

CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD

Jordi Ribes González

ADVERTIMENT. L'accés als continguts d'aquesta tesi doctoral i la seva utilització ha de respectar els drets de la persona autora. Pot ser utilitzada per a consulta o estudi personal, així com en activitats o materials d'investigació i docència en els termes establerts a l'art. 32 del Text Refós de la Llei de Propietat Intel·lectual (RDL 1/1996). Per altres utilitzacions es requereix l'autorització prèvia i expressa de la persona autora. En qualsevol cas, en la utilització dels seus continguts caldrà indicar de forma clara el nom i cognoms de la persona autora i el títol de la tesi doctoral. No s'autoritza la seva reproducció o altres formes d'explotació efectuades amb finalitats de lucre ni la seva comunicació pública des d'un lloc aliè al servei TDX. Tampoc s'autoritza la presentació del seu contingut en una finestra o marc aliè a TDX (framing). Aquesta reserva de drets afecta tant als continguts de la tesi com als seus resums i índexs.

ADVERTENCIA. El acceso a los contenidos de esta tesis doctoral y su utilización debe respetar los derechos de la persona autora. Puede ser utilizada para consulta o estudio personal, así como en actividades o materiales de investigación y docencia en los términos establecidos en el art. 32 del Texto Refundido de la Ley de Propiedad Intelectual (RDL 1/1996). Para otros usos se requiere la autorización previa y expresa de la persona autora. En cualquier caso, en la utilización de sus contenidos se deberá indicar de forma clara el nombre y apellidos de la persona autora y el título de la tesis doctoral. No se autoriza su reproducción u otras formas de explotación efectuadas con fines lucrativos ni su comunicación pública desde un sitio ajeno al servicio TDR. Tampoco se autoriza la presentación de su contenido en una ventana o marco ajeno a TDR (framing). Esta reserva de derechos afecta tanto al contenido de la tesis como a sus resúmenes e índices.

WARNING. Access to the contents of this doctoral thesis and its use must respect the rights of the author. It can be used for reference or private study, as well as research and learning activities or materials in the terms established by the 32nd article of the Spanish Consolidated Copyright Act (RDL 1/1996). Express and previous authorization of the author is required for any other uses. In any case, when using its content, full name of the author and title of the thesis must be clearly indicated. Reproduction or other forms of for profit use or public communication from outside TDX service is not allowed. Presentation of its content in a window or frame external to TDX (framing) is not authorized either. These rights affect both the content of the thesis and its abstracts and indexes.



UNIVERSITAT
ROVIRA i VIRGILI

Cryptographic Techniques for Securing Data in the Cloud

JORDI RIBES-GONZÁLEZ

DOCTORAL THESIS
2018

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Jordi Ribes González

CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD

DOCTORAL THESIS

supervised by Dr. Oriol Farràs Ventura

Departament d'Enginyeria Informàtica i Matemàtiques



UNIVERSITAT ROVIRA i VIRGILI

Tarragona, Catalonia, Spain
August 2018

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

UNIVERSITAT ROVIRA i VIRGILI

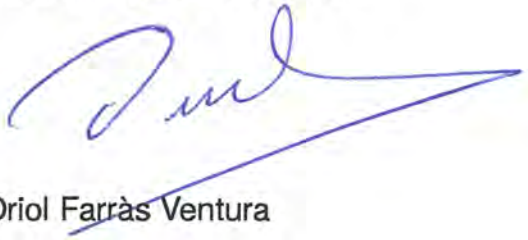
FAIG CONSTAR que aquest treball, titulat "Cryptographic Techniques for Securing Data in the Cloud", que presenta Jordi Ribes-González per a l'obtenció del títol de Doctor, ha estat realitzat sota la meva direcció al Departament d'Enginyeria Informàtica i Matemàtiques d'aquesta universitat.

HAGO CONSTAR que el presente trabajo, titulado "Cryptographic Techniques for Securing Data in the Cloud", que presenta Jordi Ribes-González para la obtención del título de Doctor, ha sido realizado bajo mi dirección en el Departamento de Ingeniería Informática y Matemáticas de esta universidad.

I STATE that the present study, entitled "Cryptographic Techniques for Securing Data in the Cloud", presented by Jordi Ribes-González for the award of the degree of Doctor, has been carried out under my supervision at the Department Computer Engineering and Mathematics of this university.

Tarragona, June 27, 2018

Doctoral Thesis Supervisor



Oriol Farràs Ventura

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Acknowledgements

First of all, I would like to express my sincere gratitude to my advisor Oriol Farràs for his continuous guidance, motivation and patience. On a personal level, it is hard to imagine having a better advisor than him. Whatever my appeal to applied mathematics (however that is defined!) has grown in these years is thanks to his influence, and for this I feel indebted to him.

Besides my advisor, I want to extend my deep gratitude to Josep Domingo-Ferrer and to David Sánchez for their support and for making my PhD possible in the first place.

I would also like to thank all the members of the CRISES research group at the Universitat Rovira i Virgili, with a special mention to Sara Ricci and Javier Parra-Arnau for the time spent discussing mathematics and everything else, to Manel Ruiz for his support, and to Jesús A. Manjón for his committed assistance.

I want to express my gratitude to Keith M. Martin for hosting me at the Information Security Group of the Royal Holloway University of London. It was a privilege to have the opportunity to benefit from the research environment at the Royal Holloway. My thanks also go to James Alderman and Benjamin Curtis for all the fruitful discussions during my visit and for providing the insight and expertise to carry out our joint work.

I am using this opportunity to thank my previous bachelor and master advisors Francesc Bars and Josep González. My sincere thanks also go to everyone who supported me throughout my career, including my friends and past colleagues Miquel, Martin, Mar, Albert, Miguel, Ferran, Ivan, Alfonso, David, Josep, Petr, César and the many more I am forgetting to mention. Thank you for being there.

I must also acknowledge the various organizations that partially supported this work, including the European Commission (through H2020-ICT-2014-1-644024 "CLARUS" and H2020-DS-2015-1-700540 "CANVAS"), the Government of Spain (through TIN2014-57364-C2-1-R "SmartGlacis" and TIN2016-80250-R "Sec-MCloud"), the Government of Catalonia (through Grant 2014 SGR 537) and the COST Association (through the COST Action IC1306).

Finally, last but foremost, I must express my gratitude to my partner Tamara, to my parents and grandparents Trini and Ramon, to my brother Roger, to my godfather and godson Ramon and to the rest of my family for providing me with unfailing support and encouragement. None of this would have been possible without them.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

UNIVERSITAT ROVIRA I VIRGILI

Abstract

Escola Tècnica Superior d'Enginyeria
Departament d'Enginyeria Informàtica i Matemàtiques

Doctor of Philosophy

by Jordi Ribes González

The cloud computing paradigm provides users with remote access to scalable and powerful infrastructures at a very low cost. While the adoption of cloud computing yields a wide array of benefits, the act of migrating to the cloud usually requires a high level of trust in the cloud service provider and introduces several security and privacy concerns.

This thesis aims at designing user-centered techniques to secure an outsourced data set in the cloud computing context. The studied problems and their treatment stem from the European Commission H2020 project *CLARUS: User-Centered Privacy and Security in the Cloud*. The explored techniques and problems are searching over outsourced data, outsourcing Kriging interpolation computations, secret sharing and data splitting.

Our first work on the subject of searching over outsourced data concerns symmetric searchable encryption (SSE) schemes, and develops techniques that enable secure and efficient two-dimensional range queries in SSE. Our second work tackles this problem through public key encryption with keyword search (PEKS) schemes, and builds efficient PEKS schemes achieving conjunctive and subset queries.

Our next aim is to securely outsource Kriging computations. Kriging is a spatial interpolation algorithm designed for geo-statistical applications. We present a method for the private outsourcing of Kriging interpolation based on homomorphic encryption.

Secret sharing is a fundamental primitive in cryptography, used in many cloud-oriented techniques. One of the most important efficiency measures in secret sharing is the optimal information ratio. Since computing the optimal information ratio of an access structure is generally hard, we find properties that facilitate its description.

Our final contribution concerns the privacy-preserving data splitting technique, which aims at protecting data privacy by storing different fragments of data at different locations. We analyze the data splitting problem from a combinatorial point of view, bounding the number of fragments and proposing various algorithms to split the data.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Contents

Acknowledgements **v**

Abstract **vii**

List of Figures **xiii**

List of Tables **xiii**

Abbreviations **xiv**

Outline of the Thesis **xvii**

1 Introduction **1**
1.1 Cryptography . . . 1
1.2 Cloud Computing . . . 3
1.2.1 CLARUS . . . 5
1.3 Research Directions . . . 8
1.3.1 Searching over Outsourced Data . . . 9
1.3.2 Computing on Outsourced Data . . . 11
1.3.3 Efficiency of Secret Sharing Schemes . . . 12
1.3.4 Privacy-Preserving Allocation of Data . . . 14
1.4 Research Contributions . . . 14
1.4.1 Range Queries in SSE . . . 15
1.4.2 Conjunctive and Subset PEKS . . . 16
1.4.3 Outsourcing Kriging Interpolation . . . 17
1.4.4 The Optimal Information Ratio of Secret Sharing Schemes . . . 18
1.4.5 Privacy-Preserving Data Splitting . . . 19
1.5 Publication List . . . 20

2 Background Theory **21**
2.1 Basic Definitions . . . 21
2.2 Symmetric-Key Encryption Schemes . . . 22
2.2.1 Provable Security . . . 22
2.2.1.1 Security Definitions . . . 23
2.2.1.2 Digression: Bilinear Groups . . . 25

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

2.2.1.3 Computational Hardness Assumptions . . . 26
2.2.1.4 Security Proofs . . . 28
2.3 Hash Functions and Pseudorandom Functions . . . 30
2.3.1 Pseudorandom Functions . . . 30
2.3.2 Hash Functions . . . 31
2.3.2.1 Hashing onto Elliptic Curves . . . 32
2.4 Public-Key Encryption Schemes . . . 32
2.4.1 Homomorphic Encryption . . . 34
2.5 Searchable Encryption . . . 35
2.5.1 Searchable Symmetric Encryption . . . 37
2.5.2 Public-Key Searchable Encryption with Keyword Search . . . 42
2.6 Secret Sharing . . . 45

3 Searchable Encryption 49
3.1 Two-Dimensional Range Queries Over Encrypted Data . . . 49
3.1.1 Introduction . . . 49
3.1.2 Problem Definition . . . 51
3.1.2.1 Searchable Symmetric Encryption . . . 51
3.1.3 Data Structure for Range Queries . . . 52
3.1.3.1 Data Model . . . 52
3.1.3.2 Binary Trees and Prefixes . . . 53
3.1.3.3 Quadtrees and Prefixes . . . 53
3.1.3.4 Prefix Decomposition . . . 54
3.1.3.5 Over-Covers . . . 54
3.1.4 Searchable Encryption for Two-Dimensional Data . . . 55
3.1.5 Prefix Decomposition Using Binary Trees . . . 56
3.1.6 Prefix Decomposition Using Quadtrees . . . 58
3.1.7 Over-Covers Using Binary Trees . . . 60
3.1.8 Over-Covers Using Quadtrees . . . 63
3.1.9 Security Analysis . . . 66
3.1.10 Efficiency Analysis . . . 68
3.1.10.1 Spatial Data . . . 68
3.1.10.2 Worst-Case Complexity Analysis . . . 68
3.1.11 Proof of Theorem 3.5 . . . 69
3.1.12 Conclusion . . . 72
3.2 PEKS with Conjunctive and Subset Keyword Search . . . 73
3.2.1 Introduction . . . 73
3.2.2 Preliminaries . . . 76
3.2.2.1 Notation . . . 77
3.2.2.2 General Model for PEKS Schemes . . . 77
3.2.2.3 Consistency Definition . . . 77
3.2.2.4 Hardness Assumptions . . . 78
3.2.2.5 Security Definition . . . 79
3.2.2.6 Implementation Remarks . . . 80
3.2.3 Conjunctive PEKS Scheme . . . 80
3.2.4 Consistency and Security Proofs for Conjunctive PEKS Scheme $\mathcal{S}_1$ 82
3.2.4.1 Consistency Proof for the Conjunctive PEKS Scheme $\mathcal{S}_1$ 82

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

3.2.4.2 Security Proof for the Conjunctive PEKS Scheme $\mathcal{S}_1$ . . . 83
3.2.5 Subset PEKS Scheme . . . 86
3.2.6 Consistency and Security Proofs for the Subset PEKS Scheme $\mathcal{S}_2$. 88
3.2.6.1 Consistency Proof for the Subset PEKS Scheme $\mathcal{S}_2$ . . . 88
3.2.6.2 Security Proof for the Subset PEKS Scheme $\mathcal{S}_2$ . . . 89
3.2.7 Efficiency Analysis . . . 91
3.2.7.1 Conjunctive PEKS Scheme . . . 92
3.2.7.2 Subset PEKS Scheme . . . 93
3.2.8 Conclusion . . . 94

4 Outsourcing Kriging Interpolation Computations 97
4.1 Introduction . . . 97
4.2 Kriging Interpolation . . . 99
4.2.1 Random Fields and Stationarity Assumptions . . . 100
4.2.2 The Kriging Prediction . . . 101
4.3 Private Outsourced Kriging Interpolation . . . 103
4.4 Our Techniques . . . 106
4.5 Our Construction . . . 108
4.6 Discussion . . . 110
4.7 Conclusion . . . 111

5 Local Bounds for the Optimal Information Ratio of Secret Sharing Schemes 113
5.1 Introduction . . . 113
5.2 Definition of Secret Sharing . . . 116
5.3 Preliminaries . . . 118
5.3.1 Some Families of Ideal Access Structures . . . 118
5.3.2 ANDs and ORs of Secret Sharing Schemes . . . 119
5.4 The Main Result . . . 121
5.4.1 The Lipschitz Constant of the Optimal Information Ratio . . . 123
5.5 Asymptotic Behavior of the Bound . . . 124
5.6 Other Constructions for Close Access Structures . . . 125
5.6.1 $(\mathcal{B}_1, \mathcal{B}_2)$-Coverings . . . 125
5.6.2 Secret Sharing Constructions Using Coverings . . . 127
5.6.3 A Construction Using Sunflowers . . . 128
5.7 Lower Bounds on the Information Ratio . . . 130
5.7.1 Proof of Theorem 5.23 . . . 131
5.8 Bounds for Linear Secret Sharing Schemes . . . 133
5.8.1 Razborov Rank Measure . . . 133
5.8.2 Critical Subfamilies . . . 135
5.9 Formulas for Monotone Boolean Functions . . . 136
5.9.1 Definitions . . . 136
5.9.2 Bounds on the Size of Formulas . . . 137
5.9.3 Submodular Formal Complexity Measures . . . 139
5.10 Conclusion . . . 140

6 Data Splitting 143

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

6.1 Introduction . . . 143
6.1.1 Our Results . . . 145
6.1.2 Related Work . . . 146
6.1.3 Outline of the Work . . . 148
6.2 A Combinatorial Approach . . . 148
6.2.1 Multi-Colorings of Hypergraphs . . . 151
6.2.2 Optimal Covers . . . 152
6.3 Algebraic Formulation of the Problem . . . 152
6.4 A Greedy Algorithm . . . 155
6.4.1 Our Construction . . . 156
6.4.2 An Heuristic Improvement . . . 157
6.5 Experimental Results . . . 159
6.5.1 Medical Data Example . . . 159
6.5.2 Performance Analysis over Random Graphs . . . 161
6.6 Conclusion . . . 162

