and* and *xor* gates; this means that a simple arithmetic circuit with b-bit numbers needs a high amount of binary homomorphic operations that increase the noise within the Voronoi region of the lattice, whose volume determines the maximum number of operations that do not lead to a decoding error. The maximum depth of an executable polynomial has been calculated empirically by Gentry and Halevi [105], and used for bootstrapping the decryption circuit and achieving a full homomorphism.

In this section we extend the plaintext-size, allowing for homomorphic additions and multiplications in $(\mathbb{Z}_{2^k}, +, \cdot)$ (powers of two are chosen for convenience); we also give a theoretical lower bound on the maximum number of executable multiplications, that also supports Gentry's empirical study for $\mathbb{Z}_2$. The extension seeks to enhance the efficiency of arithmetic non-interactive operations and decrease the cipher expansion rate and to trade the full homomorphic property by the possibility of dealing with a limited but high number of sequential arithmetic processing without interaction. Furthermore, the key-generation process does not need to be changed, so the same keys can be used for the binary cryptosystem and for the proposed extension.

6.4.1.1. Encryption

In Gentry's original cryptosystem, the encryption operation of a bit $b \in \mathbb{Z}_2$ uses a random noise vector $\boldsymbol{u} \in \{0, \pm 1\}^n$, with each entry chosen as 0 with probability q and ± 1 with probability $(1-q)/2$ each; we extend the encryption for coping with $m \in \mathbb{Z}_{2^k}$

$$\boldsymbol{a} = 2^k \boldsymbol{u} + m \cdot \boldsymbol{e}_1; \quad \boldsymbol{c} = \boldsymbol{a} \mod \boldsymbol{B} = [a(r)]_d \cdot \boldsymbol{e}_1,$$

where $\boldsymbol{e}_1$ is the first vector of the canonical basis. The vector $\boldsymbol{c}$, as in the original construction, has only one non-zero component, representative of the encryption:

$$c = [a(r)]_d = [m + 2^k \sum_{i=0}^{n-1} u_i r^i]_d.$$

The complexity of encrypting a k-bit number is the same as for encrypting a bit in the original system. Furthermore, the security in terms of Birthday-type attacks is not altered either, as the noise vector has the same bits of entropy; hence, given a security level λ, q may still be chosen such that

$$2^{(1-q)n} \cdot \binom{n}{qn} > 2^{2\lambda}.$$

A discussion about the security of the extended cryptosystem can be found in Section 6.6.

6.4.1.2. Decryption

For the decryption, the original scheme uses an optimized procedure that only needs one of the odd coefficients of $\boldsymbol{w} \mod d$, denoted w_i. Hence, the decryption for a k-bit message m becomes

$$m = [c \cdot w_i]_d w_i^{-1} \mod 2^k.$$

The only difference w.r.t. the original decryption is the product by $w_i^{-1} \mod 2^k$; being w_i odd, it always exists: the choice of powers of two for the extended plaintext allows for keeping the same key generation process, while the added decryption complexity is negligible compared to modulo d operations.

6.4.2. Homomorphically Achievable Polynomial Degree

Incorrect decryption may only happen when the error vector added to a lattice point lies outside the Voronoi region of the used lattice. This condition boils down to

$$||\boldsymbol{a}^T \boldsymbol{W}||_\infty < d/2, \tag{6.2}$$

where $\boldsymbol{W}$ is the rotation basis that generates $(w(x))$, having in each row the coefficients of $w(x) \cdot x^i \mod f_n(x)$. Due to the structure of $\boldsymbol{W}$ (a circulant matrix with negated lower triangular part), we can bound

$$||\boldsymbol{a}^T\boldsymbol{W}||_\infty \leq ||\boldsymbol{a}||_\infty ||\boldsymbol{W}||_\infty = \max_i(|a_i|) \cdot \sum_{i=0}^{n-1}|w_i| \leq \sum_{i=0}^{n-1}|w_i|\sum_{i=0}^{n-1}|a_i|,$$

$$\sum_{i=0}^{n-1}|w_i|\sum_{i=0}^{n-1}|a_i| < d/2 \Rightarrow ||\boldsymbol{a}^T\boldsymbol{W}||_\infty < d/2.$$

The number of non-zero elements (Nz_j) of a chosen $\boldsymbol{u}_j$ follows a Binomial distribution $Nz_j \sim Bi(n, 1-q)$. In a fresh encryption, each of these elements has modulus 2^k, while the message has a modulus $|m| < 2^k$. Hence, $\sum_{i=0}^{n-1}(|a_i|) < 2^k(1 + Nz_j)$.

On the other hand, after a multiplication between two ciphertexts $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ (in the polynomial quotient ring $\mathbb{Z}_d[x]/(f_n(x))$), the resulting point must also be within the Voronoi region. The product of two polynomials modulo $f_n(x)$ is equivalent to a cyclic convolution of their coefficient vectors (with a sign change for the overlapped subvector). Furthermore, as fresh encryptions have the same absolute value (2^k) for all the non-zero coefficients of $\boldsymbol{u}$, the L^1-norm of the resulting coefficient vector of the product of a given ciphertext $\boldsymbol{c}_1$ and a fresh encyrption $\boldsymbol{c}_2$ is upper-bounded by $||\boldsymbol{c}_1||_1 \cdot 2^k(1 + Nz_2)$. In general, we have that, after n_m successive products of a cipher by fresh encryptions,

$$||\boldsymbol{a}_{n_m}^T\boldsymbol{W}||_\infty \leq \left(\sum_{l=0}^{n-1}|w_l|\right)\prod_{i=0}^{n_m}2^k(1 + Nz_i).$$

Hence, we can bound the probability of decryption error

$$P[\text{dec error}] = P[||\boldsymbol{a}^T\boldsymbol{W}||_\infty \geq d/2] \leq$$
$$P\left[\underbrace{\sum_{i=0}^{n_m}\log(1 + Nz_i)}_{N_{n_m}} \geq \log\left(\frac{d}{2^{k(n_m+1)+1}\sum_{l=0}^{n-1}|w_l|}\right)\right],$$

where N_{n_m} is a random variable with bounded support ($N_{n_m} \in [0, (n_m+1)\log(n+1)]$); thus, it may happen that for a low number of dimensions and few multiplications the probability of decryption error be zero. Nevertheless, due to q being chosen such that $(1-q) << 1$, for high enough n (like the commonly used n even for short-term security) the error probability will not get to be identically zero in any case, and the pdf of N_{n_m} will present a narrower bell as n or n_m increase, so by virtue of the Central Limit Theorem (CLT), N_{n_m} can be accurately approximated by a Gaussian variable with parameters

$$\mu_{n_m} = (n_m + 1) \cdot \mu \doteq (n_m + 1) \cdot \sum_{i=0}^{n} \log_2(1+i) \binom{n}{i} (1-q)^i q^{n-i},$$
$$\sigma^2_{n_m} = (n_m + 1) \cdot \sigma^2 \doteq (n_m + 1) \cdot \sum_{i=0}^{n} (\log_2(1+i) - \mu)^2 \binom{n}{i} (1-q)^i q^{n-i},$$

that will provide a very accurate approximation near the bell and an overestimation of the decoding error probability in the tails, due to the bounded support of N_{n_m}.

We may bound the maximum number of bits to which we can extend the ciphertext for allowing a given number n_m of successive multiplications with a given probability of error p_e using the Q function[2]:

$$k_{max} = \left\lfloor \frac{\log_2(d/||\boldsymbol{w}||_1) - 1}{n_m + 1} - \mu - \frac{Q^{-1}(p_e)\sigma}{\sqrt{n_m + 1}} \right\rfloor. \tag{6.3}$$

As expected, the maximum number of bits decreases with increasing n_m, and it is heavily influenced by the quotient $d/||\boldsymbol{w}||_1$, that intuitively indicates the effective radius of the Voronoi region, supporting noise addition. On the other hand, the choice of t (bit-size of the coefficients of the generating polynomial for the ideal lattice J) determines the maximum value of this quotient: as the polynomial product of $v(x) \times w(x) = d \mod f_n(x)$, in vector notation this means that, using the Hölder inequality:

$$d = \boldsymbol{v}^T \cdot [w_0, -w_{n-1}, \ldots, -w_1]^T \leq ||\boldsymbol{v}||_\infty ||\boldsymbol{w}||_1 < 2^t ||\boldsymbol{w}||_1 \Rightarrow \frac{d}{||\boldsymbol{w}||_1} < 2^t.$$

Hence, for a good lattice, the maximum correctable noise norm (decryption radius) will be close to t bits (cf. Figure 6.5b), and we can provide an estimation of the maximum plaintext bit-size for correct decryption after a given number of multiplications for a generic good lattice, just substituting $\log_2(d/||\boldsymbol{w}||_1)$ by t in Eq. (6.3). Reciprocally, the inverse of this expression yields the maximum number of affordable multiplications with a bounded decryption error. It must be noted that n_s consecutive homomorphic additions can increase at most in $\log_2(n_s)$ bits the size of the ∞-norm of the noise vector (Eq. (6.3) can take this into account by subtracting $\log_2(n_s)$ from t). Hence, when determining the maximum degree of a polynomial run on fresh ciphered variables, the maximum number of multiplications is the determining factor. Gentry and Halevi provide an approximation of

[2] The Q function can be defined as

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-u^2/2} du$$

Table 6.1: Lower bound on the maximum number of products and Gentry's empirically obtained maximum degree polynomial as a function of t, with $n = 128$

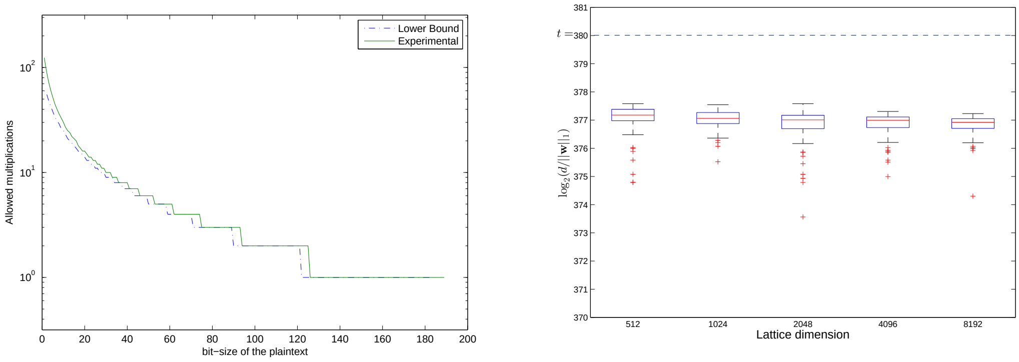
t	64	128	256	384
Lower bound	10	22	46	69
Empirical [105]	13	33	76	128

the maximum degree deg of an elementary symmetric polynomial evaluated on m encrypted binary variables, bounding the decryption radius by the approximated Euclidean norm of the polynomial output: $2^t \geq c^{deg}\sqrt{\binom{m}{deg}}$; the results deviate from this expression for large m due to the overestimation of the effect of additions, as the combinatorial number of summed monomials grows above the dimensionality of the lattice, and they cannot be considered independent anymore. Table 6.1 shows the validity of our bound compared to the experimental results obtained by Gentry.