Bibliography **165**

List of Figures

4.1 Timing costs of our private outsourced Kriging solution. 111

6.1 Two optimal solutions computed by the Gröbner basis method. 161

List of Tables

3.1 Size and false-positive rate values for binary over-covers. 63
3.2 Size and false-positive rate values for quadtree over-covers. 65
3.3 Worst-case efficiency comparison between 2-DRSSE schemes. 69
3.4 Size efficiency comparison of conjunctive PEKS schemes. 92
3.5 Time efficiency comparison of conjunctive PEKS schemes. 93
3.6 Performance analysis of conjunctive PEKS schemes. 93
3.7 Size efficiency comparison between subset PEKS schemes. 94
3.8 Time efficiency comparison between subset PEKS schemes. 94

4.1 Data protection offered by our private outsourced Kriging scheme. 106

6.1 Example of patient Healthcare features. 160
6.2 Comparison between the Gröbner basis method and Algorithm 7. 160
6.3 Time performance analysis of Algorithms 6 and 7. 161
6.4 Average percent size reduction given by Algorithm 7. 162
6.5 Average percent size increase given by Algorithm 7. 162

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Abbreviations

2-DRSSE	**2-D**imensional **R**ange **S**earchable **S**ymmetric-key **E**ncryption
3DES	**Triple D**ata **E**ncryption **A**lgorithm
ABE	**A**ttribute-**B**ased **E**ncryption
AES	**A**dvanced **E**ncryption **S**tandard
BDHI	**B**ilinear **D**iffie **H**ellman **I**nversion
BRGM	**B**ureau de **R**echerches **G**éologiques et **M**inières
CDH	**C**omputational **D**iffie-**H**ellman
CMAC	**C**ipher-Based **M**essage **A**uthentication **C**ode
CSP	**C**loud **S**ervice **P**rovider
DBDH	**D**ecisional **B**ilinear **D**iffie-**H**ellman
DDH	**D**ecisional **D**iffie-**H**ellman
DLP	**D**iscrete **L**ogarithm **P**roblem
EHR	**E**lectronic **H**ealth **R**ecords
FHE	**F**ully **H**omomorphic **E**ncryption
GDPR	EU **G**eneral **D**ata **P**rotection **R**egulation
GEUS	**Ge**ological **Su**rvey of Denmark and Greenland
HE	**H**omomorphic **E**ncryption
HIPAA	**H**ealth **I**nsurance **P**ortability and **A**ccountability **A**ct
HMAC	Keyed-**H**ash **M**essage **A**uthentication **C**ode
HVE	**H**idden-**V**ector **E**ncryption
ICD-10	**I**nternational Statistical **C**lassification of **D**iseases **10**th Revision
IND-CPA	**Ind**istinguishability Against **C**hosen **P**laintext **A**ttacks
IND-PCPA	**P**seudo-Randomness Against **C**hosen-**P**laintext **A**ttack
INSPIRE	**In**frastructure for **Sp**atial **Inf**or**m**ation in **E**urope
IP	**I**ntersection **P**attern

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

IT	**I**nformation **T**echnologies
LOINC	**L**ogical **O**bservation **I**dentifiers **N**ames and **C**odes
MPC	Secure **M**ulti-**P**arty **C**omputation
NP	**N**ondeterministic **P**olynomial time
OGC	**O**pen **G**eospatial **C**onsortium
OPE	**O**rder **P**reserving **E**ncryption
ORAM	**O**blivious **R**andom **A**ccess **M**achine
OXT	**O**blivious **C**ross-**T**ags
PEKS	**P**ublic-Key **E**ncryption with **K**eyword **S**earch
PHE	**P**artially **H**omomorphic **E**ncryption
PIR	**P**rivate **I**nformation **R**etrieval
PPT	**P**robabilistic **P**olynomial **T**ime
PRF	**P**seudo**R**andom **F**unction
PRP	**P**seudo**R**andom **P**ermutation
RAM	**R**andom **A**ccess **M**achine
ROM	**R**andom **O**racle **M**odel
RP	**R**esults **P**attern
SE	**S**earchable **E**ncryption
SEAL	**S**imple **E**ncrypted **A**rithmetic **L**ibrary
SHE	**S**omewhat **H**omomorphic **E**ncryption
SP	**S**ize **P**attern
SSE	**S**earchable **S**ymmetric-key **E**ncryption
WFS	**W**eb **F**eature **S**ervice Interface Standard

Outline of the Thesis

This thesis is organized in six chapters.

Chapter 1 states the thesis aims and objectives and frames it in the *CLARUS* project. It also outlines the directions taken in our research, and introduces the results contained in the remainder of the thesis.

Chapter 2 introduces some basic cryptographic preliminaries required throughout the remaining material. It briefly explains symmetric-key and public-key encryption schemes, provable security, hash and pseudorandom functions, and introduces homomorphic encryption schemes, searchable encryption schemes and secret sharing schemes.

Our results in the field of searchable encryption are presented in Chapter 3. This chapter is divided in two sections. Section 3.1 describes our approach to enable two-dimensional range queries over encrypted data, aimed at dealing with geo-referenced data sets. Section 3.2 describes two proposed PEKS schemes, which achieve conjunctive and subset queries with great efficiency.

Chapter 4 states our contributions to secure outsourced computation. We focus in outsourcing Kriging interpolation computations to the cloud, which applies to a practical use case where sensitive16 measurements are taken at certain geographic locations. We protect measurement data as well as interpolation results, and we additionally support architectures with multiple writers and a single reader.

Our contributions to the field of secret sharing are stated in Chapter 5. Our main contribution in this chapter states that *close* access structures admit secret sharing schemes with similar information ratio. We show that this property is also true for other families of secret sharing schemes and models of computation such as linear secret sharing schemes, span programs and Boolean formulas and circuits.

Chapter 6 presents our results in data splitting. We give a new combinatorial formulation to the data splitting problem, which consists in splitting sensitive data into several fragments. Through this formulation, we take two different approaches to solve the data splitting problem. The first one exploits algebraic techniques to compute all possible optimal decompositions. The second one sacrifices the solution optimality to achieve polynomial running time.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

To my family and friends.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Chapter 1

Introduction

In this chapter we motivate the directions taken in our research, and we briefly introduce each of the contributions in this thesis. This chapter is divided in five sections. In Section 1.1 we give a general account of cryptography. In Section 1.2 we expose the security and privacy concerns in cloud computing that motivate our research. We also present the *CLARUS* project, whose use cases and requirements guide all of our research. Next, in Section 1.3 we lay out the research directions taken in this thesis, explaining the topics we deal with and listing some of the applicable existing techniques. In Section 1.4 we give a general description of each of the contributions of this thesis. Finally, Section 1.5 lists the publications emanating from this thesis.

1.1 Cryptography

In recent years, the adoption of information technologies has seen a global and ceaseless growth, which in turn has boosted the relevance of IT security in society. Nowadays, securing information and digital systems is a priority for governments, businesses and users alike. We see cryptographic techniques as one of the strongest and most critical backbones in many of the technical measures taken to provide digital security in this context.

Cryptography is a discipline that studies techniques to transform information in order to achieve certain information security goals. Cryptography was originally born as the study of encryption schemes, which are algorithmic constructions whose main aim is to provide secure communication between two parties in the presence of an eavesdropper. Using encryption schemes, a user can protect a plaintext message at the source and send the resulting ciphertext over an insecure channel, preventing eavesdropping third parties to read the message in transit and preserving data confidentiality.

While encryption schemes are ubiquitous in all sorts of information systems, there exist other real-world security needs that can be addressed by cryptographic means. As a field, cryptography has evolved to consider a wide range of information security objectives, user architectures and functionalities. Real-world scenarios may require security objectives beyond data confidentiality, for example data integrity, message authentication and non-repudiation of signatures. Also, in practice users can take part in complex architectures, as in the case where messages are broadcasted to a wide audience and should only

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

reach authorized recipients. Finally, one may wish to carry out functionalities richer than simple communication, such as computing over encrypted data. Alternative factors, such as the exponential increases in computational power that quantum computing promises, have also driven the expansion of cryptographic research.

One of the main strengths of modern cryptography lies in its methodical treatment of security. In the cryptographic design process, real-world adversaries are modelled as abstract algorithms, and security goals are described through rigorous security definitions. Security definitions consist of two components: a threat model and a security guarantee. The threat model shapes the potential adversaries, specifying computational and functional restrictions to match the abilities of the adversaries in the actual scenario where attacks may take place. The security guarantee establishes the necessary conditions to assert that any given adversary breaks the security of the scheme. In this context, cryptographic constructions are often proven secure against security definitions. This approach allows to build very robust constructions that are secure according to the considered security guarantee, and that resist all adversaries that fit the threat model. We note that adversaries are allowed to take arbitrary strategies within the threat model constraints.

All this said, we must remind that cryptography is not the only security ingredient of information systems. In practice, cryptographic designs are implemented in software products and in hardware technologies, which in turn are connected to networks and embedded in complex societal systems involving users, companies, infrastructures, economics and laws. The security of all components in play must be taken into account in order to fulfill the guarantees that cryptography promises. Cryptography by itself, even when properly designed and implemented, does not suffice to provide security in practice.

This thesis is directed at studying cryptographic techniques to address data security and privacy issues raised in cloud computing. We seek for techniques set in the client-server model. The client-server model takes place between a client and a server, and the client starts the communication by requesting access to the resources of the server, who in turn provides a service to the client. In the setting we consider, the client uploads a protected data set to a cloud server and then wants to remotely execute some functionality over the outsourced data. The obtained solutions must be *user-centered*, meaning both the data protection and the storage of keying material must be carried out at the client trusted zone. They must also provide *end-to-end* security, meaning that the data must stay in a protected form whenever it leaves the client trusted zone. Finally, the sought techniques must efficiently support functionalities arising in cloud computing, as can be searching over the outsourced data, computing on the outsourced data or controlling access to the outsourced data.

A natural approach to enhance security in the client-server model is to encrypt the whole data set in a straightforward way, using traditional encryption schemes, prior to outsourcing it. However, this approach diminishes the benefits of cloud computing, since it forces users to retrieve the whole encrypted data set and locally decrypt it in case they want to make use of the data. To solve this issue, recent cryptographic research focuses on developing special encryption techniques that enable the secure delegation of particular functionalities over encrypted data to the cloud. Some of these functionalities are searching for data items which contain certain keywords, ordering data, computing statistics or restricting data access to particular users.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

In this thesis we seek techniques to efficiently delegate useful functionalities to the cloud while keeping the outsourced data set private. We approach three cryptographic techniques that fit this description: Searchable Encryption, Homomorphic Encryption and Attribute-Based Encryption. These techniques aim at delegating three of the main functionalities that are difficult to execute remotely in an efficient way when using traditional encryption schemes, namely searching, computing and controlling data access. Some of the obtained solutions, particularly for searchable and homomorphic encryption, have been implemented as privacy-preserving mechanisms in a cloud security toolkit [1].

Searchable encryption schemes are cryptographic schemes that enable searching over encrypted data. By using searchable encryption schemes, users can encrypt a data set in a specific way and upload it to the cloud, and still preserve remote searching capabilities. After the outsourcing phase, users can retrieve the segment of records in the outsourced data set that satisfy certain query conditions by sending an encrypted query to the cloud. All this process is carried out efficiently and in such a way that information leakage is reduced and data confidentiality is ensured. Since remote searching capabilities take a central role in the business logic of many commercial and industrial applications, searchable encryption schemes can provide huge security benefits in many applications.

Next, we approach homomorphic encryption schemes. An encryption scheme is homomorphic if it allows to encrypt data in a way that enables computing on the resulting ciphertexts. Given two ciphertexts generated with an homomorphic encryption scheme, it is possible to operate on them to produce a third ciphertext. This is done so that the performed operations translate to operations in the underlying plaintexts, thus allowing to effectively perform computations over encrypted data. Remotely performing computations is a major feature of cloud computing, and outsourcing computations on encrypted data can be very convenient for a wide range of applications.

Attribute-based encryption schemes are cryptographic encryption schemes that enable fine-grained access control to the encrypted data. Attribute-based encryption schemes are used concurrently by a set of users, and they typically rely on a central authority. In a setup stage, the central authority creates keying material and distributes secret keys to each user individually. Then, users are able to encrypt messages while designating a set of authorized recipients, without requiring any interaction with the central authority in the process. As a result, only authorized users are able to decrypt the resulting ciphertext, thus providing access control over the original message.

Additionally, in this thesis we explore data splitting, which is a non-cryptographic data anonymization tool. Data splitting is a privacy-preserving technique that consists in dividing sensitive tabular data into fragments, in such a way that individual fragments do not disclose any confidential information. Data privacy is enforced by allocating data fragments at different non-communicating servers. This approach is particularly suitable for multi-cloud architectures, where a single client operates with several independent cloud service providers.

1.2 Cloud Computing

The cloud computing paradigm enables the outsourcing of data storage and data computation to external servers, relieving storage and computational burden off of users. By using cloud computing services, users can benefit of top-of-the-line IT infrastructures

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

while saving on operational and maintenance costs. All this is achieved by providing users with remote access to a shared pool of computing systems, whose resources are managed to simultaneously meet all issued requests. The use of cloud technologies has seen tremendous growth over the past few years. Cloud computing brings economic and practical benefits such as ubiquitous access, scalable infrastructures and reduced costs. Over time, cloud adoption has led to a wide dissemination of computing capabilities and services, benefiting society at large.

Nowadays, many popular distributed applications take advantage of the cloud computing paradigm. Cloud infrastructures allow organizations to provide multiple users with remote access to services at an affordable price, sometimes even to a universal audience and at no cost to the end user. Some of the most frequently outsourced services are e-mail, conferencing, web mapping, file sharing, storage and computing solutions, and the deployment of other services in the cloud is expected to grow in the future. Leading cloud benefits reported by organizations mainly relate to a faster and cheaper access to high-end platforms, since cloud computing increases performance and lowers operational costs with respect to in-house computing infrastructures. Reported benefits also include data ubiquity and data availability, since cloud service providers often use data redundancy to achieve fault-tolerance and to prevent data loss.

Whereas migrating to the cloud can indeed bring great benefits to data owners and users, it also introduces several security and privacy concerns related to the fact that the cloud service provider has full access to the outsourced data. These concerns are considered to be the main barrier to the implementation of cloud strategies. Outsourcing sensitive data and computations to the cloud requires a high amount of trust in the cloud service provider and in the security measures it applies. By agreeing to the terms of service of cloud services, users frequently give up the ownership of their data and accept the usage of insufficient security measures. Still nowadays, many cloud services store data at rest in an unencrypted form, leaving it open to all kinds of attacks. Many recent data breach cases have been caused by poor implementation practices or by the adoption of inappropriate security measures. Furthermore, outsourced data is often sold to data brokers unknowingly to users, increasing the attack surface and exposing the likely sensitive data to potentially malicious actors.

This state of affairs is aggravated by the tremendous growth of cloud adoption. Indeed, recent data breaches [2, 3] include half a billion records leaked only in 2016, many of them containing personal details, credit card information, e-mail passwords, health records and full bank account details. Poor security measures and plain public release are among the main reported causes of data leakage. For a recent survey on the current state of privacy and security issues in cloud computing, see [4, 5].