Fig. 6.5a represents the number of sequentially performed products with new fresh ciphers before a decryption error occurs (for $n = 512$, $t = 380$ and $q = 1 - 20/512$, picking the minimum of 1000 trials), compared to the given lower bound for $p_e = 10^{-4}$. The bound is fairly conservative for small plaintexts that allow for a high amount of products, as it is a worst-case bound, but it becomes tight for medium-to-high k, even when the Gaussian approximation in those cases provides an overestimation of the decryption error. We have also obtained very similar results with bigger lattices (as Gentry and Halevi did for the binary case), due to the quotient $\log_2(d/||\boldsymbol{w}||_1)$ being virtually constant for all the found lattices (Figure 6.5b shows this quotient for different random lattices of several dimensions with fixed $t = 380$), and the binomial distribution barely changing with high n when fixing the rate $(1 - q) \cdot n$.

6.5. Fully Private NonInteractive Face Verification

The combination of the model of Section 6.3 for optimal quantization of Gabor coefficients magnitude together with the extension of Gentry's cryptosystem presented in Section 6.4 provides an efficient and accurate solution to the problem of fully private face verification, adjusting and limiting the cardinality of the plaintext through efficient coefficient quantization with little impact on recognition performance (cf. Section 6.5.1) and hence making possible the use of the homomorphic cryptosystem for the recognition operation without any intermediate decryption, i.e., in a fully noninteractive way.



(a) (b)

Figure 6.5: (a) Minimum number of multiplications (Eq. (6.3)) without decoding error after 1000 trials as a function of k and (b) quotient $\log_2(d/||\boldsymbol{w}||_1)$ for random lattices of several dimensions

This application also showcases the benefits and versatility of the extended cryptosystem in a typical scenario of outsourced face verification with privacy constraints. In this scenario, a query user presents his face features and a tentative ID against a database; the system must determine if those features actually correspond to the previously enrolled ID. The target of the outsourced privacy-preserving system is to conceal both the presented face features and the database templates to the party that runs the verification process, while the database templates are also not disclosed to the query user.

Other privacy-preserving systems presented in the literature, like [89] or [196], are based on Eigenfaces [231]. In the clear, Gabor filters provide a slightly more complex solution with a better performance (about 8% increase, cf. Section 6.5.1) in known databases like LFW [125], thanks to the biological models that support the use of Gabor filters. Unlike previous works [215], for the private system we work with the integer indices of quantized coefficients instead of the actual quantized values; this allows for a hugely reduced plaintext size without much degradation in system performance (cf. Section 6.5.1), and benefits from an inherent normalization of the Jets, as the Lloyd-Max quantization already performs a nonlinear normalization (cf. Section 6.3.2). The verification algorithm is based on either average correlation (cosine distance) or average Euclidean distance; both can be efficiently calculated in the encrypted domain, and there are no statistically significant differences in recognition performance between both distances.

In the enrollment phase, the presented feature vectors are encrypted and stored in a central database for later use as templates. The verification threshold η is a system parameter also kept encrypted. We employ a linear-kernel SVM [73], previously trained on local distances, that produces a weight vector $\boldsymbol{\alpha}$, resulting from the linear combination of the support vectors $\{\boldsymbol{s}_j\}_{j=1}^{M-1}$

$$\text{score}_{SVM}(\boldsymbol{x}) = \sum_{j=0}^{M-1} \beta_j \boldsymbol{s}_j^T \cdot \boldsymbol{x} - \eta = \boldsymbol{x}^T \underbrace{\sum_{j=0}^{M-1} \beta_j \boldsymbol{s}_j}_{\boldsymbol{\alpha}} - \eta; \tag{6.4}$$

the score is classified as *true* if it is non-negative, and as *false* otherwise. For each pair of compared feature vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, if the input to the SVM is chosen as $x_j = (a_j - b_j)^2$, the effect of the weight vector α is to produce a weighted Euclidean distance $\text{dist}(\boldsymbol{a}, \boldsymbol{b}) = \sum_{i=0}^{N-1} \alpha_i \cdot (a_i - b_i)^2$ as the recognition metric. In the verification phase, a user presents an ID to be matched together with the encrypted quantization indices $\boldsymbol{g}$ of the Gabor coefficient vector from his face. The database holder homomorphically calculates the encryption of the *soft* score

$$\text{soft\_score}(\boldsymbol{g}, id_i) = \sum_{i=0}^{N_{templates}-1} \text{dist}(template_i, \boldsymbol{g}) - N_{templates}\eta,$$

that is provided as the output of the verification process. Using the linear SVM in this way adds little computation complexity to the non-weighted original approach (the number of performed products is doubled), while producing a considerable enhancement of the recognition accuracy (cf. Section 6.5.1).

As a last remark, a hard score may be required for some applications. We will not consider that case explicitly in this chapter, as we aim at testing the raw performance of the extended cryptosystem in the envisioned biometric scenario in a fully noninteractive system. Nevertheless, the private implementation of the last comparison step needed for providing a hard score ($[\text{soft\_score}(\boldsymbol{g}, id_i) \geq 0]$) could be easily produced with one of the many interactive comparison protocols available for an additive homomorphic cryptosystem, like the one used in [89, Section 5]; the realization of this protocols starting from the extended-Gentry encryption of the soft scores must take into account that for performing a statistically blinding decryption, necessary for the intermediate steps of the protocol, the cipher must support the encryption of numbers with a length κ bits higher than the normal coefficients and results; this means that for normal values of the security parameter κ ($\kappa \approx 70$ bits) and normal working magnitudes (around 20 bits for this application and thanks to the used efficient quantization), the extended cryptosystem will need to cope with ~ 90 bits clear-text sizes, and hence, it will be able to support at least two correct consecutive homomorphic products (Eq. (6.3)), being this enough for calculating the needed weighted Euclidean distance.

6.5.1. Recognition Performance Results

In order to evaluate the impact of data quantization on system performance, we conducted experiments on the XM2VTS [160], and the LFW databases [125], whose results are shown in the following subsections. Taking into account that this chapter seeks the secure evaluation of the scores in a privacy-preserving encrypted system, we have used a baseline recognition method (similar to the ones in [114, 215], that employ a simple fusion of linear classifiers); we are not aiming at improving the recognition rate of state-of-the-art classifiers, but showing instead that the presented accurate feature model combined with optimal quantization does not hinder the recognition performance of the system. Thus we choose a baseline system to better show the actual effects of quantization. For that purpose, we firstly use just the Euclidean distance between the pre-normalized magnitudes of the jets of compared faces, without any additional weighting on the quantized coefficients; we also provide the results of using a simple but effective linear SVM trained on the evaluation sets, that effectively provides a weight vector $\boldsymbol{\alpha}$ for calculating a weighted Euclidean distance (Eq. (6.4)) and obtain improved results with a very little complexity overhead, also in a suitable configuration for the privacy-preserving implementation shown afterwards.

6.5.1.1. XM2VTS database

Experiments were performed on XM2VTS following configuration I of the Lausanne protocol. The XM2VTS is divided into three sets: training, evaluation, and test. The training set was used to build client templates, estimate model parameters (c and σ), and calculate the representative values (centroids) for the following set of quantization levels $N_L = \{2, 4, 8\}$. The evaluation set was used to estimate thresholds that discriminate between client and impostor attempts, and train the linear SVM classifier for providing the appropriate weight vector. These thresholds are chosen so that the False Acceptance Rate (FAR) equals the False Rejection Rate (FRR) on the evaluation set. Finally, using the obtained thresholds, FAR and FRR are measured on the separate test set. Table 6.2 presents, for the set of quantization levels $N_L = \{2, 4, 8\}$, the comparison between **a)** Quantizing the complex coefficients (real and imaginary parts separately), as proposed in [114], **b)** Quantizing the magnitudes of coefficients, as proposed in [215], and **c)** the usage of integer quantization indices instead of actual quantized values for our model (for [114], only quantized values are used); the three approaches are compared with and without the use of the linear SVM; when using quantization levels, as well as for the original unquantized system, a prenormalization is undertaken so that each 40-coefficient jet for each localization has unit norm[3].

Table 6.2 shows the Total Error Rates (TER=FAR+FRR) on the test set for the compared approaches, as well as for the unquantized system. We can draw the following conclusions:

[3] It should be noted that, for fair comparison, the same feature extraction parameters of [114] were used here.

Table 6.2: TER (%) on the test set of the XM2VTS for quantized data.

N_L		2	4	8	Unquant.
No SVM	[114]	28.8	27.54	26.65	12.33
	[215] (levels)	16.37	12.13	12.21	
	Proposed (indices)	27.04	21.92	20.52	
SVM	[114]	15.11	11.22	10.53	7.68
	[215] (levels)	7.41	6.07	7.50	
	Proposed (indices)	6.73	7.62	7.62	

1. Our model significantly outperforms [114] for $N_L = \{2, 4, 8\}$ in all the configurations; [114] models and quantizes independently the real and imaginary parts of the Gabor coefficients, resulting in not so accurate results. Our model correctly captures the dependency between marginals and produces a much better fit, preserving much more useful information for recognition in less bits.

2. Using quantization levels and without the help of SVM classifiers (as in [215]), the system achieves original performance with only $N_L = 4$ levels, i.e. only 2 bits are needed per coefficient, partly thanks to the prenormalization of the Gabor jets.

3. Using the proposed system with quantization indices, the performance with no weighting is decreased with respect to the original system, but it is far better than using pre-normalized quantization levels as done in [114], with a largely reduced storage.

4. With the linear SVM, the prenormalization has no effect, but all the results are greatly improved. The use of either levels [215] or quantization indices does not significantly affect the performance, achieving a very good accuracy with indices and just 2 quantization bins (one bit per coefficient), even better than the unquantized system, due to the elimination of some non-informative noise during the quantization process, and benefiting from the non-linear scaling.

Therefore, we can conclude that even $N_L = 2$ indices per coefficient are enough for achieving the original performance with a linear-kernel SVM. Compared to [114], where more than $N_L = 8$ levels are needed (and two quantizations–real and imaginary parts–must be performed per coefficient), we obtain a storage reduction of $\frac{\log_2(2)}{\log_2(8)+\log_2(8)} = \frac{1}{6}$, for an even better performance level.

Additionally, a simple linear-kernel SVM lets us obtain a considerable performance boost with only linear operations that, as we show in the following sections, can also be executed noninteractively in the presented encrypted system. Furthermore, quantization allows for an unaltered recognition performance just using two indices. This means that a Lloyd-Max binary quantization of the coefficients does not hinder the recognition ability and preserves the identification information present in those coefficients.

Finally, it is worth noting that the weighting coefficients must also be quantized so that they can be used in the encrypted private system. We have checked that these coefficients $\{\alpha_i\}_{i=0}^{N-1}$ obtained from the training phase of the SVM approximately follow a Gaussian distribution with zero mean (these coefficients come from the sum of the signed–almost independent–coefficients of the support vectors, hence converging to a Gaussian due to the CLT); applying a Lloyd-Max quantizer based on this Gaussian, we found that using two levels (i.e., preserving just the sign of each α_i) has a negligible impact on the recognition performance.

6.5.1.2. Labeled Faces in the Wild (LFW) database

In order to show that the proposed quantization scheme works also with more challenging imagery, we conducted experiments on the LFW database [125]. This recently collected dataset contains 13,233 face images which have several compound problems (imperfect localizations, in-plane rotations, non-frontal poses, low resolution, non-frontal illumination, varying expressions...). The images were obtained by running an automatic face detector on images collected from the Internet, followed by face centering, scaling and cropping. In our experiments we extracted closely cropped faces using a fixed bounding box placed in the same location for each LFW image, resulting in faces of 120 × 100 pixels, from where Gabor features were extracted.

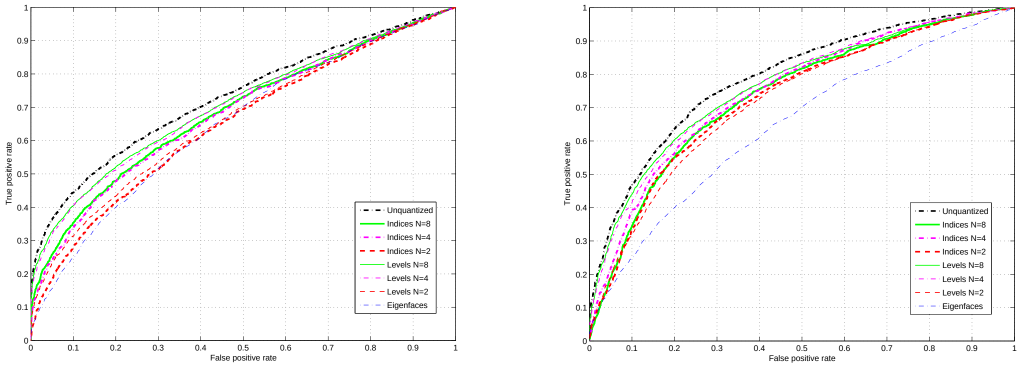
Images from *view 1* of the LFW database were used to estimate model parameters and the representative values (centroids) for the same set of quantization levels $N_L = \{2, 4, 8\}$ as in the previous subsection. The experiments on this dataset were carried out following the *image restricted* paradigm, and performance was reported on *view 2* using the 10 fold, leave-one-out cross-validation scheme described in [125]. Figure 6.6 presents the ROC curves for the set of quantization levels $N_L = \{2, 4, 8\}$ and quantization indices for those levels, along with performance using non-quantized coefficients. The classification accuracy μ averaged over the 10 folds (%) is presented in Table 6.3. The recognition accuracy is not reduced in a great amount when quantizing to levels and applying the prenormalization (cf. Section 6.5.1.1); furthermore, original performance is recovered for $N_L = 4$ without SVMs and with $N_L = 8$ with SVMs. For this database, the use of quantization indices instead of levels has an impact on performance, and unquantized accuracy is not totally recovered; this can point to the fact that the non-linear normalization inherently performed when extracting the quantization indices produces a loss of information, leaving all the consecutive quantized

Table 6.3: Average classification accuracy μ (%) on *view 2* of the LFW database using both original and compressed data. Eigenfaces is shown for comparison.

N_L		2	4	8	Unquant.	Eigenfaces
No SVM	Levels	61.93	65.93	66.00	65.97	60.02
	Indices	60.67	62.60	62.87		
SVM	Levels	65.9	70.67	71.90	72.63	
	Indices	67.5	68.73	68.23		

values equidistant, and affecting the calculation of the actual distance between two feature vectors. Nevertheless, the performance of the system without SVMs can be assimilated to baseline V1-like models [184], while for the system with SVM weighting, the results are at the level of other Gabor-based schemes (≈68% for V1-like+ models). As shown in Figure 6.6 for comparison, the performance for Eigenfaces [231] drops down to 60% accuracy.

As happens with the XM2VTS, the weighting coefficients that the SVM produces after training are approximately Gaussian, and quantizing them to two or four levels produces a negligible impact on recognition accuracy. This system, in which all the involved values are integers with a very low cardinality is the one that we use in this chapter as the basis for our non-interactive privacy-preserving face recognition protocol.



(a) (b)

Figure 6.6: ROC curves averaged over 10 folds of LFW's view 2 without SVMs (a) and with a linear-kernel SVMs (b)

6.5.2. Complexity Analysis

In order to test the efficiency of the system in a real scenario, we have implemented the extended encryption and applied it as the basis for privately executing the weighted Euclidean distance between quantized Gabor jets, in the LFW pair-matching setting.

We take the 8-indices quantization for its good compromise between clear-text cardinality and recognition performance for both XM2VTS and LFW. We have fixed the size of the used lattice to $n = 512$ dimensions, with $t = 380$ and $q = 1 - 20/n$, for a security parameter of $\lambda \approx 70$. We work with 5200-dimensional Gabor vectors for each face (13×10 localizations, 8 orientations and 5 scales) with 3-bit coefficients, so calculating the weighted Euclidean distance between two vectors needs two multiplications per pair of values, 5199 additions and one subtraction. Hence, starting from 8-level coefficients and 4 level weights, the resulting score is correctly represented using $\lceil \log_2(5200 \cdot 2 \cdot 8^2 \cdot 4) \rceil = 22$ bits (20 bits without weights), so we use $k = 22$ bits for the extended cryptosystem. Taking into account the $\log_2(5200) = 12.3$ bits of decrease for the effective decryption radius, Eq (6.3) yields 13 supported consecutive multiplications, so the extended cryptosystem can perfectly cope with the whole distance calculations, without incurring in decryption errors (but with negligible probability).

For implementation we have used the GMP [3] and NTL [6] libraries in C++, and tested the time efficiency without any kind of parallelization in one core of an Intel i5 at 3.30GHz with 8GB of RAM. Tables 6.4 and 6.5 show the efficiency figures for the proposed algorithm compared to the expected running times of a *traditional* implementation based on an additive homomorphism (Paillier-based [177], using a 2048-bit modulus), with either clear-text templates and weights (PaillierCT, partial privacy) and with encrypted templates and weights (PaillierE, total privacy using interactive multiplication protocols); in both Paillier-based systems the client provides the encryptions of both his/her face coefficients and their squared value, in the most favorable case for Paillier's homomorphism; we have also included, for reference, the estimated execution time of Gentry's original binary cryptosystem using binary circuits for addition and multiplication; this system cannot provide valid outputs without using homomorphic deciphering circuits, as the degree of the distance circuit exceeds the noise capacity of the used lattice; each of these circuits (for bootstrapping the cipher of a bit), that needs to be applied after each binary multiplication gate, runs in about 8 seconds in our test machine; this would highly increase the server's computational load in the binary version of Gentry, rendering it completely impractical (the whole circuit involves around $3.2 \cdot 10^5$ products); hence, we do not include them in the time evaluation, but they are an inherent limitation of the original binary cryptosystem that one must be aware of.

Thanks to our extension, the system becomes feasible both in terms of band-

Table 6.4: Efficiency figures for the privacy-preserving face-verification algorithm with no weighting: client times, server homomorphic processing (HP) time and needed communication

Execution times	Client		Server	Communication
	Cipher	Decrypt	(HP)	
Proposed	0.4s	0.0026s	12.3s	127MB
Gentry (binary)	1.3s	0.052s	968.1s	380MB
PaillierCT	15.4s	0.0043s	57.99s	5.3MB
PaillierE	22s	43.2s	479.0s	13.3MB

Table 6.5: Efficiency figures for the privacy-preserving face-verification algorithm with weighting: client times, server homomorphic processing (HP) time and needed communication

Execution times	Client		Server	Communication
	Cipher	Decrypt	(HP)	
Proposed	0.4s	0.0026s	25.7s	127MB
Gentry (binary)	1.3s	0.052s	1053s	380MB
PaillierCT	15.4s	0.0043s	87.01s	5.3MB
PaillierE	29.5s	86.5s	907.23s	21.3MB

width and server processing time overcoming the pointed out limitation; the use of homomorphic operations in $\mathbb{Z}_{2^k}$ instead of $\mathbb{Z}_2$ reduces the server computation time in almost two orders of magnitude (furthermore, as noticed, binary encryptions do not provide a correct output without the needed deciphering circuits), while the bandwidth is divided by a factor of three.

In terms of computational efficiency, the extended cryptosystem yields a clear advantage w.r.t. any of the others, even for Paillier with clear-text templates. The load for the client is decreased in two orders of magnitude w.r.t. Paillier, while the server's load decreases in a factor of almost 50. This is due to the lighter homomorphic operations compared to Gentry's, even when working with larger ciphertexts. Conversely, the transferred encryptions for the proposed system are less than one order of magnitude higher than for encrypted Paillier templates, due to the larger expansion factor that lattice cryptosystems like Gentry's present; this is the main fact that holds back the performance of the homomorphism; the presented extension advances in this path, reducing the expansion factor and greatly increasing the efficiency of the operations performed noninteractively at the server. Furthermore, when the scenario of interest is an outsourced system that processes private data, the initial bandwidth is not critical: the more operations can be performed *unattendedly*, the more versatile and powerful the system becomes.

6.6. Security Considerations

In this section we briefly discuss some security considerations about the proposed extension to Gentry's cryptosystem and the privacy-preserving face recognition system.

On the one hand, the extended cryptosystem has the same Birthday attack security as the original one, as the random vectors $\boldsymbol{u}$ used for encryption are chosen with the same security criterion (sparse vectors with a sufficiently small q), and they can be guessed at random with a low probability given by the security parameter λ such that $2^{(1-q)n} \cdot \binom{n}{qn} > 2^{2\lambda}$.

Regarding the dimensionality n of the lattice L and the hardness of finding the closest lattice vector without a good basis, it directly involves the γ-BDDP [101] (Bounded Distance Decoding Problem), in which given a vector $\boldsymbol{c}$, a lattice point must be found, knowing that there is at least one lattice point $\boldsymbol{p} \in L$ at a distance $\text{dist}(\boldsymbol{p}, \boldsymbol{c}) \leq \det(L)^{1/n}/\gamma$, with $\gamma > 1$. The best known algorithms for solving the γ-BDDP have exponential time-complexity in $n/\log \gamma$.[4] As our extension increases the radius of the noise in fresh encryptions with respect to the original scheme by Gentry [105] (this radius is approximately $2^k\sqrt{(1-q) \cdot n}$ for our extension, against $2\sqrt{(1-q) \cdot n}$ for the encryptions in [105]), this means that for the same lattice dimension, breaking our extended cryptosystem would imply solving the $\gamma/(2^{k-1})$-BDDP, instead of the γ-BDDP for the original cryptosystem. Hence, the complexity of the cryptanalysis algorithms based on BDDP needed for breaking an extended encryption would be on the order of $2^{(k-1) \cdot n/\log \gamma}$, and our extended cryptosystem can achieve a reasonable security level with much smaller dimension than the original one: with $n = 512$ and $k = 33$, the BDDP security is equivalent to the original system working with a 16384-dimensional lattice, at the expense of a lower degree homomorphically computable polynomial.

Additionally, the performance of the presented system is really promising, and even with a bigger lattice ($n = 2048$), execution times are comparable to those obtained with a Paillier-based system. The main drawback for even higher-dimension lattices is the increase in the size of the keys, that imposes a very high bandwidth for transferring the encryptions. In this sense, there are two research directions targeted at alleviating this problem, and they are related to reducing either the size of the keys [66], or the cipher expansion; the present work falls under the second category.