Due to the current data breach outbreak, cloud technologies are often regarded as insecure by users with serious data privacy needs or with a privacy-aware mindset. The main perceived threats to data confidentiality in the outsourcing process are that the security mechanisms offered by cloud service providers are usually exclusively located within the cloud platform, and that the cloud service provider has full access to the possibly sensitive data. If users want to implement a cloud strategy, this situation compels them to either completely trust cloud service providers, agreeing with their privacy policies and relying on the security measures they may implement, or to take action and seek additional legal regulations and technical measures in order to guarantee the privacy of their data.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

On the legal side, and in an effort to mitigate this situation, several regulations have been put in place with the aim of giving control over data back to users and of enforcing the implementation of appropriate technical safeguards. Some examples of such policies are the recently approved E.U. General Data Protection Regulation (GDPR), which regulates the processing and movement of personal data of EU citizens, the Health Insurance Portability and Accountability Act (HIPAA), which regulates the use and disclosure of Protected Health Information in the U.S., and the E.C. INSPIRE directive, which defines a framework to enable the sharing of environmental spatial information among public sector organizations.

On the technical side, a reasonable approach to address security and privacy concerns in this setting is to apply additional privacy-preserving measures in the client side. Therefore, over the last few years there have been massive research efforts in order to design user-centered solutions that provide appropriate security guarantees to cloud users without trumping the performance and cost-saving benefits of cloud architectures. Data encryption techniques are among the candidate solutions to address privacy issues in this context. The application of these techniques would benefit both end users and cloud service providers, as enhancing trust in cloud storage would clearly foster the migration to cloud architectures.

1.2.1 CLARUS

This thesis is carried out in the frame of the European Commission project *CLARUS: User-Centered Privacy and Security in the Cloud*, funded under the EU Framework Programme for Research and Innovation Horizon 2020. The *CLARUS* project was launched on January 1st 2015, and ended in December 31st 2017. The *CLARUS* consortium was integrated by leading players in security development and research, including the Universitat Rovira i Virgili, the Royal Holloway University of London, Eurecom, the Katholieke Universiteit Leuven, Thales Services, AKKA Technologies, Trust-it, Montimage, Offis EV and the Fundació Clínic per a la Recerca Biomèdica. All solutions proposed in this thesis emanate from use cases, requirements and specifications suggested during the course of the project (see [1]).

The main aim of *CLARUS* is to develop a framework to assure secure storage and processing of data outsourced to the cloud, in the face of a not fully trusted cloud service provider. The *CLARUS* project seeks to develop a proxy that is able to interoperate with cloud servers through standard communication interfaces, and which provides cloud users with adequate security measures in this context. The proxy is assumed to be deployed on a local environment trusted by the cloud user, and the cloud user is thought to interact with the cloud through this proxy. The proxy protects confidential and sensitive data towards the cloud service provider, and it must operate transparently for the cloud user. The implemented privacy and security techniques must allow end users to securely store, exploit, monitor, audit and control their outsourced data while still gaining all the practical and cost-saving benefits that cloud services bring. Our objective in this thesis is to contribute to the pool of privacy-enhancing techniques suitable for *CLARUS*. Some of the techniques proposed in this thesis have been implemented in the *CLARUS* suite [1].

The *CLARUS* proxy is divided in four inter-communicating modules: the *CLARUS* access module, the access policy and key management module, the data operations

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

module and the monitoring and administration module. The *CLARUS* access module provides users with access to the *CLARUS* proxy deployed in their local premises. The access policy and key management module manages all the keying materials and data access policies. The data operations module implements the data protection techniques and applies them to input data. All the previous modules follow a plug-in approach, where supported protocols and techniques can be conveniently installed, uninstalled and configured. Finally, the monitoring and administration module implements intrusion-tolerant measures and is used to configure all other modules. The *CLARUS* solution has been implemented as an extensible application in the Java 8 platform, and code dependencies have been managed with Maven.

There exist a wealth of service suites alternative to *CLARUS* that share part of its objectives. The most well-recognized system is CryptDB [6], which aims at providing data confidentiality in SQL databases. Another similar project is MONOMI [7], which aims to improve CryptDB to allow more complex queries. Other similar tools implementing data encryption techniques for cloud environments are CipherCloud [8, 9], SecureCloud [10], PerspecSys [11], BoxCryptor [12] and CloudFogger [13]. There also exist several FP7 and H2020 projects aimed at data protection in the cloud, such as CloudSpaces [14], TRESSCA [15], ESCUDO-CLOUD [16] and MUSA [17]. One of the main differences between *CLARUS* and other existing projects lies in the considered techniques, since both cryptographic and data anonymization techniques have been implemented. Another key difference is the user-centered approach, because the proxy is considered to be deployed in a trusted zone owned by the user.

The threat model followed in *CLARUS* considers the cloud service providers as the adversaries, and defines both honest-but-curious and malicious adversaries. Honest-but-curious cloud service providers may passively eavesdrop on all the data made available to them while following the prescribed protocols, and malicious cloud service providers can both eavesdrop and perform active attacks by deviating from the protocols.

The architectural frames considered in *CLARUS* include multi-user and multi-cloud settings, i.e., they admit one or many users communicating with one or many cloud servers through a single dedicated proxy. In this setting, end users are classified according to their roles into three classes: security managers, data providers and data consumers. Security managers are in charge of configuring access control and technical aspects of the *CLARUS* proxy, managing security policies and selecting access protocols and security techniques according to user needs. Data providers may use the *CLARUS* proxy to securely store data in the cloud. Data consumers may want to interact with the cloud infrastructure to access the data or to perform some action on it, having previously received authorization from the corresponding data providers.

One of the functionalities to be supported by *CLARUS* is the ability to create and store data securely. Data retrieval must also be implemented, and enabled queries must include searching by id and searching according to other identified complex queries. Support for dynamic data sets is a requisite as well, and thus updating data items by id and by other attributes is required. Finally, performing some identified computations over the outsourced data is also a requirement. In summary, the considered use cases demand two main functionalities to be delegated to the cloud: searching over outsourced data and computing on outsourced data.

Two business application cases of interest have been contemplated in the *CLARUS* project, one dealing with e-health data sets and the other dealing with geo-referenced

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

data sets. The user and technical requirements for the *CLARUS* solution have been motivated by these use cases.

In the e-health scenario, we study the outsourcing of electronic health records (EHRs), which for instance can consist of result reports, clinical notes, diagnostics (ICD-10) and lab results (LOINC). The typical volume of medical data sets and the need for data ubiquity and availability make the cloud infrastructure an appealing tool for hospital information systems. However, the sensitive and highly personal nature of medical data makes outsourcing a high-risk operation if no proper security guarantees are taken. In this respect, several legislative measures stipulate that medical data must be adequately protected.

In the cloud computing setting, data consumers may want to access patient data in order to retrieve a segment of the medical data according to a set of attributes, such as the id of a certain patient or the ICD-10 number corresponding to a certain disease. Medical data consumers may need to update or remove data as well. In turn, data owners may want to control which data consumers are allowed to access any given document, for instance to restrict access to patient history to the authorized personnel. Performing statistical computations on the outsourced data, such as means, standard deviations and frequencies, may also be of interest. We seek to apply suitable cryptographic techniques that adequately protect the outsourced medical data while enabling the required functionalities to be carried out over the protected data.

The requirements considered in this domain stem from the use cases proposed by the biomedical research-oriented *CLARUS* partner Fundació Clínic per a la Recerca Biomèdica. The requirements here mainly focus in single user architectures, where a single actor both owns and manages the outsourced sensitive information. This actor can be thought of as the medical institution in charge of the EHRs of the patients. Functionalities to support include creating, updating and removing medical histories from a single patient (i.e., by id), and retrieving the history of a single patient. Shared access to the EHRs according to access policies specified by the data providers is also desirable.

In the geo-referenced data case, we study the outsourcing of confidential data containing locations and measurements of critical natural earth resources as water boreholes, minerals or rare earth materials. Geo-referenced data may be judged sensitive, for instance because of its high business potential or its private nature.

In the cloud storage setting, data consumers may want to access the data in order to retrieve a segment of it. In this case the issued queries can take specially complex forms, for instance when they refer to particular geographic areas. Data consumers may also want to remotely perform geo-statistical computations on dynamic data sets. Among such computations we highlight the Kriging interpolation algorithm [18–21], which is of particular relevance in this context. We seek for cryptographic techniques that protect the outsourced data while enabling meaningful geographic queries and the private evaluation of geo-statistical computations.

The considered requirements have their source in industrial users dealing with groundwater borehole data in France (maintained by the BRGM [22]) and borehole data from the oil industry in Denmark and Greendland (maintained by GEUS [23]), and fall into the themes and topics specified in the INSPIRE directive. In order to comply with this directive, we must take into account data provided by multiple users and distributed

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

among multiple cloud servers. Main functionalities to support in this setting are creating, updating and removing outsourced spatial data, retrieving data that lies in some specified geographical location and remotely performing specific computations over the outsourced data. In the spirit of the INSPIRE directive, shared access to spatial data according to access policies specified by the data providers is also desirable.

1.3 Research Directions

In this section we survey the different research themes explored during the course of the thesis. The taken research directions are clearly motivated by the *CLARUS* use cases stated above, and can be broadly divided into four topics: searching over outsourced data, computing over outsourced data, the efficiency of secret sharing schemes and privacy-preserving allocation of data.

Searching over outsourced data refers to the act of retrieving the segment of records of an outsourced data set that satisfy certain query conditions. It can be argued that searching is one of the most basic operations that can be carried out on outsourced data. Since remote searching capabilities take a central role in the business logic of many commercial and industrial applications, solutions to search securely over outsourced data can provide huge security benefits. Achieving secure search over encrypted data may help in solving security issues related to storage and database servers, enhance security in e-mail routing, or even aid in overcoming legal difficulties due to data extradition treaties.

Computing over outsourced data is also a central functionality to cloud computing. Due to the major performance advantage and the data ubiquity and availability features provided by cloud infrastructures, remotely performing computations can be very convenient for a wide range of applications. Such applications include for instance computing statistical indicators, accounting and financial applications, e-voting or machine learning.

Secret sharing is a fundamental cryptographic primitive that allows the distribution of a secret amongst parties, in such a way that the distributed shares do not hold any information on the secret. By combining together some of the shares, parties are able to reconstruct the secret. Secret sharing has found applications in many cloud-oriented cryptographic techniques, the most relevant to our setting being attribute-based encryption. Attribute-based encryption schemes [24] can be built by embedding secret sharing into public-key constructions. The efficiency of secret sharing as a primitive has a considerable impact on the storage and computational cost of the resulting attribute-based encryption schemes, and thus we seek for properties that help in analyzing and quantifying this efficiency.

Privacy-preserving allocation of data refers to storing fragments of a data set at different locations to enhance privacy. In order to guarantee data availability and integrity, cloud users often adopt a multi-cloud approach in their cloud strategy. Data splitting techniques are implemented in multi-cloud architectures, and they are supported in a variety of database management systems (such as SQL or MongoDB) in the form of database shards and partitions, where they provide flexibility and data redundancy. Data splitting can also be adopted to mitigate data privacy concerns, and we explore the efficiency and the feasibility of this privacy-preserving method.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

This section is divided into four parts, corresponding to each of the four different topics laid out in the preceding paragraphs. After briefly introducing each topic and the security concerns behind it, we give an overview of existing solutions that tackle the exposed security and privacy issues, and we briefly comment on the particular direction taken in our research.

1.3.1 Searching over Outsourced Data

Retrieving outsourced data selectively is a basic functionality of cloud storage servers. Searching over outsourced data is a fundamental feature required in a wealth of applications, such as file storage systems, outsourced backup services, accounting systems or e-mail servers. Such applications benefit greatly from the economic and operational advantages that cloud computing brings. However, entrusting cloud service providers with the data set, with the query and with the search execution encompasses many privacy threats, since it compels users to completely trust cloud service providers on their privacy policy and on the security measures they may implement.

To address the aforementioned issues we consider user-centered security measures, in which information is protected at the user's trusted zone prior to outsourcing it. In this respect, the security techniques to be considered in the context of searching over outsourced data may vary mainly according to two factors: the security aims and the architectural setting.

As for the security aims to be considered, we can classify potentially sensitive information in this context into three classes: outsourced data, query data and search information. Outsourced data refers to information enclosed in the data set owned by the user, such as the data content or the number of data items. In case users delegate data of sensitive nature, as in the use cases posed in *CLARUS*, data items may need to be protected. Query data refers to data enclosed in queries. Queries themselves can be of a sensitive nature, for instance if they contain personal information details, which may reveal information on the outsourced data. Among the query data, we highlight the search pattern. The *search pattern* is the information of whether any two given queries are identical or not. Finally, search information fits all the data generated in the searching process, such as the number of data items that match a given query. Among the search information, we highlight the *access pattern*, which is the knowledge of which outsourced data items match any given query.

The architectural frame where the searching process takes place involves three distinct types of actors: writers, readers and the cloud service provider. Writers are cloud users in charge of outsourcing the data (i.e., data providers), and readers hold the ability to remotely search over the outsourced data (i.e., data consumers). The architectural setting can be single or multi-cloud depending on the number of involved cloud service providers. Likewise, we depict four possible architectures according to cloud users:

- Single user: This is the simplest context, where a single user outsources a data set and wants to query over it remotely. An example application could be a file storage service without sharing capabilities.
- Single reader, multiple writers: In this setting, multiple users push searchable data to the cloud so that a single user is able to search over the outsourced data. This may be the case in e-mail or sensor network applications.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

- Multiple readers, single writer: In this context, a single user pushes data to the cloud so that a set of user-defined readers can search over the outsourced data. As an example, some single-writer electronic message boards or file storage servers can fit this setting.
- Multiple readers and writers: This is the most complex setting, where each of a set of users is able create searchable content for a chosen set of users. This could be the case in a file storage service with sharing capabilities.

Other considerations can also apply, such as computational limitations, whether the data set is to remain static throughout the whole searching process, or if dynamic data sets are allowed.

In this thesis we analyze the cryptographic approach to solve the data confidentiality issues related to searching over outsourced data. The principal techniques considered to this aim are Searchable Encryption (SE), Oblivious RAM (ORAM) and Private Information Retrieval (PIR). In this thesis we provide solutions based on SE, which is arguably the most practical technique for secure delegated search. Although not as relevant to the searching case, other techniques such as Order-Preserving Encryption (OPE) and Public-Key Encryption with Equality Test are also useful in some applications involving secure search.

Searchable Encryption was first instantiated by Song, Wagner and Perrig [25] in 2000. SE schemes aim at encrypting the data contents while enabling secure search over the outsourced data. Typically, SE schemes leak the search and access patterns in order to achieve greater efficiency, and protect the remaining query and search data. While this security and efficiency trade-off has been shown to induce leakage-abuse attacks [26–32] in various cases, SE is the only cryptographic solution known to date that allows clients to remotely perform queries on an outsourced encrypted data set with near-optimal search efficiency. Moreover, most SE schemes require very low to no interactivity, constant client computational complexity and linear to sublinear search time.

Oblivious RAM was first formulated by Goldreich in 1987 [33]. ORAM achieves the highest security features and protects all data contents and the search and access patterns. However, the practical applicability of ORAM techniques is restricted due to technical reasons, and most instances require a great amount of interactivity and client computations. For example, it has been shown in [34] that, if n documents are to be stored and if server space is $O(n)$, then ORAM imposes an $\Omega(n)$ communication overhead.

Private Information Retrieval was introduced by Chor, Goldreich, Kushilevitz and Sudan in 1995 [35]. PIR aims at protecting the access pattern, and does not protect the outsourced data. Most PIR schemes force at least linear computational complexity $\Omega(n)$ onto the search operation (with notable exceptions, such as [36]). This efficiency obstacle often makes PIR unusable when large data sets are involved.

Recently, intensive research efforts in the field of SE have provided schemes that support all the architectures proposed above. Much like traditional encryption schemes, searchable encryption schemes come in two very different types: symmetric-key and public-key. These correspond to the schemes in the single-user setting and to schemes in the single reader, multiple writer setting.

The symmetric-key type of SE, namely Searchable Symmetric Encryption (SSE), was principally developed by Curtmola, Garay, Kamara and Ostrovsky in 2006. In [37],

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

they introduce efficient constructions and rigorous security definitions for searchable encryption. Since their foundational work, the field has expanded in many directions, by enhancing the security and efficiency of the schemes, by improving expressiveness of the queries, by dealing with multi-user settings and by supporting various functionalities.

The public-key type, named Public-Key Encryption with Keyword Search (PEKS), was firstly proposed by Boneh, Di Crescenzo, Ostrovsky and Persiano [38] in 2004. Since their pioneering work, there have appeared several PEKS schemes in the literature, improving the scheme in [38] in terms of efficiency, security or functionality. Recent improvements achieve sublinear search [39], reduction of communication and storage costs [40, 41], extension to multi-user systems [42] or various security improvements [43–48].

1.3.2 Computing on Outsourced Data

Computing over outsourced data is a central functionality to cloud computing. The wide range of advantages provided by cloud infrastructures, such as computational performance, data ubiquity and data availability, make data outsourcing an attractive solution for applications involving computational tasks. Examples of such applications are accounting and financial systems, applications involving the computation of statistical indicators, decision support systems, e-voting systems or machine learning applications. Some of these applications may require to perform computationally expensive operations and to potentially deal with complex architectures and dynamic data sets.

However, outsourcing data and computations in the face of an untrusted cloud service provider poses many privacy concerns. In this context, we can classify potentially sensitive information into three classes: outsourced data, outsourced computation and computation outcome. Outsourced data refers to information enclosed in the data set owned by the user, just as in the case of searching over outsourced data posed in the previous section. Outsourced computation refers to information about the particular operations to carry out. For example, whether the outsourced computational task is to compute the mean of the outsourced data set, or to perform a linear regression. Computation outcome refers to information about the computation results, which can reveal information both about the outsourced data and about the outsourced computation.

The architectural frame where computing over outsourced data takes place involves three distinct types of actors. The first is the cloud service provider. Regarding cloud users, we make a distinction between writers (or data providers) and readers. Writers provide the data to be outsourced, and readers specify the computation to be outsourced and receive the computation outcome. Regarding writers and readers, we contemplate the same four possible architectures as in the context of searching over outsourced data, depending on whether there is a single or multiple readers and writers. Moreover, the architectural setting can be single or multi-cloud depending on the number of cloud service providers involved.

In order to address data privacy and confidentiality concerns when computing over outsourced data, the data privacy literature has developed numerous solutions. Here, we outline the cryptographic techniques aimed at solving these issues. There exist two main areas of cryptography that deal with data confidentiality issues in outsourced computation: one based on Homomorphic Encryption (HE) and one based on Secure Multi-Party Computation (MPC).

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Homomorphic encryption schemes are cryptographic schemes that enable computing over encrypted data, and in this thesis we use public-key encryption schemes with homomorphic properties to this end. Homomorphic encryption schemes are set in the single-reader architecture, and they enable users to encrypt their data in such a way that operations carried out with the generated ciphertexts translate to operations on the underlying plaintexts. By outsourcing the data in an encrypted form, users can request computations to be performed on the protected data and then receive an encrypted outcome. Therefore, this usage of HE schemes protects both the outsourced data contents and the computation outcome.

We can classify homomorphic encryption schemes according to the operations that they support over encrypted data. The cryptographic literature usually distinguishes three classes:

- Partially homomorphic encryption (PHE): PHE schemes support a single operation on ciphertext, either multiplication or addition.
- Somewhat homomorphic encryption (SHE): SHE schemes support both addition and multiplication of ciphertexts, but they support a limited number of operations.
- Fully homomorphic encryption (FHE): FHE schemes support an unlimited number of additions and multiplications on ciphertexts, thus enabling arbitrary computations over encrypted data.

While FHE schemes are very convenient from the functionality side, the solutions known up to date are computationally expensive when evaluating functions with a high multiplicative depth. This, paired up with the current technological status, restricts the applicability of FHE in practice. Therefore, in our research we turn to PHE to tackle some of the concerns posed in the *CLARUS* use cases. In particular, in Chapter 4 we propose a construction to outsource data processing to the cloud by combining PHE with a tailored modification of the computations.

Secure multi-party computation [49] deals with the general setting of multiple writers and readers, and aims to protect only the individual inputs from users. MPC allows multiple users to jointly evaluate a function on their private inputs. By agreeing on a particular function and by engaging in an elaborate protocol, MPC allows users to learn the joint function evaluated on the inputs of all users, while they learn nothing else about the other user's inputs in the process.

1.3.3 Efficiency of Secret Sharing Schemes

Secret sharing is a fundamental primitive in cryptography, and it is an essential building block for many other cryptographic applications. Some of these applications are of interest in the cloud computing setting posed in *CLARUS*, and most notable examples are attribute-based encryption [24], secure multi-party computation [49], e-voting and joint signatures.

Secret sharing schemes aim at protecting a secret piece of information by dividing it into shares. This is done so that it is possible to reconstruct the secret value from certain specified combinations of shares. The use of secret sharing schemes guarantees that it

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

is hard to recover the secret from individual shares, and that shares must be combined in order to recover the secret. In *information-theoretically* (also called *unconditionally*) secure schemes, security does not rely on any computational assumption, and the resulting individual shares do not hold any information on the secret value when considered in isolation.

Typically, secret sharing schemes are used in *distribution schemes*, where they unfold in a frame involving various *participants*. In this frame, secret sharing prevents both the disclosure and the loss of secrets. A special participant, called the *dealer*, initially holds a *secret value*. Using secret sharing, the dealer can divide the secret into *shares* and distribute them to other participants, in such a way that only specific subsets of participants can recover the secret by pooling their shares together. In this context, the main proposed application of secret sharing schemes consists in protecting both the integrity and the confidentiality of important information, such as numbered bank accounts or encryption keys, by letting multiple participants safeguard it. Nevertheless, as a cryptographic primitive, secret sharing has found many other uses.

Subsets of participants are *authorized* if they are able to recover the secret, and *forbidden* if their shares hold no information on the secret. The family of authorized subsets of participants is called the *access structure* of the scheme. If every subset of participants is either authorized or forbidden, we say that the scheme is *perfect*. In a perfect scheme, the size of each share is at least the size of the secret, and a scheme is called *ideal* if every share is exactly as large as the secret. In our research we only consider perfect and information-theoretically secure secret sharing schemes.

The first secret sharing schemes were proposed by Shamir [50] and Blakley [51] in 1979. By using polynomial interpolation techniques, they achieve perfect and ideal schemes that support any threshold access structure, meaning that the only authorized sets of participants are those larger than a certain threshold. In subsequent works, Ito, Saito and Nishizeki [52] and Benaloh and Leichter [53] presented perfect schemes supporting any monotone increasing access structure. Nevertheless, for almost all access structures their schemes are not ideal, and the size of the shares increases exponentially in the number of participants. The size of the shares is an important property of secret sharing schemes, and many applications require small-sized shares.

The main measure of the efficiency of a secret sharing scheme is the *information ratio*. The information ratio of a secret sharing scheme is defined as the ratio of the maximum length in bits of a share generated with the scheme to the bit-length of the secret value. While this efficiency measure is generally easy to compute and very useful for evaluating the efficiency of explicit secret sharing schemes, in practice a secret sharing scheme is instantiated from a given access structure of interest. Therefore, we may be more interested in measuring the well-conditioning of an access structure, and to do so we turn to the *optimal information ratio*. The optimal information ratio of an access structure is defined as the infimum of the information ratios of all the perfect secret sharing schemes realizing the access structure. Thus, it indicates the size of the shares required to realize a given access structure, giving a tight lower bound on the information ratio of schemes realizing a particular access structure.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

1.3.4 Privacy-Preserving Allocation of Data

Due to the rising use of relational databases, nowadays many applications handle data in a tabular format. Tabular data consists of *records* structured according to a set of *attributes*, where each record typically holds up to one value per attribute. In this context, outsourcing services that handle tabular data to the cloud may bring many operational and economic advantages in a wealth of applications. However, in the case of outsourcing tabular data, some combinations of attributes in the data set may constitute sensitive information. For instance, in a medical data set case, a couple of attributes holding the passport number and disease diagnostic information can be jointly regarded as personally identifiable and sensitive information. Still, other attribute combinations, such as blood pressure metrics and disease diagnostic information, may not violate privacy.

Privacy-preserving allocation of data refers to storing fragments of the data at different locations to enhance data privacy. In a cloud computing setting, one can take advantage of *multi-cloud* architectures to provide data privacy. The term multi-cloud expresses the concurrent use of multiple cloud services in a single organization or application. Multi-cloud architectures are widely adopted in practice, despite the inevitable increase in economic cost they entail. A suitable strategy to enhance data privacy in the data outsourcing context is to leverage multi-cloud architectures, and to store fragments of the data at different cloud servers.

There exist a variety of tools aimed at enhancing data privacy in the described setting. While we have already briefly surveyed some of the tools on the cryptographic side, some of them are non-cryptographic, and include for instance data sanitization, k-anonymity, file splitting, data obfuscation, data merging and data splitting. In this thesis we deal with privacy-preserving data splitting.

Privacy-preserving data splitting was first introduced by Aggarwal et al. [54] in 2005. Several subsequent works have improved on their construction in various directions [55–58]. Privacy-preserving data splitting deals with the problem of sensitive attribute combinations by realizing privacy-preserving allocation of data. The objective of privacy-preserving data splitting is to find a decomposition into fragments of a given tabular data set and to distribute the fragments among several servers, so that no one server holds simultaneous information on any sensitive attribute combination. In this way, assuming that servers do not communicate with each other, data splitting attempts to minimize the leakage of information that can be extracted by each individual server.

In addition to the *privacy constraints* determined by the sensitive attribute combinations that must not be stored together in any fragment, the data splitting literature also considers *processing constraints*. These constraints specify combinations of attributes that must be jointly stored in some server, be it because they must be queried on more efficiently or because complex computations are to be remotely performed on them.

1.4 Research Contributions

In this section we briefly describe each of the contributions of this thesis. In the order of appearance in the remainder of the text, the research contributions are: Range Queries

in SSE, Conjunctive and Subset PEKS, Outsourcing Kriging Interpolation, the Optimal Information Ratio of Secret Sharing Schemes and Privacy-Preserving Data Splitting.

1.4.1 Range Queries in SSE

The *CLARUS* geo-referenced data use case poses the problem of searching over a sensitive spatial geo-referenced data set outsourced to the cloud. We contribute to solve this problem by designing a SSE scheme that handles data sets whose items are referenced by two-dimensional coordinates. This can be the case of data containing information about civil constructions and infrastructures, or about the location and measurements of critical natural earth resources such as water boreholes, minerals and rare earth materials.

Since the outsourced information can be valuable and confidential, users may want to protect it against the cloud server and any external attackers. At the same time, users may wish to securely search over the protected outsourced data, preserving the ability to retrieve the data items that are located at a certain chosen region.

In our work we take a cryptographic approach to this problem. The existing cryptographic literature tackles the problem of efficiently executing range queries over encrypted data by applying Deterministic Encryption [6, 59–62], Order-Preserving Encryption [6, 63, 64] and Searchable Encryption [64–72] schemes. Since solutions that apply SSE provide the best overall trade-off in efficiency and security among all the options available up-to-date, we aim at applying SSE to protect the outsourced data. Our contribution to SSE is to study how to remotely perform queries on an encrypted two-dimensional data set. We focus on the two-dimensional case, in which data items can be seen as points in a planar region. In this case, range queries match all features in the data set which intersect a chosen rectangle in the planar region. Some of our approaches can be applied to multi-dimensional range queries as well.

Searchable Symmetric Encryption [37] schemes have been extended in recent works to achieve secure and efficient range queries [64, 67–69, 71, 72]. These works typically exploit a special indexation of documents, expanding both query and data coordinates and making black-box use of a searchable encryption scheme on the modified coordinates. We improve on the approach of [68], which expands coordinates according to a binary tree structure in order to enable efficient two-dimensional range queries over encrypted data. Due to the considered use case requirements, we set our solutions in the single-user architecture.

Achieving two-dimensional range queries in the fashion described in [68] has an impact on data leakage when issuing queries, and also on computational and communication costs. In our work we take a step forward in addressing both concerns by introducing *quadtrees* and by applying *over-covers*.

It is worth noting that the cryptanalysis of SE schemes has been recently studied in various works [26–32], which develop attacks on SE schemes by exploiting the security models and the leakage associated to various SE schemes. Among these, we highlight some very recent attacks oriented to range searchable symmetric encryption [30]. These attacks work under the assumptions that the data set is dense enough and that a sufficient amount of uniformly distributed queries are issued, and they exploit the knowledge of the access pattern (and possibly some rank information) to expose all the queried and

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

database values to an external attacker. To the best of our knowledge, up to date no SE schemes that avoid this and similar attacks have been proposed, and the solutions proposed in Chapter 3.1 are also vulnerable if the premises stated in [30] hold.

The use of over-covers was introduced by Faber et al. [68], in the setting of one-dimensional range searchable encryption. Using over-covers, the user queries for ranges that can be larger than the desired ones, thus potentially inducing false positives in the search results. However, the ranges are chosen in such a way that information leakage and communication costs are reduced.

We analyze the use of over-covers, generalizing the constructions by Faber et al. [68] to two-dimensional range queries. We also extend the result in [68] to reduce the false-positive rate by providing larger-sized over-covers. In this way, it is possible to substantially rise accuracy while increasing the communication costs.

We also propose the use of *quadtrees*, introducing a new indexing technique for two-dimensional range queries in searchable encryption that reduces information leakage. Quadtrees induce a decomposition of the two-dimensional grid into quadrants. This allows to improve security by reducing leakage with respect to alternative constructions [66, 68] for two-dimensional data. This technique can be extended to multi-dimensional range queries as well.

Since using quadtrees increases the computational and communication costs, we mitigate this downside by proposing the mixed use of quadtrees and over-covers. For all the proposed techniques, we analyze the trade-off between efficiency, security and accuracy.

1.4.2 Conjunctive and Subset PEKS

Supporting secure searching capabilities in multi-user architectures is a key requirement in many cloud computing applications, including, for instance, audit logs, e-mail gateways, distributed sensor networks or cloud storage with sharing capabilities. Public Key Encryption with Keyword Search (PEKS) schemes are searchable encryption schemes set up in the single reader, multiple writer architecture. Our main contribution to the field of PEKS is the design of two PEKS schemes with extended functionalities, which achieve high efficiency marks and support conjunctive and subset queries respectively.

To support an asymmetric architecture, PEKS schemes generate two kinds of keys: public keys and secret keys. Data providers hold the public key, and a single reader client holds the secret key. By using the public key, data providers are able to generate searchable encrypted data, while only the client is able to generate queries for this encrypted data by using the secret key. In the case of PEKS, searching over an encrypted data set usually takes linear time in the number of data items in the data set (except in the case of [39], where it takes linear time in the number of data providers). Hence, supporting this asymmetric architecture comes at the cost of sequential search.