Finally, regarding the security of the private face recognition protocol, as the underlying cryptosystem is semantically secure, the whole protocol can be proven secure for semi-honest adversaries in the random oracle model. The only

[4] We refer the interested reader to the discussion in [101] by Gama and Nguyen, about the feasibility of the γ-BDDP in n dimensional lattices with $n \in [100, 400]$.

thing that a semi-honest attacker may learn from the execution of the protocol is the soft score resulting from the face comparison. This is indeed a piece of information that can be used (by a malicious attacker) in an oracle attack for extracting the information of a template for a given user, or the information for the used weight vector. If we want to restrict this kind of attacks limiting the given information to just one bit (a binary verification result), we could resort to many interactive comparison protocols present in the literature (cf. Section 6.5), like those used by Erkin *et al.* [89] or Sadeghi *et al.* [196]. Nevertheless, this would involve a final interactive step that is not desired in an autonomous outsourced system. The development of noninteractive comparison protocols using fully-homomorphic encryptions is one of the open research lines that will follow the work presented in this chapter.

6.7. Conclusions

In this chapter we propose a fully private noninteractive face recognition system, involving two novel contributions, that only when joined together allow for the sought goal: an extension of Gentry's fully homomorphic cryptosystem that allows for noninteractively computing low to medium degree polynomials with inputs of small plaintext cardinality, and an optimal quantization strategy for Gabor-based face features; when combined, these two contributions allow for the reduction of the needed representation length for a given recognition performance, and make possible the execution of the whole recognition algorithm with a reduced plaintext cardinality using only homomorphic operations and without any interaction.

The novel statistical model for the magnitude of Gabor coefficients extracted from face images is based on the assumption that both real and imaginary parts are marginally Generalized Gaussian distributed, and circularly symmetric. Thus, unlike earlier attempts to fit the magnitude of Gabor coefficients, the proposed magnitude's distribution is compatible with current GG models for the real and imaginary parts. The fitting accuracy to the data was evaluated using the Kullback-Leibler divergence on two different datasets: XM2VTS [160] and LFW [125] databases, obtaining much better results than those achieved with other previously used distributions. This model opens a wide range of independent interest applications, besides the presented data compression following a minimum MSE criterion for producing considerable savings in storage and allowing for low-cardinality plaintexts for encrypted processing of face templates.

Regarding the extension of Gentry's fully-homomorphic cryptosystem, it trades the homomorphic decryption capability for high gains in efficiency when executing low-to-medium degree arithmetic operations. We provide a bound for the number of allowed sequential multiplications, and show the performance of the cryptosystem in a practical scenario combined with Lloyd-Max quantized magnitudes of Gabor coefficients for face verification for reducing the plaintext cardinality. Contrary to traditional systems based on additive homomorphisms, the presented one allows for a completely private verification, with both encrypted templates and queried faces, opening up the possibility of outsourced noninteractive face recognition within an untrusted environment like a Cloud, being the only needed interaction in that case the initial transmission of the encrypted inputs.

Several future research lines can be highlighted: the specification of the homomorphic decryption circuit for the non-binary case; achieving other ways of decreasing the cipher expansion of the cryptosystem while keeping the good homomorphic properties, by either increasing the plaintext size or decreasing the public key size for bigger lattices; finally, providing a noninteractive solution for comparison operations and other nonlinear operations that cannot be directly mapped by the nonbinary homomorphism is also challenging.

Chapter 7

Conclusions and further work

This section briefly summarizes the conclusions that may be extracted from the research work undertaken in the present thesis.

Regarding low-level general-purpose protocols, this work presents new privacy-preserving primitives valid for semi-honest adversaries, for securely solving the N-dimensional point inclusion problem in polytopes and in hyperellyptic regions (useful in biometrics, classification, database queries, positioning and watermarking), solving systems of linear equations with direct and iterative methods, and execution of finite automata (aiming at approximate search and match and regular expression matching).

Privacy problems in adaptive filtering applications have been addressed, presenting several representative scenarios and their trust model and privacy requirements. A whole framework has been proposed for tackling the trade-off among time complexity, used bandwidth and fixed-precision error propagation in privacy-preserving implementations, while comparing novel solutions that employ different techniques, like garbled circuits, additive homomorphisms and interactive protocols; this comparison aims at the optimal trade-off in terms of complexity and output error; this work also provides several private quantization algorithms of broad applicability to tackle the cipher blowup problem, implementing all the novel protocols in a working prototype, whose analysis reveals that garbled circuits are still far from providing an efficient solution to adaptive filtering, while interactive approximate protocols with statistical security can yield much more practical solutions.

Other relevant application scenarios that are addressed in this thesis include zero-knowledge watermark detection for private detection with symmetric key schemes with improved resilience to sensitivity attacks, private cloud computing for outsourcing the processing of sensitive data to untrusted environments, and medical scenarios, prototypical of environments dealing with privacy-sensitive signals, like DNA approximate searching.

This work also takes a look towards other solutions different from encrypted processing for privacy preservation, and exemplifies them in videosurveillance, presenting a system based on the combination of an unusual configuration for DRM and a smart use of video coding and representation standards, providing an automated standardized treatment of surveillance activities and putting in the hands of the users the appropriate technical means to control the access to their private information.

Finally, foreseeing the not so distant goal of fully private noninteractive outsourced processing in untrusted environments, an extension to a recent fully-homomorphic cryptosystem is proposed, and its applicability and efficiency is showcased in a biometric face recognition application that does not use any cleartext value, but only encrypted faces, templates and parameters; it is further combined with the use of a novel model for face Gabor coefficients that allows for a high signal compression without hindering recognition performance. The results are indeed promising, proving that encrypted processing is reaching a level of development that can envisage fully privacy-protecting systems working in a completely unattended manner.

7.1. Future Research Lines

Signal Processing in the Encrypted Domain is still a young discipline, and there are many open hot topics that will be progressively tackled in the near future. The ones most directly related to the research covered in this thesis are briefly highlighted in the following points:

1. Most of the studied systems are proven secure only under a semi-honest adversary model. This model is, in some cases, too distant from real scenarios, and solutions that address the possibility of treating with malicious adversaries are needed. These solutions commonly involve an excessive overhead in terms of computation and communication load that must be optimized and lowered in future research advances.

2. The most efficient privacy-preserving protocols presented within the SPED research field mostly rely on additive homomorphic encryption, that can directly cope with linear processing, but is quite inefficient or useless when tackling non-linear operations. These nonlinear processing is commonly handled by garbled circuits or approximate interactive protocols, greatly increasing the communication burden and the complexity of the obtained solutions. The search for an optimal trade-off among the available approaches is one of the points that has been covered in this thesis, but further advances in the versatility of homomorphic processing, the efficiency of garbled circuits or the accuracy and bandwidth use of interactive protocols

may provide new insights in the trade-off optimization and tip the scales by changing the evaluated cost function.

3. As for the privacy-preserving DNA searching protocol, it constitutes the first efficient privacy-preserving solution for error-resilient DNA searching and, due to the versatility of finite state machines, the presented protocol can also be used for privately solving any problem that involves matching a string against a regular expression, such as searching a DNA database with incomplete definitions, oblivious spam checkers and virus analyzers. This work on privacy-preserving DNA queries opened a research line that has been followed by numerous subsequent works, like [134], [136], [45], [103], [119], or [36], dealing with performance optimizations and new proposals for addressing more expressive formal languages.

4. Finally, the last chapter points to the main research line in cryptography that may open the door to fully private noninteractive outsourced processing through the use of fully homomorphic cryptosystems, mainly based on lattice cryptography. This is a hot topic in the field of cryptography, and two main open research lines can be highlighted, both targeted at achieving efficient lattice cryptosystems for practical homomorphic processing by decreasing the cipher expansion of the cryptosystem while keeping good homomorphic properties: either increasing the plaintext size or decreasing the public key size for big and *secure* lattices. This would be one of the most important breakthroughs for SPED applications; on the other hand, the achievement of a noninteractive solution for nonlinear operations that, currently, cannot be directly mapped by nonbinary ring homomorphisms (without resorting to approximate solutions) would also open the door to efficiently addressing most of the open privacy-related problems in signal processing.

Bibliography

[1] Crypto++ Library.

[2] The extensible rights markup language. http://www.xrml.org.

[3] GNU MP Bignum Library.

[4] Healthgrid initiative.

[5] Human gemome project. `http://genomics.energy.gov`.

[6] Number Theory Library (NTL).

[7] Oasis standards and other approved work.

[8] P3P: The Platform for Privacy Preferences. http://www.w3.org/P3P/.

[9] Signal Processing in the EncryptEd Domain project (SPEED).

[10] Directive 95/46/EC of the European Parliament and of the Council. Official Journal L 281, 23/11/1995 P. 0031 - 0050, October 1995.

[11] Organic law 15/1999, on the protection of personal data, December 1999.

[12] Directive 2002/58/EC of the European Parliament and of the Council. Official Journal L 201, 31/07/2002 P. 0037 - 0047, July 2002.

[13] Privacy with security, December 2002. DARPA ISAT Study Group.

[14] Privacy review: video surveillance programs in Peterborough. Information and Privacy Commissioner Office, Ontario, December 2004. Report.

[15] Web Services Security Rights Expression Language (REL) Token Profile 1.1, February 2006. OASIS Standard.

[16] Web services security: Soap message security 1.1 (ws-security 2004), February 2006. OASIS Standard.

[17] *EURASIP Journal on Information Security. Special Issue on SPED.* Hindawi, December 2007.

[18] Under the watchful eye: the proliferation of video surveillance systems in California. Americal Civil Liberties Union (ACLU), August 2007. Report.

[19] ONVIF core specification ver 1.0, November 2008.

[20] The future of cloud computing. Opportunities for european cloud computing beyond 2010, January 2010. European Commission. Information Society and Media.

[21] New machine makes $1,000 human genome scans a reality. Online, January 2012. http://www.innovationnewsdaily.com/779-machine-1-000-human-genome-scans-reality.html.

[22] André Adelsbach, Stefan Katzenbeisser, and Ahmad-Reza Sadeghi. Watermark detection with zero-knowledge disclosure. In *Multimedia Systems*, volume 9, pages 266–278. Spriger-Verlag, 2003.

[23] André Adelsbach, Markus Rohe, and Ahmad-Reza Sadeghi. Overcoming the obstacles of zero-knowledge watermark detection. In *Proceedings of ACM Multimedia and Security Workshop*, pages 46–55, Magdeburg, Germany, 2004.

[24] André Adelsbach, Markus Rohe, and Ahmad-Reza Sadeghi. Complementing zero-knowledge watermark detection: Proving properties of embedded information without revealing it. *ACM Multimedia Systems Journal*, 11(2):143–158, 2005.

[25] André Adelsbach, Markus Rohe, and Ahmad-Reza Sadeghi. Non-interactive watermark detection for a correlation-based watermarking scheme. In *Communications and Multimedia Security: 9th IFIP TC-6 TC-11International Conference, CMS 2005*, volume 3677 of *Lecture Notes in Computer Science*, pages 129–139. Spriger-Verlag, September 2005.

[26] André Adelsbach and Ahmad-Reza Sadeghi. Zero-knowledge watermark detection and proof of ownership. In *Information Hiding – 4th International Workshop, IHW 2001*, volume 2137 of *Lecture Notes in Computer Science*, pages 273–288. Spriger-Verlag, 2001.