The earliest PEKS scheme was presented by Boneh et al. [38] in 2004. Since their pioneering work there have appeared several other PEKS schemes in the literature, improving [38] in terms of efficiency, security and expressiveness. In our research we follow this trail by exploring expressiveness and efficiency enhancements of PEKS, and we accordingly describe *conjunctive* and *subset PEKS* schemes.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Conjunctive PEKS schemes [42, 65, 73, 74], which enable *conjunctive field keyword* queries, are one of the most common enhancements of PEKS. The first conjunctive PEKS scheme was presented by Park et al. [74] in 2004. Typically, in conjunctive PEKS data providers use the public key of the client to send to the cloud server a ciphertext encrypting a tuple of keywords $(w_1, \ldots, w_m)$. The client can then use its secret key to query the cloud server with a tuple of keywords $(w'_1, \ldots, w'_l)$, along with a set of positions $\{j_1, \ldots, j_l\} \subseteq \{1, \ldots, m\}$. When receiving this query, the cloud server is able to learn if the predicate $\bigwedge_{i=1}^{l}(w_{j_i} = w'_i)$ holds, and nothing else is disclosed about the ciphertext.

Subset PEKS [65] is another enhancement of PEKS which enables *subset* queries. In subset PEKS, data providers use the public key of the client to encrypt a tuple of keywords $(w_1, \ldots, w_m)$. The client can then produce a query associated to m arbitrary sets of keywords $(A_1, \ldots, A_m)$. When receiving this query, the storage server can check whether the conjunctive subset query predicate $\bigwedge_{i=1}^{m}(w_i \in A_i)$ holds or not, without learning anything else from the ciphertext.

As in all previous works, our schemes are pairing-based constructions. Our first proposed scheme supports conjunctive queries. The proposed scheme can be seen as an analog to Boneh et al.'s scheme [38] by replacing the underlying hardness assumption by a stronger one. This modification allows one to take advantage of the bilinearity of pairings and build a conjunctive PEKS scheme with very small ciphertexts, with an efficient search process and with very small queries, which consist of a single group element. Our second proposed scheme supports subset queries and some more general predicates. We show that it improves previous related schemes in terms of efficiency and expressiveness. Moreover, unlike previous related schemes [65], our subset PEKS scheme admits an arbitrary keyword space.

1.4.3 Outsourcing Kriging Interpolation

In this thesis, we present a method for the private outsourcing of *Kriging interpolation* using a tailored modification of the Kriging algorithm in combination with homomorphic encryption. Our solution allows the client to outsource Kriging computations while hiding crucial information from the cloud service provider.

The *CLARUS* geo-referenced data use case poses the problem of privately outsourcing Kriging computations. Kriging has been identified as a good candidate process to be outsourced, based on the practical and legislative requirements of industrial users (for instance, [1, 75]). In this thesis we present a practical, efficient and secure solution to privately outsource Kriging interpolation.

Kriging interpolation [18–21] is a well-recognized form of linear interpolation technique designed with geo-statistical applications in mind, used to analyze all kinds of spatially dependent phenomena. The Kriging algorithm is used to predict the value of a previously sampled phenomena at an unobserved location in a two-dimensional region. In this case, Kriging builds a prediction as a weighted sum of prior measurements, where measurements taken close to the unobserved location are given a greater weight than those far away.

Consider a client that owns a *Kriging data set* (i.e. a set of measurements taken at various locations), and suppose that it wishes to outsource this data set to an honest-but-curious cloud service provider to take advantage of the various cloud benefits. Further,

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

other data generating nodes may be authorized by the client to add and remove data from the outsourced data set. The client would like to make use of both the storage and computational power of the server to make a Kriging service on its data set available to multiple users. Since the outsourced data set may contain sensitive information, the client may need to protect the Kriging data set prior to outsourcing it.

Our solution to this problem uses additive homomorphic encryption to outsource Kriging interpolation efficiently. We show that the Kriging process can be adapted so that all sensitive information can be either encrypted or factored out of the computation. In this way, the Kriging computation may be performed on encrypted measurement values, by using an additively homomorphic encryption scheme. We make a trade-off by protecting only the information relating to measurement values in the data set, and we do not hide locations of prior measurements and of queries. This choice is justified in cases where measurement locations are externally observable (e.g. if measurements come from previous mining operations).

Cryptographically-secured Kriging was previously studied in a different setting, where a *server* owns a data set and clients may query the data set at a previously unsampled location [76]. Two solutions are proposed in [76] which support only one variogram model and which require high communication complexity, interactivity and local computation. In [77] collaborative private Kriging was investigated, where users combine their data sets to gain more accurate Kriging predictions.

1.4.4 The Optimal Information Ratio of Secret Sharing Schemes

Secret sharing is a fundamental primitive in cryptography, and it has been previously applied to build other cryptographic schemes. The most compelling case to the *CLARUS* project is that of attribute-based encryption schemes. In key-policy attribute-based schemes [24], a central authority distributes keys to a set of users, and keys are attached to a policy instantiated by an access structure. Users can encrypt messages while attaching them to attributes, and the resulting ciphertexts can only be decrypted by users holding keys with a matching policy.

In [24] Goyal et al. show how to use perfect linear secret sharing schemes to build attribute-based encryption schemes, and this strategy has been adopted in several other works. Yet, the key length resulting from their strategy is directly proportional to the information ratio of the employed secret sharing scheme, and thus the efficiency of the underlying secret sharing scheme has a direct impact on the computational and communication complexity of the resulting attribute-based encryption scheme. For many access structures, schemes built using known general methods [52, 53, 78] have been shown to have exponential information ratio. Therefore, this raises the question of which access structures admit an efficient secret sharing scheme.

In this sense, the optimal information ratio gives an idea of the well-conditioning of the access structure. The *optimal information ratio* of an access structure is the infimum of the information ratio of all the perfect secret sharing schemes realizing the access structure. In our research, we approach the problem of finding properties that facilitate the description of the optimal information ratio. More concretely, we show that any two access structures that are close (i.e., whose symmetric difference is small) admit secret sharing schemes with similar information ratios. Therefore, if we modify the access

structure of an efficient secret sharing scheme by adding or removing a small amount of authorized sets, we will still be able to find an efficient scheme realizing the modified access structure.

Computing the optimal information ratio is generally a difficult task, and concrete values are known only for certain families of access structures. For instance, since threshold access structures admit ideal access structures [50] their optimal information ratio is 1. Other examples of access structures with known optimal information ratio include particular families of multipartite access structures [79–81], access structures with a small number of participants [82] and access structures with small minimal sets [83]. However, it is not currently known how to compute the optimal information ratio of general access structures. Our research is a step forward in addressing this problem, since we provide theoretical bounds for the optimal information ratio. Our results effectively expand the number of cases in which efficient secret sharing schemes are known to exist.

By taking advantage of the combinatorial nature of our main result, we also extend our results to other models of computation such as the formula leafsize and the monotone span program size for monotone Boolean functions.

1.4.5 Privacy-Preserving Data Splitting

The privacy-preserving data splitting technique aims at preserving the privacy of a sensitive data set by decomposing it into fragments. In a cloud computing setting, data splitting techniques are usually deployed in multi-cloud architectures. Assuming that the cloud service providers are not aware of each other, data splitting can be used to enforce data privacy by distributing the data fragments among different cloud servers. In this thesis, we formulate the splitting problem in combinatorial terms. Using this formulation, we describe three general algorithms for data splitting, and we provide theoretical bounds on the required number of fragments.

The existing data splitting literature [54–58] divides the data into a small number of fragments, frequently into two or three. Since this small number of fragments does not usually suffice to ensure privacy, existing solutions couple cryptographic techniques with data splitting, and they use encryption on part of the data fragments. In contrast, we engage with the data splitting problem without relying on other privacy-preserving techniques. This approach allows to keep all fragments in plaintext form, which can be useful in terms of efficiency and functionality.

In our contribution to data splitting we take two sets of constraints into account: privacy constraints and processing constraints. Privacy constraints determine the sensitive attribute combinations, which no single cloud server must simultaneously hold. Processing constraints impose some sets of data attributes to be stored together in some fragment, to reduce query workloads.

Firstly, we present a reformulation of the stated problem in combinatorial terms, by introducing the notion of $(\mathcal{A}, \mathcal{B})$-coverings. Using this reformulation, we start by considering the problem of finding the optimal number of fragments required to satisfy privacy and processing constraints. We show that it can be solved using purely algebraic techniques, and we exhibit an algorithm to compute optimally-sized data decompositions by using Gröbner bases.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Since finding an optimal covering is an NP-hard problem, obtaining optimal solutions through our first algorithm is often unfeasible in practice. We hence present greedy algorithm that sacrifices the solution optimality to obtain efficient running times, achieving polynomial running time in the size of the considered problem. We further present an heuristic improvement of this greedy algorithm that provides smaller decompositions when the family of constraints is sparse enough. Our greedy algorithms require milliseconds to find a solution in the studied cases, whereas computing an optimal solution may require hours depending on the problem at hand. Theoretical upper and lower bounds on the number of fragments are also provided in the process.

1.5 Publication List

The following publications and preprints form the core of this thesis.

Publications:

1. O. Farràs, J. Ribes-González, and S. Ricci. Local bounds for the optimal information ratio of secret sharing schemes. *Designs, Codes and Cryptography*, 2018. URL `https://doi.org/10.1007/s10623-018-0529-7`
2. J. Alderman, B.R. Curtis, O. Farràs, K.M. Martin, and J. Ribes-González. Private outsourced Kriging interpolation. In *Financial Cryptography and Data Security (FC'17) International Workshops: 4th Workshop on Encrypted Computing and Applied Homomorphic Cryptography (WAHC'17), Revised Selected Papers*, pages 75–90. Springer International Publishing, 2017
3. O. Farràs and J. Ribes-González. Searchable encryption for geo-referenced data. In *15th IFIP Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net'16)*, pages 1–8, 2016

Preprints and submitted articles:

1. O. Farràs, J. Ribes-González, and S. Ricci. Privacy-preserving data splitting: A combinatorial approach. *Submitted, preprint at arXiv:1801.05974*, 2018
2. O. Farràs and J. Ribes-González. Provably secure public key encryption with conjunctive and subset keyword search. *Submitted to the International Journal on Information Security*, 2017

Chapter 2

Background Theory

In this chapter we lay out the fundamental cryptographic theory needed in the rest of this thesis. This chapter is divided in six sections. In the first section, we introduce some basic mathematical notation and definitions. In the second section we present symmetric-key encryption schemes and the fundamentals behind computational provable security. Next, in the third section we introduce hash functions and pseudorandom functions. The remaining three sections deal with the foundational theory behind three of the cryptographic schemes explored in this thesis: public-key encryption schemes (with a special mention of homomorphic encryption), searchable encryption and secret sharing. Most of the material in the first four sections is taken from [88, 89].

2.1 Basic Definitions

Given a function $f : \mathbb{N} \to \mathbb{R}_{\geq 0}$, we say that $f(\lambda)$ is *negligible in* λ if for every positive polynomial p in one variable there exists a λ_0 such that for all integers $\lambda > \lambda_0$ we have that $f(\lambda) < 1/p(\lambda)$. This statement is equivalent to $f(\lambda) = o(1/\lambda^c)$ for all $c \in \mathbb{N}$. Thus, f is negligible in λ if it decreases faster than any polynomial in λ for all sufficiently large values of λ.

We denote by $\{0,1\}^*$ the set of strings of finite length with characters 0 and 1.

An *algorithm* is a well-defined computational procedure $\mathcal{A}$ that takes a variable x as input and outputs a function y of this input. We often denote this by $y \leftarrow \mathcal{A}(x)$ or by $y = \mathcal{A}(x)$. An algorithm is *probabilistic* if it takes auxiliary random bits as input.

We say an algorithm $\mathcal{A}$ runs in *polynomial time* (or equivalently that it is *efficient*) if, for every input $x \in \{0,1\}^*$ of bit length $|x|$, the computation of $\mathcal{A}(x)$ takes $O(|x|^c)$ time for some constant c. A *probabilistic polynomial time* (PPT) algorithm is one that is probabilistic and runs in polynomial time. Given a function f, if an algorithm $\mathcal{A}$ has black-box access to the function f we denote it by $\mathcal{A}^{f(\cdot)}$.

2.2 Symmetric-Key Encryption Schemes

In the next definition we introduce symmetric-key encryption schemes. Symmetric-key encryption schemes typically deal with the problem of secret communication between two parties, assuming that a secret key has been previously shared between them. By using symmetric-key encryption schemes to encrypt and decrypt messages, two parties can privately exchange information through an insecure channel. In this way, the exchanged messages remain protected even in the presence of an eavesdropper who has access to all information exchanged through this channel.

Definition 2.1. An $(\mathcal{M}, \mathcal{C}, \mathcal{K})$ – *symmetric-key encryption scheme* $\mathcal{S}$ consists of three polynomial-time algorithms:

$\mathcal{S}.\mathrm{Gen}(\lambda)$**:**
Probabilistic algorithm that, given a security parameter λ, returns a secret (also called symmetric) key $k \in \mathcal{K}$ and the public parameters params of the scheme.

$\mathcal{S}.\mathrm{Enc}_k(m)$**:**
Probabilistic algorithm taking as input a message $m \in \mathcal{M}$, a key $k \in \mathcal{K}$ and the public parameters params of the scheme. It returns a ciphertext $c = \mathrm{Enc}_k(m) \in \mathcal{C}$.

$\mathcal{S}.\mathrm{Dec}_k(c)$**:**
Deterministic algorithm taking a ciphertext $c \in \mathcal{C}$ and the public parameters params of the scheme as input. It returns a plaintext $m = \mathrm{Dec}_k(c) \in \mathcal{M}$.

We say $\mathcal{S}$ is *correct* if, for every $m \in \mathcal{M}$, $\mathcal{S}.\mathrm{Dec}_k(\mathcal{S}.\mathrm{Enc}_k(m)) = m$ with all but negligible probability, where probabilities are taken over the choice of key k generated with the probabilistic algorithm $\mathcal{S}.\mathrm{Gen}$ and over the random bits of $\mathcal{S}.\mathrm{Enc}$.

Frequently used symmetric-key encryption schemes include, for example, the AES [90], 3DES [91] and Blowfish [92] encryption schemes.

As we see above, encryption schemes define three spaces: the message space $\mathcal{M}$, the ciphertext space $\mathcal{C}$ and the key space $\mathcal{K}$. From now on, we will drop all reference to these spaces whenever their choice is implicit or irrelevant. Also, in practice, all key-generating functions $\mathcal{S}.\mathrm{Gen}$ typically just choose the key k uniformly at random from the key space $\mathcal{K}$. This will be the case in all of our schemes.

2.2.1 Provable Security

Traditional cryptography revolves around a break-then-fix paradigm where proposed schemes are deemed secure as long as no successful attacks breaking their claimed security are devised. Due to this approach all traditional encryption schemes have been broken, and are deemed unsuitable for cryptographic purposes at this point. This illustrates the important lesson that designing secure encryption schemes is hard. Although very often the first steps of cryptographic design still starts with this break-and-fix approach, modern cryptography relies in a more deep treatment of security.

In this subsection we give an account of this treatment in the setting of symmetric-key encryption schemes. We do it in three stages. We initially discuss security definitions,

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

which formalize the security properties sought in symmetric-key encryption schemes. After a short digression to define bilinear groups, we present computational hardness assumptions, which are necessary hypotheses to elaborate computational security proofs. Finally, we describe security proofs, which give solid evidence on the security of cryptographic constructions, according to some security definition and possibly accepting some computational security assumption as a premise.