[27] Spanish Data Protection Agency. Instruction 1/2006, on processing personal data for surveillance purposes through camera or video-camera systems, November 2006.

[28] Rakesh Agrawal and Ramakrishnan Srikant. Privacy preserving data mining. In *Proc. of the 2000 ACM SIGMOD international Conference on Management of data*, pages 439–450. ACM Press, May 2000.

[29] Teodor Iulian Alecu, Sviatoslav Voloshynovsky, and Thierry Pun. The gaussian transform. In *EUSIPCO'05*, 2005.

[30] S. Alexander. Transient weight misadjustment properties for the finite precision LMS algorithm. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 35(9):1250–1258, Sep 1987.

[31] S. Thomas Alexander. *Adaptive Signal Processing*. Springer-Verlag, 1986.

[32] M. J. Atallah and W Du. Secure multiparty computational geometry. In *Proceedings of the 7th International Workshop on Algorithms and Data Structures*, volume 2125 of *Lecture Notes in Computer Science*, pages 165–179. Springer-Verlag, 2001.

[33] Mikhail J. Atallah, Florian Kerschbaum, and Wenliang Du. Secure and private sequence comparisons. In *Proceedings of the 2003 ACM Workshop on privacy in the electronic society*, pages 39–44, Washington, DC, 2003. ACM Press.

[34] Mikhail J. Atallah and Jiangtao Li. Secure outsourcing of sequence comparisons. *International Journal of Information Security*, 4(4):23–36, October 2005.

[35] D. E. Bakken, R. Parameswaran, D. M. Blough, A. A. Franz, and T. J. Palmer. Data obfuscation: Anonymity and desensitization of usable data sets. *IEEE Security and Privacy*, 2(6):34–41, 2004.

[36] Pierre Baldi, Roberta Baronio, Emiliano De Cristofaro, Paolo Gasti, and Gene Tsudik. Countering gattaca: Efficient and secure testing of fully-sequenced human genomes. In *Proceedings of the 17th ACM conference on Computer and communications security*. ACM, 2011.

[37] M. Barni and F. Bartolini. *Watermarking Systems Engineering*. Signal Processing and Communications. Marcel Dekker, 2004.

[38] M. Barni, F. Bartolini, and T. Furon. A general framework for robust watermarking security. *Signal Processing*, 82(10):2069–2084, 2003.

[39] M. Barni, P. Failla, V. Kolesnikov, R. Lazzeretti, A.-R. Sadeghi, and T. Schneider. Secure evaluation of private linear branching programs with medical applications. In *ESORICS'09*, LNCS, 2009.

[40] A. Becker, A. Arnab, and M. Serra. Assessing privacy criteria for drm using eu privacy legislation. In *Proceedings of the 8th ACM workshop on Digital Rights Management*, pages 77–86, Alexandria, Virginia, USA, October 2008.

[41] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of ACM Computer and Comm. Security*, pages 62–73. ACM Press, 1993.

[42] Richard Ernest Bellman. *Dynamic Programming.* Courier Dover Publications, 2003.

[43] T. Bianchi, A. Piva, and M. Barni. Encrypted Domain DCT based on Homomorphic Cryptosystems. *EURASIP Journal on Information Security*, 2009(Article ID 716357), 2009.

[44] T. Bianchi, A. Piva, and M. Barni. On the Implementation of the Discrete Fourier Transform in the Encrypted Domain. *IEEE Transactions on Information Forensics and Security*, 4(1):86–97, 2009.

[45] M. Blanton and M. Aliasgari. Secure outsourcing of DNA searching via finite automata. In *Data and Applications Security and Privacy XXIV*, volume 6166 of *LNCS*, pages 49–64. Springer, 2010.

[46] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search. In C. Cachin and J. Camenisch, editors, *EUROCRYPT'04*, number 3027 in LNCS, pages 506–522. Springer, 2004.

[47] Fabrice Boudot. Efficient proofs that a committed number lies in an interval. In *Advances in cryptology - EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 431–444. Springer-Verlag, 2000.

[48] Gilles Brassard, David Chaum, and Claude Crépeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37(2):156–189, October 1988.

[49] Ernest F. Brickell and Yacov Yacobi. On privacy homomorphisms (extended abstract). In *Advances in Cryptology-EUROCRYPT'87*, Lecture Notes in Computer Science, pages 117–125. Springer-Verlag, 1987.

[50] R. Brinkman. *Searching in Encrypted Data.* PhD thesis, University of Twente, 2007.

[51] R. Brinkman, J. M. Doumen, and W. Jonker. Using secret sharing for searching in encrypted data. In *Workshop on Secure Data Management in a Connected World (SDM 2004)*, volume 3178 of *Lecture Notes in Computer Science*, pages 18–27. Springer-Verlag, 2004.

[52] R. Canetti, Y. Ishai, R. Kumar, M. Reiter, R. Rubinfeld, and R. Wright. Selective private function evaluation with applications to private statistics. In *Proc. 20th Annual ACM Symposium on Principles of Distributed Computing*, pages 293–304. ACM Press, 2001.

[53] J. Canny. Collaborative filtering with privacy. In *Security and Privacy, 2002. Proceedings. 2002 IEEE Symposium on*, pages 45 – 57, 2002.

[54] C. Caraiscos and Bede Liu. A roundoff error analysis of the LMS adaptive algorithm. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 32(1):34–41, Feb 1984.

[55] P. Carrillo, H. Kalva, and S. Magliveras. Compression independent object encryption for ensuring privacy in video surveillance. In *IEEE International Conference on Multimedia and Expo*, 2008.

[56] A. Cavallaro. Privacy in video surveillance [in the spotlight]. *IEEE Signal Processing Magazine*, 24(2), March 2007.

[57] A. Chattopadhyay and T.E. Boult. Privacycam: a privacy preserving camera using uCLinux on the Blackfin DSP. In *IEEE Conference on computer vision and pattern recognition, CVPR'07*, Minneapolis, MN, USA, 17-22 June 2007.

[58] Bernard Chazelle and Joel Friedman. Point location among hyperplanes and unidirectional ray-shooting. *Computational Geometry: Theory and Applications*, 4:53–62, 1994.

[59] B. Chen and G. Wornell. Quantization index modulation: a class of provably good methods for digital watermarking and information embedding. *IEEE Transactions on Information Theory*, 47:1423–1443, May 2001.

[60] Datong Chen, Yi Chang, Rong Yan, and Jie Yang. Tools for protecting the privacy of specific individuals in video. *EURASIP Journal on Advances in Signal Processing*, 2007:1–9, 2007.

[61] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults. In *25th Annual Symposium on Foundations of Computer Science FOCS'85*, pages 383–395. IEEE Computer Society, 1985.

[62] Richard Chow, Philippe Golle, Markus Jakobsson, Elaine Shi, Jessica Staddon, Ryusuke Masuoka, and Jesus Molina. Controlling data in the cloud: outsourcing computation without outsourcing control. In *Proceedings of the 2009 ACM workshop on Cloud computing security*, CCSW '09, pages 85–90, New York, NY, USA, 2009. ACM.

[63] G. Clark, S. Mitra, and S. Parker. Block implementation of adaptive digital filters. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 29(3):744–752, June 1981.

[64] Pedro Comesaña, Luis Pérez Freire, and Fernando Pérez-González. Blind newton sensitivity attack. *IEE Proceedings on Information Security*, 153(3):115–125, September 2006.

[65] Pedro Comesaña and Fernando Pérez-González. Breaking the bows watermarking system: Key guessing and sensitivity attacks. *EURASIP Journal on Information Security*, 2007.

[66] Jean-SÃ©bastien Coron, Avradip Mandal, David Naccache, and Mehdi Tibouchi. Fully homomorphic encryption over the integers with shorter public keys. In *Advances in Cryptology - CRYPTO11*, volume 6841, page 483, 2011.

[67] T.M. Cover and J.A. Thomas. *Elements of Information Theory.* John Wiley & Sons, New York, 1991.

[68] I. J. Cox, J. Killian, T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for images, audio and video. *IEEE Transactions on Image Processing*, 6:1673–1687, December 1997.

[69] R. Cramer, I Damgård, and J.B. Nielsen. Multiparty computation from threshold homomorphic encryption. In *Advances in Cryptology EUROCRYPT'01*, volume 2045 of *LNCS*, pages 280–300. Springer-Verlag, October 2001.

[70] Ronald Cramer and Ivan Damgård. Secure distributed linear algebra in a constant number of rounds. In *21st Annual International Cryptology Conference on Advances in Cryptology*, volume 2139 of *Lecture Notes in Computer Science*, pages 119–136. Springer, 2001.

[71] Ronald Cramer, Ivan Damgård, Jesper B. Nielsen, and Birgit Pfitzmann. Multiparty computation from threshold homomorphic encryption. In *Advances in cryptology - EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 280–300. Springer, May 2001.

[72] Scott Craver. Zero knowledge watermark detection. In *Information Hiding: Third International Workshop*, volume 1768 of *Lecture Notes in Computer Science*, pages 101–116. Springer, 2000.

[73] Nello Cristianini and John Shawe-Taylor. *Support Vector Machines and other kernel-based learning methods.* Cambridge University Press, 2000.

[74] Germund Dahlquist and Åke Björck. *Numerical methods.* Dover Publications, 2003.

[75] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. volume 1, pages 886–893, Los Alamitos, CA, USA, 2005.

[76] Ivan Damgård. Commitment schemes and zero-knowledge protocols. In *Lectures on data security: modern cryptology in theory and practise*, volume 1561 of *Lecture Notes in Computer Science*, pages 63–86. Springer-Verlag, 1998.

[77] Ivan Damgård, Matthias Fitzi, Eike Kiltz, Jesper Buus Nielsen, and Tomas Toft. Unconditionally secure constant-rounds multi-party computation for equality, comparison, bits and exponentiation. In *Proceedings of the third Theory of Cryptography Conference, TCC 2006*, volume 3876 of *Lecture Notes in Computer Science*, pages 285–304. Springer-Verlag, 2006.

[78] Ivan Damgård and Eiichiro Fujisaki. A statistically-hiding integer commitment scheme based on groups with hidden order. In *ASIACRYPT 2002: 8th International Conference on the Theory and Application of Cryptology and Information Security*, volume 2501 of *Lecture Notes in Computer Science*, pages 125–142. Spriger-Verlag, December 2002.

[79] Ivan Damgård and Mads Jurik. A generalisation, a simplification and some applications of paillier's probabilistic public-key system. In Kwangjo Kim, editor, *Public Key Crytography 2001*, volume 1992 of *Lecture Notes in Computer Science*, pages 119–136, Cheju Island, Korea, February 2001. Springer.

[80] Ivan Damgård and Mads Jurik. Client/server tradeoffs for online elections. In *Proceedings of the 5th International Workshop on Practice and Theory in Public Key Cryptosystems: Public Key Cryptography*, volume 2274 of *Lecture Notes in Computer Science*, pages 125–140, Paris, France, February 2002. Springer-Verlag.

[81] J. G. Daugman. Complete Discrete 2D Gabor Transforms by Neural Networks for Image Analysis and Compression. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 36(7):1169 – 1179, July 1988.

[82] Jacques Stern David Naccache. A new public key cryptosystem based on higher residues. In *Conference on Computer and Communications Security, Proceedings of the 5th ACM conference on Computer and communications security*, pages 59–66, San Francisco, California, United States, 1998.