2.2.1.1 Security Definitions

Security definitions describe the security properties of cryptographic constructions, establishing concrete security guarantees against characterized adversaries by means of precise mathematical statements. They are the main ingredient that enable the security proofs of particular solutions. They allow refuting security as well, by showing that a particular solution does not satisfy a given security definition.

For any cryptographic primitive there may be many valid ways to define security, and therefore several valid formal security definitions may exist. A security definition may or may not be adequate for a particular application depending on the way it is used. Moreover, in many cases we observe a trade-off between security and efficiency. It is often the case that weaker security definitions enable more efficient solutions than stronger security definitions. Thus, security definitions also allow practitioners to choose between cryptographic solutions, generally by adopting the most efficient solution among the ones whose security definition suits the required security needs.

Generally, security definitions have two components: a security guarantee and a threat model.

The security guarantee describes which security threats can be contained, in terms of which information is protected from a potential attacker. Equivalently, it describes in which situations the scheme can be considered to be broken. In the context of encryption, most classical security definitions provide roughly the same security guarantee: that the attacker does not learn any additional information about the underlying messages from ciphertexts, aside from the information it already owns.

The threat model shapes the adversary, describing the information it has access to, without restricting the particular strategy it can follow when launching attacks. The threat model captures the fact that real world adversaries may or may not be able to observe certain information when attacking the system. In the context of encryption, classical security definitions describe the following threat models, in increasing order of strength:

- Ciphertext-only attack: The adversary only has access to a challenge ciphertext, and tries to obtain information about the underlying plaintext. Typically, it is assumed that the adversary knows the length of the plaintext, but nothing else about it.
- Known-plaintext attack: The adversary learns a set of plaintext/ciphertext pairs. Then it tries to derive information about the underlying plaintext of a challenge ciphertext.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

- Chosen-plaintext attack: The adversary is able to obtain any number ciphertexts from plaintexts of its choice. It then tries to derive information about the underlying plaintext of a challenge ciphertext that it did not already receive.
- Chosen-ciphertext attack: The adversary is able to gather information by querying for the decryption of ciphertexts of its choice. Then it tries to derive information about the underlying plaintext of a non-queried challenge ciphertext.

The most common type of security definitions, namely game-based security definitions, describe an adversarial setting and a set of possible attacks through a security game. Then they state that any PPT adversary has negligible probability in executing these attacks successfully.

We now introduce a game-based security definition for symmetric encryption schemes used in our work, namely *Indistinguishability against Chosen Plaintext Attacks* (or IND-CPA). In the following, we introduce the definition by means of a game between two entities, a challenger and an adversary. This game captures the chosen-plaintext attack threat model, by letting the adversary query the challenger for the encryption of chosen plaintexts. At the challenge phase, the adversary chooses two messages and is handed a challenge ciphertext corresponding to one of them. The aim of the adversary is then distinguishing which message this challenge ciphertext corresponds to. The security definition simply states that any PPT adversary has negligible success probability in winning this game.

Let $\mathcal{S} = \{\mathcal{S}.\text{Gen}, \mathcal{S}.\text{Enc}, \mathcal{S}.\text{Dec}\}$ be an encryption scheme. Given a security parameter λ, define a security game in the following five phases:

- *Setup.* The challenger runs $\mathcal{S}.\text{Gen}$ on input λ, obtains the secret key k, and hands over the public parameters to the adversary.
- *Query Phase 1.* The adversary adaptively requests the challenger for ciphertexts $\mathcal{S}.\text{Enc}_k(m)$ for a polynomial number of messages m of its own choice.
- *Challenge.* The adversary outputs two challenge candidate messages m_0, m_1 of the same length. The challenger throws a fair coin $b \in \{0, 1\}$, and outputs the ciphertext $c = \mathcal{S}.\text{Enc}_\text{k}(m_b)$ corresponding to m_b.
- *Query Phase 2.* The adversary proceeds just as in Query Phase 1.
- *Guess.* The adversary outputs a guess $b' \in \{0, 1\}$ for b.

Definition 2.2 (Indistinguishability against Chosen Plaintext Attacks)**.** We say that an encryption scheme $\mathcal{S} = \{\mathcal{S}.\text{Gen}, \mathcal{S}.\text{Enc}, \mathcal{S}.\text{Dec}\}$ satisfies indistinguishability against chosen plaintext attacks if the advantage of every PPT adversary $\mathcal{A}$ in distinguishing b in the above game

$$\begin{aligned}\text{Adv}_{\mathcal{A}}(\lambda) =& |\Pr(b' = b) - 1/2| \\ =& |\Pr(\mathcal{A}(X) = b | X = b) - \Pr(\mathcal{A}(X) = b | X = 1 - b)|\end{aligned}$$

is negligible in λ.

Another security definition used in related works [37] is *Pseudo-Randomness against Chosen-Plaintext Attacks* (IND-PCPA). This security notion differs from IND-CPA only in the challenge phase, where the adversary commits to a single message m, and the challenger outputs either $c = \mathcal{S}.\mathrm{Enc}_\mathrm{k}(m)$ or a uniformly-random chosen ciphertext depending on the outcome of the fair coin b.

It is worth noting that not all security definitions are game-based. Another common style of security definition used in referenced works [37, 93] is the *real-ideal paradigm*, also called *simulator-based*. Simulator-based definitions are introduced through a *real world* and an *ideal* (or *simulated*) *world* setting. In the real world, the adversary interacts with honest parties in a realistic setting involving the actual cryptographic solution. In the ideal world, the adversary interacts with parties named simulators, which only take public, non-sensitive information and randomness as input and do not generally use any cryptography. Simulator-based security definitions state that there exists some choice of simulators for which no polynomial-time algorithm can distinguish the view of the adversary in these two settings. That is, there does not exist a PPT distinguisher algorithm that is able to tell whether the interaction was simulated or not. See [94] for an introduction to simulator-based security definitions and proofs.

One of the earliest simulator-based definitions in the literature is *semantic security* [95]. This security definition is equivalent to the IND-CPA definition presented above. The adversary starts by randomly choosing a message m. In the ideal world, a simulator algorithm receives certain information $h(m)$ of some message m as input, while in the real world a PPT algorithm additionally receives an encryption $\mathrm{Enc}_k(m)$ of the message m. The scheme $\mathcal{S}$ is semantically secure if the probability that both algorithms output some property $f(m)$ of the message m is negligibly close for all efficiently computable functions h, f. Intuitively, the ciphertext $\mathrm{Enc}_k(m)$ does not provide any extra knowledge about $f(m)$ not given already by $h(m)$. Note that the distinguisher algorithm here is embedded in the property f.

Definition 2.3 (Semantic Security [88])**.** We say that an encryption scheme $\mathcal{S} = \{\mathcal{S}.\mathrm{Gen}, \mathcal{S}.\mathrm{Enc}, \mathcal{S}.\mathrm{Dec}\}$ is semantically secure if for every PPT algorithm $\mathcal{A}$, all efficiently computable functions f, h and every distribution χ on the message space $\mathcal{M}$ (all of which depend on the security parameter λ), there exists a simulator $\mathcal{B}$ such that

$$|\Pr\left(\mathcal{A}(\mathcal{S}.\mathrm{Enc}_k(m), h(m)) = f(m)\right) - \Pr\left(\mathcal{B}(h(m)) = f(m)\right)|$$

is negligible in λ, where the probabilities are taken over the choice of m according to χ, over the choice of key k generated with $\mathcal{S}.\mathrm{Gen}$, and over the random coins used by $\mathcal{A}$, $\mathcal{B}$ and by the encryption algorithm.

2.2.1.2 Digression: Bilinear Groups

Before continuing on to the next section, we make a brief digression to define symmetric and asymmetric bilinear groups. In this thesis, group operations are often written multiplicatively, even when the group is abelian.

Definition 2.4 (Bilinear Groups)**.** Let $(\mathbb{G}_1, \cdot), (\mathbb{G}_2, \cdot)$ be two cyclic groups of prime order q with generators g, h respectively (usually denoted by $\mathbb{G}_1 = \langle g \rangle$, $\mathbb{G}_2 = \langle h \rangle$), and suppose that there exists a cyclic group $\mathbb{G}_T$ of order q and a non-degenerate bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$.

We say that $\mathbb{G}_1$ is a *symmetric bilinear group* if there exists an efficiently computable isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$. Under such an isomorphism, we denote $\mathbb{G}_1$ and $\mathbb{G}_2$ by $\mathbb{G}$.

Similarly, we say that $\mathbb{G}_1, \mathbb{G}_2$ are *asymmetric bilinear groups* if there exist no non-trivial efficiently computable homomorphisms from $\mathbb{G}_2$ to $\mathbb{G}_1$.

In practice, the bilinear groups $\mathbb{G}, \mathbb{G}_1, \mathbb{G}_2$ are taken to be subgroups of the group of points of an elliptic curve, and $\mathbb{G}_T$ is a subgroup of the multiplicative group of a finite field [96]. The definition of symmetric and asymmetric bilinear groups corresponds to Type 1 and Type 3 pairings in the article by Galbraith et al. [97]. The term *pairing* refers to the non-degenerate bilinear map in the definition of bilinear group. We refer the reader to their article for properties of particular instantiations, and to [98, 99] for techniques to speed up pairing computation.

2.2.1.3 Computational Hardness Assumptions

Although in some cases security can be proved unconditionally and can take into account computationally unlimited attackers, some cryptographic constructions do not admit such proofs. Instead, it is possible to achieve higher efficiency and a greater variety of cryptographic designs by falling back to computational security and by making use of computational hardness assumptions.

Computational security consists in assuming that the adversary is computationally limited. That is, computationally secure schemes can be instantiated in such a way that the current technologies need an impracticable (albeit finite) amount of time to break them.

Computational security proofs ultimately rest on assumptions. Assumptions can generally be of two types: assumptions on the security of a cryptographic primitive and computational hardness assumptions. The first type simply assumes the security of a cryptographic primitive used as a building block in the actual scheme. In turn, computational hardness assumptions lie on a lower level and state that a given problem, which is usually of number-theoretical nature, can not be solved in a practical amount of time. Security proofs are developed based on these assumptions.

More precisely, computational hardness assumptions refer to the computational intractability of precisely stated mathematical problems. These mathematical problems are typically parametrized by a security parameter. Computational hardness assumptions state that there is no efficient general algorithm that solves a problem in polynomial time in the security parameter for all large enough parameter choices. In other words, they state that the success probability of any probabilistic polynomial-time adversary in solving the parametrized problem is negligible in the security parameter.

One of the hardest and most fundamental assumptions for cyclic groups in computational security is the computational intractability of the *Discrete Logarithm Problem* (DLP). We recall that if g is a generator of a multiplicative group $\mathbb{G}$ and if $h \in \mathbb{G}$, then any integer a that solves $h = g^a$ is called a discrete logarithm of h to the base g. The DLP assumption holds in a multiplicative cyclic group $\mathbb{G}$ when discrete logarithms are generally hard to compute.

Definition 2.5 (Discrete Logarithm Problem)**.** Let $\mathbb{G} = \langle g \rangle$ be a multiplicative cyclic group of prime size q (where q has at least λ bits) deterministically generated according to the security parameter λ. We say the DLP *assumption* holds in $\mathbb{G}$ if for every PPT algorithm $\mathcal{B}$,

$$\mathrm{Adv}_{\mathcal{B}}(\lambda) = \Pr\left(\mathcal{B}(g, g^a) = a\right)$$

is negligible in λ, where the probability is taken over a uniformly distributed in $\mathbb{F}_q$ and over the random bits of $\mathcal{B}$.

At the time of writing this, for the elliptic curve groups $\mathbb{G}$ used in practice [100], there are no known classical algorithms for solving the DLP in sub-exponential time in λ. We should note here that there exists an efficient quantum algorithm, due to Peter Shor, that efficiently breaks the DLP assumption (and several other frequently used assumptions) over generic groups. Thus, advances in quantum computing may render this assumption invalid, along with the security claims of a large corpus of cryptographic constructions in the literature.

The DLP assumption has given rise to several cryptographic solutions. However, considering possibly stronger assumptions (i.e., more easily solvable problems) turns out to enable the design of a greater diversity of constructions. One of the first relaxations of the DLP assumption is the Computational Diffie-Hellman (CDH) Assumption. The security of many cryptographic designs, such as the Diffie-Hellman key exchange protocol, rests on it. The CDH assumption holds in a multiplicative cyclic group $\mathbb{G} = \langle g \rangle$ if it is hard to compute g^{ab} from g, g^a and g^b in a vast majority of cases.

Definition 2.6 (Computational Diffie-Hellman Assumption)**.** Let $\mathbb{G} = \langle g \rangle$ be a multiplicative cyclic group of prime size q (of bit-length $|q| \geq \lambda$) deterministically generated according to the security parameter λ. We say the CDH *assumption* holds in $\mathbb{G}$ if for every PPT algorithm $\mathcal{B}$,

$$\mathrm{Adv}_{\mathcal{B}}(\lambda) = \Pr\left(\mathcal{B}(g, g^a, g^b) = g^{ab}\right)$$

is negligible in λ, where the probability is taken over a, b uniformly distributed in $\mathbb{F}_q$ and over the random bits of $\mathcal{B}$.

The CDH assumption implies the DLP assumption. An efficient CDH solver can be built from an efficient DLP solver, simply by letting it compute the exponent a of g^a, and raising g^b to that same exponent. However, it is unknown if DLP and CDH are equivalent. Thus, it might be the case that the CDH assumption is false and the DLP assumption is true. Nevertheless, the best known algorithm breaking CDH is to solve the DLP in $\mathbb{G}$.

An attractive assumption even stronger than CDH is the Decisional Diffie-Hellman (DDH) assumption, proposed by Boneh in [101]. The DDH assumption holds in a multiplicative cyclic group $\mathbb{G} = \langle g \rangle$ if, given the elements g, g^a, g^b, g^r, it is generally hard to decide whether or not r equals ab.

Definition 2.7 (Decisional Diffie-Hellman Assumption)**.** Let $\mathbb{G} = \langle g \rangle$ be a multiplicative cyclic group of prime size q (of bit-length $|q| \geq \lambda$) deterministically generated according to the security parameter λ. We say the DDH *assumption* holds in $\mathbb{G}$ if for every PPT

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

algorithm $\mathcal{B}$,

$$\mathrm{Adv}_{\mathcal{B}}(\lambda) = \left|\Pr\left(\mathcal{B}(g, g^a, g^b, g^{ab}) = 1\right) - \Pr\left(\mathcal{B}(g, g^a, g^b, g^r) = 1\right)\right|$$

is negligible in λ, where the probabilities are taken over a, b, r uniformly distributed in $\mathbb{F}_q$ and over the random bits of $\mathcal{B}$.

The DDH assumption enables the construction of a remarkably wide variety of efficient cryptographic systems with strong security properties. Most notable examples are the ElGamal and the Cramer-Shoup cryptographic schemes.

Now, one of our proposed schemes is proved secure under a bilinear version of the DDH assumption, called the asymmetric Decisional Bilinear Diffie-Hellman assumption (asymmetric DBDH). This assumption is proposed in the work [102] by Boneh and Boyen as a generalization of the DBDH assumption (see [103]) to the asymmetric bilinear setting. The DBDH assumption is easily seen to imply DDH in the target group $\mathbb{G}_T$.