[83] Mark de Berg, Marc van Kerveld, Mark Overmars, and Otfried Schwarzkopf. *Computational Geometry, Algorithms and Applications.* Springer-Verlag Berlin, 2nd edition edition, 2000.

[84] Yvo Desmedt and Yair Frankel. Threshold cryptosystems. In *Proceedings of Crypto 1989*, pages 307–315, Santa Barbara, California, USA, 1989.

[85] M.N. Do and M. Vetterli. Wavelet-Based Texture Retrieval Using Generalized Gaussian Density and Kullback-Leibler Distance. *IEEE TIP*, 11(2):146 – 158, 2002.

[86] W Du and M. J. Atallah. Privacy-preserving cooperative scientific computations. In *Proceedings of the 14th IEEE Computer Security Foundations Workshop*, pages 273–282, Nova Scotia, Canada, June 2001.

[87] Frédéric Dufaux, Mourad Ouaret, Yousri Abdeljaoued, Alfonso Navarro, Fabrice Vergnenègre, and Touradj Ebrahimi. Privacy enabling technology for video surveillance. In Sos S. Agaian and Sabah A. Jassim, editors, *Mobile Multimedia/Image Processing for Military and Security Applications*, volume 6250. SPIE, 2006.

[88] J.J. Eggers and B. Girod. *Informed Watermarking.* Kluwer Academic Publishers, 2002.

[89] Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Lagendijk, and T. Toft. Privacy-preserving face recognition. In *Privacy Enhancing Technologies Symposium, PETS'09*, number 5672 in LNCS, pages 235–253. Springer, 2009.

[90] Z. Erkin, T. Veugen, T. Toft, and R. L. Lagendijk. Privacy-Preserving Centralized Recommender System. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2010.

[91] Europasur. *Gmail, el correo espía de Google, ilegal en Europa.* January 25 2008. Retrieved August 16, 2010 from http://www.europasur.es/article/sociedad/38601/gmail /correo/espia/google/ilegal/europa.html.

[92] Eweda Eweda, Nabil Yousef, and Salwa El-Ramly. Reducing the Effect of Finite Wordlength on the Performance of an LMS Adaptive Filter. In *EEE International Conference on Signal Communications*, pages 688–692, Atlanta, GA, USA, June 1998.

[93] Mark M. Fisch, Herbert Stögner, and Andreas Uhl. Layered encryption techniques for DCT-coded visual data. In *European Signal Processing Conference on Signal Processing, EUSIPCO'04*, 2004.

[94] C. Fonteneau, J. Motscha, M. Babela, and O. Déforges. A hierarchical selective encryption technique in a scalable image codec. In *International Conference in Communications*, Bucharest, Romania, 2008.

[95] M. Franz, S. Katzenbeisser, S. Jha, K. Hamacher, H. Schroeder, and B. Deiseroth. Secure computations on non-integer values. In *IEEE WIFS'10*, Seattle, USA, December 2010. IEEE.

[96] Eiichiro Fujisaki and Tatsuaki Okamoto. A practical and provably secure scheme for publicly verifiable secret sharing and its applications. In *Proceedings of EUROCRYPT'98*, volume 1403 of *Lecture Notes in Computer Science*, pages 32–46. Springer, 1998.

[97] T. Furon. A survey of watermarking security. In M. Barni, editor, *Proc. of Int. Work. on Digital Watermarking*, volume 3710 of *Lecture Notes on Computer Science*, pages 201–215, Siena, Italy, sep 2005. Springer-Verlag.

[98] T. Furon, B. Macq, N. Hurley, and G. Silvestre. JANIS: Just Another N-order side-Informed watermarking Scheme. In *IEEE International Conference on Image Processing, ICIP'02*, volume 3, pages 153–156, Rochester, NY, USA, September 2002.

[99] S. Zacchiroli G. D'Angelo, F. Vitali. Content cloaking: Preserving privacy with google docs and other web applications. In *2010 ACM Symposium on Applied Computing*, pages 22–26, Sierre, Switzerland, March 2010.

[100] M. Viola de Azevedo Cunha G. Sartor. *The Italian Google-Case: Privacy, Freedom of Speech and Responsibility of Providers for User-Generated Contents*. Oxford University Press, May 11 2010.

[101] N. Gama and P. Q. Nguyen. Predicting lattice reduction. In *EUROCRYPT 2008*, volume 4965 of *LNCS*. Springer, 2008.

[102] D.M. Gavrila and V. Philomin. Real-time object detection for "smart" vehicles. In *Seventh IEEE International Conference on Computer Vision*, volume 1, pages 87–93, 1999.

[103] Rosario Gennaro, Carmit Hazay, and Jeffrey Sorensen. Text search protocols with simulation based security. In *Public Key Cryptography - PKC 2010*, volume 6056 of *LNCS*, pages 332–350. Springer, 2010.

[104] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st annual ACM symposium on Theory of computing, STOC'09*, pages 169–178, Bethesda, MD, USA, May-June 2009. ACM Press.

[105] Craig Gentry and Shai Halevi. Implementing gentry's fully-homomorphic encryption scheme. In *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 129–148. Springer, 2011.

[106] Bart Goethals, Sven Laur, Helger Lipmaa, and Taneli Mielikäinen. On private scalar product computation for privacy-preserving data mining. In Choonsik Park and Seongtaek Chee, editors, *7th Annual International Conference in Information Security and Cryptology (ICISC 2004)*, volume 3506 of *Lecture Notes in Computer Science*, pages 104–120, Seoul, Korea, December 2004. Springer.

[107] O. Goldreich, S. Micali, and A. Widgerson. How to play any mental game. In *Proceedings of the nineteenth annual ACM conference on Theory of Computing*, pages 218–229, New York, U.S.A., 1987. ACM Press.

[108] Oded Goldreich. Concurrent zero-knowledge with timing, revisited. In *34th Symposium on the Theory of Computing*, pages 332–340, 2002.

[109] Oded Goldreich. Zero-knowledge twenty years after its invention. Technical report, Electronic Colloquium on Computational Complexity, 2002.

[110] Oded Goldreich and Erez Petrank. Quantifying knowledge complexity. *Computational Complexity*, 8(1):50–98, 1999.

[111] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

[112] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems. In *Proceedings of the 17th Annual ACM Symposium on the Theory of Computing*, pages 291–304, 1985.

[113] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems. In *SIAM Journal of Computing*, volume 18, pages 186–208, 1989.

[114] D. González-Jiménez, F. Pérez-González, P. Comesaña-Alfaro, L. Pérez-Freire, and J.L. Alba-Castro. Modeling Gabor Coefficients via Generalized Gaussian Distributions for Face Recognition. In *ICIP*, 2007.

[115] Daniel González-Jiménez, Enrique Argones-Rúa, Fernando Pérez-González, and José Luis Alba-Castro. Modeling Magnitudes of Gabor Coefficients: the β-Rayleigh Distribution. In *IEEE International Conference on Image Processing*, Cairo, Egypt, November 2009. IEEE.

[116] K. Gopalakrishnan, Nasir D. Memon, and Poorvi Vora. Protocols for watermark verification. In *Multimedia and Security Workshop at ACM Multimedia*, pages 91–94, 1999.

[117] A. Hall. Coming soon: Your personal dna map? `http://news.nationalgeographic.com/news/2006/03/0307_060307_dna.html`.

[118] Simon Haykin. *Adaptive Filter Theory*. Information and System Sciences. Prentice Hall, fourth edition edition, 2002.

[119] Carmit Hazay and Tomas Toft. Computationally secure pattern matching in the presence of malicious adversaries. In *Advances in Cryptology - ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 195–212. Springer, 2010.

[120] J.R. Hernández, M. Amado, and F. Pérez-González. DCT-domain watermarking techniques for still images: Detector performance analysis and a new structure. *IEEE TIP*, 9(1):55–68, January 2000. Special Issue on Image and Video Processing for Digital Libraries.

[121] Juan R. Hernández, Martín Amado, and Fernando Pérez-González. DCT-domain watermarking techniques for still images: Detector performance analysis and a new structure. *IEEE Transactions on Image Processing*,

9(1):55–68, January 2000. Special Issue on Image and Video Processing for Digital Libraries.

[122] Heinz Hofbauer, Thomas Stütz, and Andreas Uhl. Selective encryption for hierarchical MPEG. In Springer Berlin / Heidelberg, editor, *Communications and Multimedia Security*, volume 4237/2006 of *Lecture Notes in Computer Science*, pages 151–160, 2006.

[123] John E. Hopcroft and Jeffrey D. Ullman. *Introduction to automata theory, languages, and computation*. Addison Wesley, 1979.

[124] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

[125] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments, Technical Report 07-49. Technical report, University of Massachusetts, Amherst,, 2007.

[126] Neil J. Hurley and Guenole C. M. Silvestre. Nth-order audio watermarking. In Edward J. Delp III and Ping Wah Wong, editors, *Security and Watermarking of Multimedia Contents IV*, volume 4675 of *Proc. of SPIE*, pages 102–109, San José, CA, USA, 2002.

[127] Maged Hamada Ibrahim. Two-party private vector dominance: The all-or-nothing deal. In *Third International Conference on Information Technology: New Generations, 2006. ITNG 2006*, pages 166–171, April 2006.

[128] Business Insider. *Google Pulls Search Engine out of China*. March 22 2010. Retrieved August 16, 2010 from http://www.businessinsider.com/google-pulls-out-of-china-2010-3.

[129] Y. Ishai, J. Kilian, K. Nissim, and E. Petrank. Extending Oblivious Transfer Efficiently. In *Advances in Cryptology CRYPTO'03*, volume 2729, pages 145–161, 2003.

[130] Markus Jacobsson and Ari Juels. Mix and match: Secure function evaluation via ciphertexts. In T. Okamoto, editor, *Advances in Cryptology – ASIACRYPT'00*, volume 1976 of *Lecture Notes in Computer Science*, pages 162–177. Springer-Verlag, 2000.

[131] Geetha Jagannathan, Krishnan Pillaipakkamnatt, and D. Umano. A Secure Clustering Algorithm for Distributed Data Streams. In *Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on*, pages 705 –710, 28-31 2007.

[132] Geetha Jagannathan, Krishnan Pillaipakkamnatt, and Rebecca N. Wright. A new privacy-preserving distributed k-clustering algorithm. In *Proceedings of the Sixth SIAM International Conference on Data Mining*, 2006.

[133] M. Jensen, J.O. Schwenk, N. Gruschka, and L.L. Iacono. On technical security issues in cloud computing. In *Proceedings of the IEEE International Conference on Cloud Computing, 2009 (CLOUD '09)*, pages 109–116, Bangalore, India, September 2009.

[134] Somesh Jha, Louis Kruger, and Vitaly Shmatikov. Towards practical privacy for genomic computation. In *IEEE Symposium on Security and Privacy*, pages 216–230, Los Alamitos, CA, USA, May 2008. IEEE.

[135] M. Johnson, P. Ishwar, V. Prabhakaran, D. Schonberg, and K. Ramchandran. On compressing encrypted data. *IEEE Transactions on Signal Processing*, 52(10):2992–3006, October 2004.