Definition 2.8 (Asymmetric DBDH Assumption)**.** Let $\mathbb{G}_1 = \langle g \rangle$, $\mathbb{G}_2 = \langle h \rangle$ be asymmetric bilinear groups of prime size q (of bit-length $|q| \geq \lambda$) deterministically generated according to a security parameter λ. We say the *asymmetric* DBDH *assumption* holds in $\mathbb{G}_1$ and $\mathbb{G}_2$ if for every PPT algorithm $\mathcal{B}$,

$$\begin{aligned}\mathrm{Adv}_{\mathcal{B}}(\lambda) = \Big|&\Pr\left(\mathcal{B}(g, g^a, g^b, h, h^a, h^c, e\,(g,h)^{abc}) = 1\right) \\ &-\Pr\left(\mathcal{B}(g, g^a, g^b, h, h^a, h^c, e\,(g,h)^{r}) = 1\right)\Big|\end{aligned}$$

is negligible in λ, where the probabilities are taken over a, b, c, r uniformly distributed in $\mathbb{F}_q$ and over the random bits of $\mathcal{B}$.

Another assumption we make use of is the Bilinear Diffie Hellman Inversion Assumption ($p-$BDHI). This assumption was first proposed in [102]. According to [100], the best known algorithm breaking $p-$BDHI is to solve the DLP in $\mathbb{G}$.

Definition 2.9 ($p-$BDHI Assumption)**.** Let $\mathbb{G} = \langle g \rangle$ denote a symmetric bilinear group of prime size q (of bit-length $|q| \geq \lambda$) deterministically generated according to a security parameter λ, and let p be a positive integer. We say the $p-$BDHI *assumption* holds in $\mathbb{G}$ if for every PPT algorithm $\mathcal{B}$,

$$\mathrm{Adv}_{\mathcal{B}}(\lambda) = \Pr\left(\mathcal{B}(g, g^a, g^{a^2}, \ldots, g^{a^p}) = e\,(g,g)^{1/a}\right)$$

is negligible in λ, where the probabilities are taken over uniformly distributed $a \in \mathbb{F}_q$ and over the random bits of $\mathcal{B}$.

2.2.1.4 Security Proofs

In many instances, it is possible to give strong evidence that a cryptographic design is secure. This can be done by proving that it satisfies a security definition, possibly under certain assumptions. Provided the proof is correctly carried out, a security proof gives a sound guarantee that no attack captured in the security definition will succeed, assuming that the assumptions made are all valid. Also, security definitions do not

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

usually capture side-channel attacks or implementation-related attacks, and therefore such attacks are out of scope in this treatment of security.

In order to rely in the security of a provably secure scheme in a practical setting, practitioners have to verify that the security definition matches the abilities of real-world adversaries and the existing threats. In addition, they must regard the assumptions as true, and choose a security parameter for which the problem or the scheme in the hardness assumption is thought to be computationally unfeasible or secure for current technologies. The security parameter thus allows honest parties to adjust security to a desired level, enabling them to defend against increases in classical computing power.

In computational security we model both adversaries and challengers by PPT algorithms, so they are computationally bounded. The input size of adversaries can be assumed to be polynomial in the security parameter, and we only consider adversaries whose running time is also polynomial in the security parameter. Moreover, we assume that all parties have access to a randomness source of their own, and this is indeed the case in a real setting.

Computational security proofs state that the validity of the assumption implies the security of the cryptographic scheme according to the security definition. This is equivalent to saying that breaking the cryptographic scheme is at least as hard as breaking the assumption. When comparing the relative difficulties of problems, one often resorts to the notion of *reducibility*.

Definition 2.10 ([89])**.** Let L_1 and L_2 be decision problems. Then L_1 is (polynomial-time) *reducible* to L_2 if there exists a polynomial-time algorithm that solves L_1 by using an algorithm that solves L_2 as a subroutine.

Intuitively, if L_1 is reducible to L_2, then L_2 is at least as difficult as L_1. This is the basis of security reductions, a technique frequently used to prove security according to game-based security definitions. Typically, security reductions build an algorithm $\mathcal{B}$ that breaks the hardness assumption by making black-box use of a subroutine $\mathcal{A}$ that is assumed to break the scheme according to the security definition. In this mathematical reduction, it is essential that the view of $\mathcal{A}$ provided by $\mathcal{B}$ is statistically close to the view it has in the security definition of the scheme.

As for security proofs according to simulator-based definitions, they usually follow a different strategy. Proofs in this setting are often carried out by considering an arbitrary adversary, and then explicitly describing a simulator algorithm that simulates the behavior of the adversary well enough to satisfy the security definition. As an example, see for instance [37].

Security proofs can be carried out in a variety of computational models. Some proofs presented in this thesis are carried out in the *Standard Model*, which only imposes limits on the computational power of all parties. On the other hand, some proofs presented in this thesis are carried out in the *Random Oracle Model* (ROM, see [104]). The random oracle model of computation builds on top of the Standard Model. The ROM additionally assumes that there exists a random oracle, i.e., a mathematical function that maps each possible query to a fixed random response from its output domain. In our security proofs, we assume that some of the low-level cryptographic primitives used in our schemes perfectly emulate a random oracle. Although we assume this fact without any rigorous proof, this is a common assumption in cryptography that allows

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

the construction of a wider variety of more efficient schemes than in the Standard Model. Proofs in the Standard Model are nonetheless preferable to proofs in the ROM.

2.3 Hash Functions and Pseudorandom Functions

The task of building cryptographic schemes often involves making use of lower-level cryptographic primitives. We present here some of the cryptographic primitives that are employed throughout this thesis: *pseudorandom functions* and *hash functions*. Most definitions in this section are taken from [88].

2.3.1 Pseudorandom Functions

A pseudorandom function (PRF) is an efficiently computable function that emulates random oracles for all practical purposes. That is, outputs of a PRF appear as fixed completely at random for every input.

Let Func_n denote the set of functions with domain and range $\{0,1\}^n$. Given a *key space* $\mathcal{K}$ and a function $F : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$, denote by F_k the function defined by $F_k(x) = F(k,x)$. Typically $\mathcal{K} = \{0,1\}^n$. Suppose, in all instances, that the input length, the output length and the key length are of polynomial size in the security parameter λ.

Pseudorandom functions efficiently realize the notion of "random-looking" functions in Func_n, in the sense that no PPT algorithm can effectively distinguish between functions chosen uniformly at random from Func_n and functions F_k for uniform choices of key $k \in \mathcal{K}$.

Definition 2.11. Let $F : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ be an efficiently computable function, where the positive integer n is polynomial in the security parameter λ. Then, F is a *pseudorandom function* if for every PPT algorithm $\mathcal{B}$

$$\text{Adv}_{\mathcal{B}}(\lambda) = \left|\Pr\left(\mathcal{B}^{F_k(\cdot)}(\lambda) = 1\right) - \Pr\left(\mathcal{B}^{f(\cdot)}(\lambda) = 1\right)\right|$$

is negligible in λ, where the probabilities are taken over the uniform choices of $k \in \mathcal{K}$ and of $f \in \text{Func}_n$, and over the random bits of $\mathcal{B}$.

We also introduce the notion of pseudorandom permutation (PRP). Let Perm_n denote the set of permutations on the set $\{0,1\}^n$ with polynomial-sized input and output. The notion of PRP is analogous to that of PRF, with the only difference being that a randomly chosen PRP must be indistinguishable from a uniformly chosen permutation of Perm_n.

Definition 2.12. Let $F : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ be an efficiently computable function, where the positive integer n is polynomial in the security parameter λ. Then, F is a *pseudorandom permutation* if for every PPT algorithm $\mathcal{B}$

$$\text{Adv}_{\mathcal{B}}(\lambda) = \left|\Pr\left(\mathcal{B}^{F_k(\cdot)}(\lambda) = 1\right) - \Pr\left(\mathcal{B}^{f(\cdot)}(\lambda) = 1\right)\right|$$

is negligible in λ, where the probabilities are taken over the uniform choices of $k \in \mathcal{K}$ and of $f \in \text{Perm}_n$, and over the random bits of $\mathcal{B}$.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Even though it is currently unknown whether true PRF and PRP functions exist, many functions have been proven to satisfy the PRF and PRP definitions under some computational hardness assumptions. Frequently used PRF and PRP families include, for example, the HMAC-SHA [105] and the AES-CMAC [106] families.

2.3.2 Hash Functions

Hash functions are efficiently computable functions used to reduce the size of input data. Therefore, their range is typically much smaller than its domain. Values mapped by hash functions are typically called *digests*, or *hash values*, of the input data.

Definition 2.13. A *hash function* $\mathcal{H} = (\mathcal{H}.\text{Gen}, \mathcal{H}.H)$ consists of a pair of PPT algorithms:

$\mathcal{H}.\text{Gen}(\lambda)$**:**
Probabilistic algorithm that, given a security parameter λ, returns a key s.

$\mathcal{H}.H_s(x)$**:**
Probabilistic algorithm taking as input a key s a string $x \in \{0,1\}^*$ of polynomial length in λ. It returns a digest $H_s(x) \in \{0,1\}^*$ of smaller bit-length than x.

From now on, unless stated otherwise, we obviate the Gen algorithm and the generated key s when using hash functions, and we simply denote hash functions H_s by $H : \{0,1\}^* \to \{0,1\}^*$. We thus implicitly assume that Gen has been run once prior to the first call to H, and that the key s is thereby fixed.

One of the main properties we require from cryptographic hash functions in this work is *collision resistance*. Given a hash function H, a *collision* is a pair of distinct values $x, y \in \{0,1\}^*$ for which $H(x) = H(y)$. Since the domain of H is larger than its range, the existence of collisions is inevitable. However, they can be hard to compute. Collision resistance states that no PPT algorithm is able to find collisions with non-negligible advantage.

Given a security parameter λ, define a security game in the following two phases:

- *Setup.* The challenger runs $\mathcal{H}.\text{Gen}$ on input λ and hands over the generated key s to the adversary.
- *Guess.* The adversary determines two strings $x, y \in \{0,1\}^*$ of the same length. It outputs a bit $b = 1$ if $x \neq y$ and $H(x) = H(y)$. Otherwise, it outputs $b = 0$.

Definition 2.14 (Collision Resistance)**.** We say that a hash function $\mathcal{H}$ is collision resistant if the advantage of every PPT adversary $\mathcal{B}$ in breaking the above game

$$\text{Adv}_{\mathcal{B}}(\lambda) = \Pr(b = 1)$$

is negligible in λ, where the probability is taken over the choice of key s generated with the probabilistic algorithm $\mathcal{H}.\text{Gen}$ and over the random bits of $\mathcal{B}$.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Although it is currently unknown whether true collision-resistant functions exist, some hash functions can be seen to satisfy the collision-resistance definition under computational hardness assumptions such as the DLP assumption (see [88]).

Of course, when proving security under the random oracle model, if we assume that a hash function H perfectly emulates a random oracle and has a large enough range, then it is collision resistant. Therefore, in this case collision resistance is embedded in the ROM computational model.

2.3.2.1 Hashing onto Elliptic Curves

Many elliptic-curve based cryptographic schemes require hashing to a bilinear group $\mathbb{G} = \langle g \rangle$ of order p. That is, they require the existence of a hash function $H : \{0,1\}^* \to \mathbb{G}$. A direct way of hashing onto a bilinear group is to consider an additional hash function $h : \{0,1\}^* \to \mathbb{F}_p$ and then define $H(m) = g^{h(m)}$. However, this approach breaks security proofs in the random oracle model, since the output of H clearly depends on that of h. Note that, once h is queried at a certain input, H does no longer behave as a random oracle for that input.

In this setting, cryptographic schemes whose proofs are carried out in the random oracle model (such as the proofs in Section 3.2) require hashing directly onto an elliptic curve group $\mathbb{G}$. That is, they require hash functions that sample from $\mathbb{G}$ without computing multiples of a generator g. In this respect, using asymmetric bilinear groups $\mathbb{G}_1, \mathbb{G}_2$ guarantees that we can securely and efficiently hash onto $\mathbb{G}_1$. See [107] for an extensive discussion on this topic and for an explicit solution for secure hashing onto Barreto-Naehrig elliptic curves.

2.4 Public-Key Encryption Schemes

The main difference between symmetric-key and public-key cryptography lies in the assumed secrecy of keys. While in symmetric-key cryptography all keys are assumed to be kept secret, in public-key cryptography some of the keys are made public, or available to wider set of users. In public-key cryptography, public keys are typically derived from secret keys. The knowledge of secret keys generally enables more capabilities than the knowledge of public keys.

A first example of public-key cryptography are public-key encryption schemes. Consider a setting with two parties, the sender and the receiver. At the setup stage of a public-key encryption scheme, the receiver generates a pair of keys $(\mathrm{sk}, \mathrm{pk})$, where sk is the secret key and pk is the public key. Then, the receiver keeps sk secret and sends the public key pk over to the sender. The sender is then able to encrypt messages using pk, and the receiver can decrypt those same messages by using sk.

Definition 2.15. An $(\mathcal{M}, \mathcal{C}, \mathcal{K})$ – *public-key encryption scheme* $\mathcal{S}$ consists of three polynomial-time algorithms:

$\mathcal{S}.\mathrm{Gen}(\lambda)$**:**
: Probabilistic algorithm that, given a security parameter λ, returns a secret key and a public key $\mathrm{sk}, \mathrm{pk} \in \mathcal{K}$.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

$\mathcal{S}.\mathrm{Enc}_{\mathrm{pk}}(m)$**:**
Probabilistic algorithm taking as input a message $m \in \mathcal{M}$ and the public key $\mathrm{pk} \in \mathcal{K}$. It returns a ciphertext $c = \mathcal{S}.\mathrm{Enc}_{\mathrm{pk}}(m) \in \mathcal{C}$.

$\mathcal{S}.\mathrm{Dec}_{\mathrm{sk}}(c)$**:**
Deterministic algorithm run by the client and taking a ciphertext $c \in \mathcal{C}$ as input. It returns a plaintext $m = \mathcal{S}.\mathrm{Dec}_{\mathrm{sk}}(c) \in \mathcal{M}$.

We say that $\mathcal{S}$ is *correct* if $\mathcal{S}.\mathrm{Dec}_{\mathrm{sk}}(\mathcal{S}.\mathrm{Enc}_{\mathrm{pk}}(m)) = m$ with all but negligible probability for every $m \in \mathcal{M}$.

The security properties of public-key encryption schemes guarantee that, given the knowledge of pk, the sender is able to generate ciphertexts that can only be decrypted with the knowledge of sk. Therefore, if two parties know the public key of each other, public-key encryption schemes enable them to communicate privately even if they don't share common secret information. In practice, public keys are either distributed to many different senders privately or publicly disseminated through a public-key infrastructure to prevent tampering attacks from active adversaries.

While the public key functionality describes a clear advantage with respect to symmetric-key encryption schemes, all known efficient public-key schemes are at least two to three orders of magnitude slower than the most efficient symmetric-key encryption schemes [88]. Thus, their use in practice is often relegated to encrypting symmetric keys. Unsurprisingly, we encounter the same issue when comparing all kinds of public-key cryptographic schemes to their symmetric-key counterparts.