[136] Jonathan Katz and Lior Malka. Secure text processing with applications to private dna matching. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 485–492, New York, NY, USA, 2010. ACM.

[137] Stefan Katzenbeisser. On the integration of watermarks and cryptography. In *International Workshop on Digital Watermarking*, pages 50–60, October 2003.

[138] Steve Kenny and Larry Korba. Applying digital rights management systems to privacy rights management. *Journal of Computers and Security*, November, 2002.

[139] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48(177):203–209, January 1987.

[140] V. Kolesnikov, A.-R. Sadeghi, and T. Schneider. How to Combine Homomorphic Encryption and Garbled Circuits – Improved Circuits and Computing the Minimum Distance Efficiently. In *SPEED Workshop*, pages 100–121, Lausanne, Switzerland, September 2009.

[141] Vladimir Kolesnikov and Thomas Schneider. Improved garbled circuit: Free XOR gates and applications. In *ICALP'08*, volume 5126 of *LNCS*, pages 486–498. Springer, 2008.

[142] Louis Kruger, Somesh Jha, Eu-Jin Goh, and Dan Boneh. Secure function evaluation with ordered binary decision diagrams. In *Proceedings of the 13th ACM conference on Computer and communications security CCS'06*, pages 410–420, Virginia, U.S.A., November 2006. ACM Press.

[143] A. K. Lenstra, H. W. Lenstra, and L. LovÃ¡sz. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:515–534, 1982.

[144] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Doklady Akademii Nauk SSSR*, 163(4):845–848, 1965. English translation at *Soviet Physics Doklady* 10(8): 707–710, 1966.

[145] Yehuda Lindell and Benny Pinkas. Privacy preserving data mining. *J. Cryptology*, 15(3):177–206, 2002.

[146] Helger Lipmaa. On diophantine complexity and statistical zero-knowledge arguments. In *ASIACRYPT 2003*, volume 2894 of *Lecture Notes in Computer Science*, pages 398–415. Spriger-Verlag, November 2003.

[147] Helger Lipmaa. Verifiable homomorphic oblivious transfer and private equality test. In *ASIACRYPT'01*, volume 2894 of *LNCS*. Springer, 2003.

[148] Helger Lipmaa. An oblivious transfer protocol with log-squared communication. In *ISC'05*, pages 314–328, 2005.

[149] Kun Liu, Hillol Kargupta, and Jessica Ryan. Multiplicative noise, random projection, and privacy preserving data mining from distributed multi-party data. Technical Report TR-CS-03-24, Computer Science and Electrical Engineering Department, University of Mariland, Baltimore County, 2003.

[150] S.P. Lloyd. Least Squares Quantization in PCM. Technical report, Bell Laboratories, 1957.

[151] Karl Martin and Konstantinos N. Plataniotis. Privacy protected surveillance using secure visual object coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(8):1152–1162, 2008.

[152] Isabel Martínez-Ponte, Xavier Desurmont, Jerome Meessen, and Jean-Fran cois Delaigle. Robust human face hiding ensuring privacy. In *International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS'05*, Montreux, Switzerland, April 2005.

[153] J. Matoušek and O. Schwarzkopf. On ray shooting in convex polytopes. *Discrete Computational Geometry*, 10:215–232, 1993.

[154] J. Max. Quantizing for Minimum Distortion. *IRE Transactions on Information Theory*, IT-6:7–12, 1960.

[155] Michael McCahill. *The Surveillance Web: The Rise of Visual Surveillance in an English City.* Willan Publishing (UK), 2002.

[156] Michael McCahill and Clive Norris. Cctv in London. Report deliverable of UrbanEye project, 2002.

[157] Qi Meibing, Chen Xiaorui, Jiang Jianguo, and Zhan Shu. Face protection of h.264 video based on detecting and tracking. In *8th International Conference on Electronic Measurement and Instruments, ICEMI'07*, volume 2, pages 172–177, 16-18 July 2007.

[158] Carlos Aguilar Melchor and Philippe Gaborit. Single-database private information retrieval protocols: Overview, usability and trends. Technical report, University of Limoges, 2007.

[159] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography.* CRC Press, 2001. 5th reprint.

[160] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre. XM2VTSDB: The Extended M2VTS Database. *AVBPA*, pages 72 – 77, March 1999.

[161] Daniele Micciancio. Improving lattice based cryptosystems using the hermite normal form. In Joseph Silverman, editor, *Cryptography and Lattices Conference — CaLC 2001*, volume 2146 of *Lecture Notes in Computer Science*, pages 126–145, Providence, Rhode Island, 29–30March 2001. Springer-Verlag.

[162] Oded Goldreich Mihil Bellare. On defining proofs of knowledge. In *Proceedings of Crypto'92*, volume 740 of *Lecture Notes in Computer Science*, pages 390–420. Springer-Verlag, 1992.

[163] Matthew L. Miller. Is asymmetric watermarking necessary or sufficient? In *Proc. XI European Signal Processing Conference, EUSIPCO'02*, pages 291–294, 2002.

[164] M. Mohri. On some application of finite-state automata theory to natural language. *Natural Language Engineering*, 2(1):1–20, 1996.

[165] M. Mohri. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23(2):269–311, 1997.

[166] Moni Naor and Kobbi Nissim. Communication complexity and secure function evaluation. *Electronic Colloquium on Computational Complexity (ECCC)*, 8(062), 2001.

[167] Moni Naor and Kobbi Nissim. Communication preserving protocols for secure function evaluation. In *ACM Symposium on Theory of Computing*, pages 590–599, 2001.

[168] Moni Naor and Benni Pinkas. Efficient oblivious transfer protocols. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 448–457, Washington, D.C., U.S.A., 2001.

[169] S. Hasan Naqvi and L. M. Patnaik. A medium access protocol exploiting multiuser-detection in cdma ad-hoc networks. *Wirel. Netw.*, 16:1723–1737, August 2010.

[170] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal on Molecular Biology*, 48:443–453, 1970.

[171] Elaine M. Newton, Latanya Sweeney, and Bradley Malin. Preserving privacy by de-identifying face images. *IEEE Transactions on Knowledge and Data Engineering*, 17(2):232–243, February 2005.

[172] Hieu V. Nguyen and Li Bai. Cosine similarity metric learning for face verification. In *ACCV (2)*, pages 709–720, 2010.

[173] Takashi Nishide and Kazuo Ohta. Constant-round multiparty computation for interval test, equality test, and comparison. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E90-A(5):960–968, May 2007.

[174] Seev Neumann Niv Ahituv, Yeheskel Lapid. Processing encrypted data. *Communications of the ACM*, 20(9):777–780, 1987.

[175] T. Okamoto and S. Uchiyama. A new public key cryptosystem as secure as factoring. In *Eurocrypt '98*, volume 1403 of *Lecture Notes in Computer Science*, pages 308–318. Springer-Verlag, 1998.

[176] C. Orlandi, A. Piva, and M. Barni. Oblivious neural network computing via homomorphic encryption. *EURASIP Journal on Information Security*, 2007, 2007. Article ID 37343.

[177] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology – EUROCRYPT'99*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238. Springer-Verlag, 1999.

[178] Su-Wan Park and Sang-Uk Shin. Efficient selective encryption scheme for the H.264/Scalable Video Coding(SVC). In *Fourth International Conference on Networked Computing and Advanced Information Management*, volume 1, pages 371–376, Gyeongju, Korea, September 2-4 2008.

[179] Siani Pearson, Yun Shen, and Miranda Mowbray. A privacy manager for cloud computing. In *Cloud Computing*, volume 5931 of *Lecture Notes in Computer Science*, pages 90–106. Springer Berlin / Heidelberg, 2009.

[180] Luis Pérez-Freire, Pedro Comesaña, and Fernando Pérez-González. Detection in quantization-based watermarking: performance and security issues. In Edward J. Delp III and Ping Wah Wong, editors, *Security, Steganography, and Watermarking of Multimedia Contents VII*, volume 5681 of *Proc. of SPIE*, pages 721–733, San José, USA, January 2005.

[181] Luis Pérez-Freire, Pedro Comesaña, Juan Ramón Troncoso-Pastoriza, and Fernando Pérez-González. Watermarking security: a survey. *LNCS Transactions on Data Hiding and Multimedia Security I*, 4300:41–72, October 2006.

[182] Fernando Pérez-González, Félix Balado, and Juan R. Hernández. Performance analysis of existing and new methods for data hiding with known-host information in additive channels. *IEEE Transactions on Signal Processing*, 51:960–980, April 2003.

[183] N. Pinto and D. D. Cox. Beyond Simple Features: A Large-Scale Feature Search Approach to Unconstrained Face Recognition. In *IEEE Automatic Face and Gesture Recognition*, 2011.

[184] N. Pinto, JJ. Dicarlo, and DD. Cox. Establishing good benchmarks and baselines for face recognition. In *IEEE ECCV*, 2008. Faces in 'Real-Life' Images Workshop.

[185] Ioannis Pitas. A method for signature casting on digital images. In *Proceedings of ICIP*, volume 3, pages 215–218, 1996.

[186] A. Piva, V. Cappellini, D. Corazzi, A. De Rosa, C. Orlandi, and M. Barni. Zero-knowledge st-dm watermarking. In Edward J. Delp III and Ping W. Wong, editors, *Security, Steganography, and Watermarking of Multimedia Contents VIII, SPIE*, San José, California, USA, January 2006.

[187] Alessandro Piva, D. Corazzi, Alessia De Rosa, and Mauro Barni. Zero knowledge st-dm watermarking. In Edward J. Delp III and Ping W. Wong, editors, *Security, Steganography, and Watermarking of Multimedia Contents VIII, SPIE*, San José, California, USA, January 2006.

[188] F. P. Preparata and R. Tamassia. Efficient point location in a convex spatial cell-complex. *SIAM Journal on Computing*, 21:267–280, 1992.

[189] The Register. *US town tells Street View to push off.* June 2 2008. Retrieved August 16, 2010 from http://www.theregister.co.uk/2008/06/02/north_oaks_street_view/.

[190] Ronald L. Rivest, Len Adleman, and Michael L. Dertouzos. On data banks and privacy homomorphisms. *Foundations of Secure Computation*, pages 169–179, 1978.

[191] Ronald L. Rivest, Adi Shamir, and Len Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–176, 1978.

[192] Daniel A. Rodríguez-Silva, F. Javier González-Castaño, Lilian Adkinson-Orellana, Alexandre Fernández-Cordeiro, Juan Ramón Troncoso-Pastoriza, and Daniel González-Martínez. Encrypted Domain Processing for Cloud Privacy: Concept and Practical Experience. In *International Conference on Cloud Computing and Services Science (CLOSER 2011)*, Noordwijkerhout, The Netherlands, May 2011.

[193] Berry Schoenmakers Ronald Cramer, Ivan Damgård. Proofs of partial knowledge and simplified design of witness hiding protocols. In *Proceedings of CRYPTO'94*, volume 839 of *Lecture Notes in Computer Science*, pages 174–187. Springer, 1997.

[194] Xie S., Shan S., Chen X, and Chen J. Fusing local patterns of gabor magnitude and phase for face recognition. *IEEE Transactions on Image Processing*, 19(5):1349–1361, 2010.

[195] K. Lauter S. Kamara. Cryptographic cloud storage. In *Financial Cryptography: Workshop on Real-Life Cryptographic Protocols and Standardization 2010*, January 2010.

[196] A.-R. Sadeghi, T. Schneider, and I. Wehrenberg. Efficient privacy-preserving face recognition. In *ICISC 2009*, volume 5984 of *LNCS*, pages 229–244. Springer, 2010.

[197] Yingpeng Sang and Hong Shen. A scheme for testing privacy state in pervasive sensor networks. In *Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA'05)*, volume 2, pages 644–648, March 2005.

[198] B. Schneier. *Applied cryptography*. Computer Networking and Distributed Systems. John Wiley & Sons, 1994.

[199] Berry Schoenmakers and Pim Tuyls. Efficient binary conversion for paillier encrypted values. In *Advances in Cryptology - EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 522–537. Springer, 2006.

[200] Andrew Senior, Sharath Pankanti, Arun Hampapur, Lisa Brown, Ying-Li Tian, and Ahmet Ekin. Blinkering surveillance: enabling video privacy through computer vision. Technical report, IBM Research Division, August 2003.

[201] Ángel Serrano, Isaac Martín de Diego, Cristina Conde, and Enrique Cabello. Recent advances in face biometrics with gabor wavelets: A review. *Pattern Recognition Letters*, 31(5):372 – 381, 2010.

[202] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[203] K. Sharifi and A. Leon-Garcia. Estimation of Shape Parameter for Generalized Gaussian Distributions in Subband Decompositions of Video. *IEEE Transactions on Circuits and Systems for Video Technology*, 5(1):52–56, 1995.

[204] L. Shen and L. Bai. A Review on Gabor Wavelets for Face Recognition. *Pattern Analysis and Applications*, 9(2):273 – 292, 2006.

[205] Victor Shoup. Practical threshold signatures. In *Advances in cryptology - EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 207–220. Springer, 2000.

[206] Klaus U. Shulz and S. Mihov. Fast string correction with levenshtein automata. *International Journal of Document Analysis and Recognition (IJDAR)*, 5(1):67–85, 2002.

[207] Jean-Jacques E. Slotine and Weiping Li. *Applied Nonlinear Control*. Prentice Hall, 1991.

[208] N.P. Smart and F. Vercauteren. Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes. In *13th International Conference on Practice and Theory in Public Key Cryptography 2010*, volume 6056 of *LNCS*, pages 420–443, Paris, France, May 2010.

[209] D. X. Song, D. Wagner, and A. Perrig. Practical techniques for searches on encrypted data. In *Proceedings of the 2000 IEEE symposium on Security and Privacy*, pages 44–55, 2000.

[210] Spiegel. *Google Prepares Street View Launch in Germany.* August 10 2010. Retrieved August 16, 2010 from http://www.spiegel.de/international/germany/ 0,1518,711090,00.html.

[211] Hiranmayee Subramaniam, Rebecca N. Wright, and Zhiqiang Yang. Experimental analysis of privacy-preserving statistics computation. In *Proc. of the VLDB Worshop on Secure Data Management*, pages 55–66, August 2004.

[212] Suriyon Tansuriyavong and Shin ichi Hanaki. Privacy protection by concealing persons in circumstantial video image. In *ACM Workshop on Perceptive user interfaces*, 2001.

[213] Nick Taylo. State surveillance and the right to privacy. *Surveillance and Society*, 1(1):66–85, 2002.

[214] T.Bianchi, A.Piva, and M.Barni. Composite Signal Representation for Fast and Storage-Efficient Processing of Encrypted Signals. *IEEE Trans. on Information Forensics and Security*, 5(1):180–187, March 2010.

[215] J.R. Troncoso-Pastoriza, D. González-Jiménez, and F. Pérez-González. A new model for Gabor Coefficients' Magnitude in Face Recognition. In *IEEE ICASSP 2010*, Dallas, USA, March 2010. IEEE.

[216] J.R. Troncoso-Pastoriza, D. González-Jiménez, and F. Pérez-González. Fully private noninteractive face verification. *IEEE Transactions on Information Forensics and Security*, 2012. Submitted.

[217] J.R. Troncoso-Pastoriza and F. Pérez-González. Fully Homomorphic Faces. In *IEEE ICIP 2012*. IEEE, 2012. Submitted.

[218] Juan Ramón Troncoso-Pastoriza. PrivateLMS: Prototype Protocols for the Private Execution of the LMS algorithm, August 2010.

[219] Juan Ramón Troncoso-Pastoriza, Pedro Comesaña, Luis Pérez-Freire, and Fernando Pérez-González. Videosurveillance and privacy: covering the two sides of the mirror with DRM. In *ACM Workshop on Digital Rights Management*, Chicago, IL, USA, November 2009. ACM.

[220] Juan Ramón Troncoso-Pastoriza, Pedro Comesaña, and Fernando Pérez-González. Secure Direct and Iterative Protocols for Solving Systems of Linear Equations. In *Proceedings of the SPEED Workshop 2009*, pages 122–141, Lausanne, Switzerland, September 2009.

[221] Juan Ramón Troncoso-Pastoriza, Stefan Katzenbeisser, and Mehmet Celik. Privacy preserving error resilient dna searching through oblivious automata. In *14th ACM Conference on Computer and Communications Security*, pages 519–528, Alexandria, Virginia, USA, October 29–November 2 2007. ACM Press.

[222] Juan Ramón Troncoso-Pastoriza and Fernando Pérez-González. Zero-Knowledge watermark detector robust to sensitivity attacks. In *8th ACM Multimedia and Security Workshop*, pages 97–107, Geneva, Switzerland, September 2006. ACM.

[223] Juan Ramón Troncoso-Pastoriza and Fernando Pérez-González. Efficient Non-Interactive Zero-Knowledge Watermark Detector Robust to Sensitivity Attacks. In Edward J. Delp III and Ping W. Wong, editors, *Security, Steganography, and Watermarking of Multimedia Contents IX*, San Jose, California, USA, January 2007. SPIE.

[224] Juan Ramón Troncoso-Pastoriza and Fernando Pérez-González. Efficient zero-knowledge watermark detection with improved robustness to sensitivity attacks. *EURASIP Journal on Information Security*, 2007. Special Issue on Signal Processing in the Encrypted Domain.

[225] Juan Ramón Troncoso-Pastoriza and Fernando Pérez-González. CryptoDSPs for Cloud Privacy. In *CISE 2010*, volume 6724 of *LNCS*, Hong Kong, China, December 2010.

[226] Juan Ramón Troncoso-Pastoriza and Fernando Pérez-González. Secure and Private Medical Clouds using Encrypted Processing. In *VPH Conference 2010*, Brussels, Belgium, October 2010. VPH NoE.

[227] Juan Ramón Troncoso-Pastoriza and Fernando Pérez-González. University of Vigo Technical Report UV/DTC/JRTP/22/12/2010. Technical report, University of Vigo, September 2010.

[228] Juan Ramón Troncoso-Pastoriza and Fernando Pérez-González. Efficient Protocols for Secure Adaptive Filtering. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2011)*, pages 5860–5863, Prage, Czech Republic, May 2011. IEEE.

[229] Juan Ramón Troncoso-Pastoriza and Fernando Pérez-González. Secure Adaptive Filtering. *IEEE Trans. on Information Forensics and Security*, 6(2):469–485, June 2011.

[230] Juan Ramón Troncoso-Pastoriza, Stefan Katzenbeisser, Mehmet Celik, and Aweke Lemma. A secure multidimensional point inclusion protocol. In *9th ACM Workshop on Multimedia and Security (MMSEC'07)*, pages 109–120, Dallas, Texas, USA, September 2007.

[231] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *J. Cognitive Neuroscience*, 3:71–86, January 1991.

[232] Pim Tuyls and Jasper Goseling. Capacity and examples of template protecting biometric authentication systems. In *Proceedings of Biometric Authentication Workshop*, volume 3087 of *Lecture Notes in Computer Science*, pages 158–170, Berlin, 2004. Springer-Verlag.

[233] Jaideep Vaidya and Chris Clifton. Privacy preserving naive bayes classifier on vertically partitioned data. In *SIAM International Conference on Data Mining*, 2004.

[234] R. Vilzmann, K. Kusume, C. Hartmann, and G. Bauch. A mac perspective on multiuser detection in ad hoc networks. In *Cross Layer Design, 2007. IWCLD '07. International Workshop on*, pages 109 –112, September 2007.

[235] T. K. Vintsyuk. Speech discrimination by dynamic programming. *Kibernetika*, 4:52–57, 1968.

[236] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 511–518, 2001.

[237] Xin Wang, Thomas DeMartini, Barney Bragg, M. Paravasivam, and Chris Barlas. The MPEG-21 Rights Expression Language and Rights Data Dicitionary. *IEEE Transactions on Multimedia*, 7(3):408–417, June 2005.

[238] Daniel J. Weitzner, Harold Abelson, Tim Berners-Lee, Chris Hanson, James A. Hendler, Lalana Kagal, Deborah L. McGuinness, Gerald Jay Sussman, and K. Krasnow Waterman. Transparent accountable data mining:

New strategies for privacy protection. In *AAAI Spring Symposium on The Semantic Web meets eGovernment*, 2006.

[239] Alan Westin. *Privacy and freedom.* The Bodley Head Ltd., 1970.

[240] B Widrow and ME Hoff. Adaptive Switching Circuits. In *IRE WESCON Convention Record*, New York, 1960. IRE.

[241] L. Wiskott, J. M. Kruger, and C. von der Malsburg. Face recognition by Elastic Bunch Graph Matching. *IEEE TPAMI*, 19(7):775 – 779, 1997.

[242] Lior Wolf, Tal Hassner, and Yaniv Taigman. Effective unconstrained face recognition by combining multiple descriptors and learned background statistics. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(10):1978–1990, 2011.

[243] Rebeca Wright and Zhiqiang Yang. Privacy preserving bayesian network structure computation on distributed heterogeneous data. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 713–718, Seattle, WA, USA, 2004. ACM Press.

[244] Rebecca N. Wright, Zhiqiang Yang, and Sheng Zhong. Distributed data mining protocols for privacy: A review of some recent results. In *Secure Mobile Ad-hoc Networks and Sensors*, volume 4074 of *LNCS*, pages 67–79. Springer, 2005.

[245] A. C. Yao. Protocols for secure computations. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, pages 160–164, 1982.

[246] Shen Yu and Qingyu Zhuang. The state complexities of some basic operations on regular languages. *Theoretical Computer Science*, 125:315–328, 1994.

[247] Xiaoyi Yu, Kenta Chinomi, Takashi Koshimizu, Naoko Nitta, Yoshimichi Ito, and Noboru Babaguchi. Privacy protecting visual processing for secure video surveillance. In *IEEE International Conference on Image Processing, ICIP'08*, pages 1672–1675, San Diego, CA, USA, 12-15 October 2008.

[248] J. R. Zeidler. Performance analysis of LMS adaptive prediction filters. *Proceedings of the IEEE*, 78:1781–1806, 1990.

[249] Jinfang Zhang, Z. Dziong, F. Gagnon, and M. Kadoch. Multiuser detection based mac design for ad hoc networks. *Wireless Communications, IEEE Transactions on*, 8(4):1836 –1846, April 2009.

[250] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face Recognition: A Literature Survey. *ACM Computing Surveys*, 35(4):399 – 458, 2003.