We now introduce a game-based security definition for public-key encryption schemes used in our work, *Indistinguishability against Chosen Plaintext Attacks* (or IND-CPA). This definition is the analog of Definition 2.2 in the public-key setting. We introduce the definition by means of a game between two entities, a challenger and an adversary. The game consists of three phases. In the Setup phase, the challenger generates the keying material and hands over the public key and the public parameters to the adversary. At the Challenge phase, the adversary chooses two messages and is handed a challenge ciphertext corresponding to one of them. In the Guess phase, the aim of the adversary is then to distinguish which message the challenge ciphertext corresponds to. The security definition, which gives the security guarantees, simply states that any PPT adversary has negligible success probability in winning this game.

Let $\mathcal{S} = \{\mathcal{S}.\mathrm{Gen}, \mathcal{S}.\mathrm{Enc}, \mathcal{S}.\mathrm{Dec}\}$ be a public-key encryption scheme. Given a security parameter λ, define a security game in the following three phases:

- *Setup.* The challenger runs $\mathcal{S}.\mathrm{Gen}$ on input λ, hands over the public parameters and the public key pk to the adversary and keeps the secret key sk private.
- *Challenge.* The adversary outputs two challenge candidate messages m_0, m_1 of the same length. The challenger throws a fair coin $b \in \{0, 1\}$, and outputs the ciphertext $c = \mathcal{S}.\mathrm{Enc}_{\mathrm{pk}}(m_b)$ corresponding to m_b.
- *Guess.* The adversary outputs a guess $b' \in \{0, 1\}$ for b.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Definition 2.16 (Indistinguishability against Chosen Plaintext Attacks)**.** We say that a public-key encryption scheme $\mathcal{S} = \{\mathcal{S}.\text{Gen}, \mathcal{S}.\text{Enc}, \mathcal{S}.\text{Dec}\}$ satisfies indistinguishability against chosen plaintext attacks if the advantage of every PPT adversary $\mathcal{A}$ in distinguishing b in the above game

$$\begin{aligned}\text{Adv}_{\mathcal{A}}(\lambda) =& |\Pr(b' = b) - 1/2| \\ =& |\Pr(\mathcal{A}(X) = b | X = b) - \Pr(\mathcal{A}(X) = b | X = 1 - b)|\end{aligned}$$

is negligible in λ.

2.4.1 Homomorphic Encryption

Homomorphic encryption schemes are cryptographic schemes that enable computing over encrypted data. Their homomorphic properties enable users to encrypt their data in such a way that operations on the encrypted data translate to operations on the underlying plaintexts. Thus, homomorphic encryption enables users to outsource their data in encrypted form to a non-trusted party and subsequently request computations to be performed on that data.

Homomorphic encryption schemes are classified according to their homomorphic properties. Cryptographic literature usually distinguishes three classes:

- Partially homomorphic encryption (PHE): PHE schemes are homomorphic encryption schemes that support just a single operation on ciphertexts. Depending on the particular scheme, they allow for addition or multiplication of ciphertexts. We respectively call them *additively* and *multiplicatively* homomorphic schemes.
- Somewhat homomorphic encryption (SHE): SHE schemes support both addition and multiplication of ciphertexts, but they support a limited number of operations. Most SHE schemes admit a large number of additions and a small number of products on ciphertexts. However, SHE schemes are usually more computationally expensive than PHE schemes.
- Fully homomorphic encryption (FHE): FHE schemes support an unlimited number of additions and multiplications on ciphertexts, thus enabling arbitrary computations over encrypted data. Unfortunately, their computational overhead is currently too large for FHE to be practical for most applications. Usually, FHE schemes are SHE schemes with an additional bootstrapping algorithm. When too many multiplications are performed, the expensive bootstrapping algorithm is used to process the encrypted data so that additional multiplications are allowed.

Most, though not all, traditional homomorphic encryption schemes are public-key. In this thesis we make black-box use of a public-key additively homomorphic encryption scheme. We now give an example of such an encryption scheme, the Paillier encryption scheme [108], and we present it as described in [88].

Definition 2.17. The Paillier Encryption scheme $\mathcal{S}$ is defined by the following algorithms:

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

$\mathcal{S}$.Gen(λ)**:**
Choose two λ-bit prime numbers p, q such that $\gcd(pq, (p-1)(q-1)) = 1$. Compute $n = pq$, $\nu = (p-1)(q-1)$ and $\mu = \nu^{-1} \bmod n$. Return the secret key $\text{sk} = (\nu, \mu)$ and the public key $\text{pk} = n$.

$\mathcal{S}$.Enc$_{\text{pk}}(m)$**:**
Given the message $m \in \mathbb{F}_n$, select a random $r \in \mathbb{F}_n^*$. Output the ciphertext $c = (n+1)^m \cdot r^n \bmod n^2$.

$\mathcal{S}$.Dec$_{\text{sk}}(c)$**:**
Given the ciphertext $c \in \mathbb{F}_{n^2}$, output the message $m = \frac{c^\nu - 1}{n} \mu \bmod n$.

The Paillier encryption scheme satisfies the IND-CPA security definition under the decisional composite residuosity assumption [108]. For a proof of correctness, see [88]. To see additivity, let $c_i = \text{Enc}_{\text{pk}}(m_i)$ for $i = 1, 2$ and note that

$$c_1 \cdot c_2 = (n+1)^{m_1} \cdot r_1^n \cdot (n+1)^{m_2} \cdot r_2^n = (n+1)^{m_1+m_2} \cdot (r_1 r_2)^n = \text{Enc}_{\text{pk}}(m_1 + m_2).$$

2.5 Searchable Encryption

In this section we survey the field of searchable encryption. This section starts with an introduction to the topic, defining the setting, architecture and some generalities required in later material. Then, we present each of the main themes in searchable encryption on a technical level, summarizing the state of the art and describing fundamental results and particularities. For an introduction to searchable encryption, see [109].

Searchable encryption is a cryptographic technique that aims to solve data confidentiality issues when searching over outsourced data in a cloud computing setting. Searchable encryption schemes allow a client to efficiently search over outsourced encrypted data, and they provide a clear description of which information is protected or leaked in the process.

The simplest setting in searchable encryption consists of two parties, a client and a server. The client first wants to upload a data set $\mathbf{D}$ to the server. This data set is composed of tuples (w_i, D_i), where the *document* D_i is any kind of plaintext (text, audio, images, etc.) and w_i is a *keyword* associated D_i. Once this data set is uploaded, the client may want to query the server for all documents associated to a particular keyword w. Having received this query, the server processes it, selects the data items (w_i, D_i) with matching keywords $w_i = w$, and returns the corresponding documents D_i to the client. The use of searchable encryption schemes preserves the efficiency of this whole process while guaranteeing that the server learns as little information as possible.

A searchable encryption scheme $\mathcal{S}$ first provides the client with a key k through the execution of an algorithm $\mathcal{S}$.Gen. Using this key, an algorithm $\mathcal{S}$.Enc allows the client to build an encryption $\mathbf{C}$ of the data set $\mathbf{D}$. Additionally, this algorithm creates a *searchable index* I in the process, which is uploaded to the server along with $\mathbf{C}$.

Afterwards, the client may want to retrieve the documents associated to the keyword w. The client then furnishes this keyword w to the algorithm $\mathcal{S}$.Trapdoor to efficiently generate a *trapdoor* (or *search token*) T encoding the given keyword. This trapdoor T is

then sent to the server. The server is able to combine the trapdoor T and the searchable index I by using a public algorithm $\mathcal{S}$.Search. In this way it efficiently obtains the encrypted documents associated to the keyword that T represents, and returns them to the client. The client, in turn, decrypts the encrypted documents and retrieves the desired search outcome.

One of the main aspects of a searchable encryption is efficiency. Indeed, alternative cryptographic schemes such as private information retrieval [35] and oblivious RAM schemes [33] provide much better security in certain practical contexts but are nevertheless impractical for efficiency reasons. Searchable encryption schemes are often very efficient, and they provide a suitable solution when dealing with large data sets, frequent queries and multiple clients. Of all the efficiency marks of searchable encryption schemes, one can argue that the most critical are the ones involved in the searching phase: the trapdoor generation time, the search time and the client decryption time. This is so because one can expect many queries to be issued for a single data set.

There are many popular trends in the searchable encryption research that aim to improve and extend the functionalities and properties of schemes. A desirable property is achieving sublinear search, that is, search time that is sublinear in the number of documents of the outsourced data set. When sublinear search is achieved, the search process does not necessarily touch all stored documents. Searchable encryption schemes have also been extended to support all sorts of different queries, such as range, conjunctive or Boolean queries. Another remarkable achievement of recent research are dynamic schemes (as opposed to static schemes), which enable the client to insert, update and/or delete documents from the outsourced data set. Complex architectures have been considered in the literature as well, by allowing various clients to search or insert documents to a common outsourced data set. Also, verifiability in searchable encryption enables the client to check the correctness of the search outcome in the face of a malicious server that may tamper with the search results.

The security properties of searchable encryption vary among different schemes. In order to guarantee efficiency in the encryption and in the search phases, searchable encryption makes a trade-off between security and efficiency. That is, queries using searchable encryption usually induce some *information leakage* on the information that is exchanged or stored. For example, if the search process returns the same search outcome for two different queries, it often reveals that the two queries are equivalent in the outsourced database. Security definitions in searchable encryption formalize and bound this leakage. Thus, searchable encryption provides an efficient solution suitable for applications where the leaked information is not sensitive. Some of the standard leakage patterns found in the literature are:

- Size pattern: The number of documents and/or trapdoors submitted by the client.
- Search pattern: Whether any two trapdoors encode the same keyword or not.
- Access pattern: Exactly which documents have been accessed when processing a query.
- Operation pattern: In dynamic searchable encryption schemes, the operation carried out (i.e., search, insert, update or delete).
- Update pattern: In dynamic searchable encryption schemes, which ones are the accessed, inserted, replaced, updated or deleted documents.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Much like traditional encryption schemes, searchable encryption schemes come in symmetric and public-key flavors. The setting explained above describes the symmetric-key case. In the public-key setting, there are two kinds of clients, writers and readers. Readers are able to generate trapdoors and search through the outsourced data, while writers can only generate searchable indexes and push them to the server. In the following subsections we describe each of both types, namely searchable symmetric encryption and public-key encryption with keyword search.

2.5.1 Searchable Symmetric Encryption

Consider the one writer and one reader setting described above, where a single client uploads a *document collection* (i.e., a data set) $\mathbf{D}$ to a server and then is able to query on it. The document collection is composed of tuples (W_i, D_i), where the *document* D_i is any kind of plaintext and $W_i = (w_{i,1}, \ldots, w_{i,q})$ is a *list of keywords* associated to D_i.

A searchable symmetric encryption (SSE) scheme $\mathcal{S}$ consists of five polynomial-time algorithms, $\mathcal{S} = \{\mathcal{S}.\text{Gen}, \mathcal{S}.\text{Enc}, \mathcal{S}.\text{Trapdoor}, \mathcal{S}.\text{Search}, \mathcal{S}.\text{Dec}\}$. The probabilistic algorithm $\mathcal{S}.\text{Gen}$ is first used to provide a client with a symmetric key k. Then, the client is able to generate a searchable index I and an encryption $\mathbf{C}$ of the data set $\mathbf{D}$ by executing $\mathcal{S}.\text{Enc}$ on input the symmetric key k and the data set $\mathbf{D}$.

Afterwards, the client may want to retrieve the documents satisfying a particular predicate f specified by a Boolean formula on a tuple of keywords. An example of such a predicate is $w_1 \wedge (w_2 \vee w_3)$, which is satisfied by documents whose associated keyword list contains the keyword w_1 and either w_2 or w_3. The client generates a trapdoor T by executing the algorithm $\mathcal{S}.\text{Trapdoor}$ on f, and sends T to the server. The server can then retrieve the encrypted documents satisfying f by combining the trapdoor T and the searchable index I and by using the public algorithm $\mathcal{S}.\text{Search}$.

We present below a general model for searchable symmetric encryption schemes. Note that dynamic [110] or verifiable [111] searchable encryption schemes may include additional algorithms.

Definition 2.18. A searchable symmetric encryption scheme $\mathcal{S}$ is comprised of five polynomial-time algorithms:

$\mathcal{S}.\text{Gen}(\lambda)$**:**
Probabilistic algorithm run by the client that, given a security parameter λ, returns a secret key k and the public parameters params of the scheme, including a symmetric-key encryption scheme $(\text{Gen}, \text{Enc}, \text{Dec})$.

$\mathcal{S}.\text{Enc}_k(\mathbf{D})$**:**
Probabilistic algorithm run by the client and taking as input a tuple of plaintexts $\mathbf{D} = (W_i, D_i)_{i=1}^N$, where W_i is a tuple of keywords attached to the document D_i. It returns an encrypted index I together with a tuple of encrypted documents $\mathbf{C} = (C_i)_{i=1}^N = (\text{Enc}_k(D_i))_{i=1}^N$ for the algorithm Enc in params.

$\mathcal{S}.\text{Trapdoor}_k(f)$**:**
Algorithm run by the client that takes a predicate on keywords f as input and returns a trapdoor T associated to f.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

$\mathcal{S}$.Search$(I, \mathbf{C}, T)$**:**
Deterministic algorithm run by the server and taking as input an encrypted index I, a tuple of ciphertexts $\mathbf{C}$ and a trapdoor T. It returns a tuple of elements of $\mathbf{C}$.

$\mathcal{S}$.Dec$_k(C)$**:**
Deterministic algorithm run by the client and taking an encrypted document C as input. It returns the output of Dec$_k(C)$ for the algorithm Dec in params.

We say that $\mathcal{S}$ is *correct* if for all security parameters λ, for all k output by $\mathcal{S}$.Gen, for all document collections $\mathbf{D}$ and for all $(I, \mathbf{C})$ output by $\mathcal{S}$.Enc, it holds with all but negligible probability that

$$\mathbf{D}(f) = \{\mathcal{S}.\text{Dec}_k(C) \ : \ C \in \mathcal{S}.\text{Search}(I, \mathbf{C}, \mathcal{S}.\text{Trapdoor}_k(f))\},$$

where $\mathbf{D}(f)$ are all documents in $\mathbf{D}$ whose attached keyword tuple satisfies f.

Searchable symmetric encryption schemes can be classified according to query expressiveness, i.e., to the class of predicates allowed by the $\mathcal{S}$.Trapdoor algorithm. If the algorithm $\mathcal{S}$.Trapdoor supports only single keywords as predicates, we say that $\mathcal{S}$ is a *single-keyword* SSE scheme. Alternatively, if $\mathcal{S}$.Trapdoor supports arbitrary conjunctions of such predicates, we say that $\mathcal{S}$ is a *conjunctive* SSE scheme. If $\mathcal{S}$.Trapdoor supports arbitrary Boolean formulas, we say that $\mathcal{S}$ is a *Boolean* SSE scheme. The searchable encryption literature deals with several other predicates, such as range [65, 93], substring [93] or subset [65] queries. We further comment on these in Chapter 3.

The first provably secure symmetric-key searchable encryption scheme was proposed in 2000 by Song, Wagner and Perrig [25]. In 2006, after a series of advances [112, 113], in the foundational work [37], Curtmola, Garay, Kamara and Ostrovsky introduced several rigorous and strong security definitions for searchable encryption, along with a scheme satisfying them. Their definitions have become standard in the searchable encryption literature, and their scheme is the basis for many other efficient SSE schemes.

Before describing the strongest security definition out of the two presented in [37], we define some concepts which have been informally introduced in the previous subsection.

Definition 2.19 (History)**.** Let $\mathbf{D} = (W_i, D_i)_{i=1}^N$ be a document collection. A *q-query history over* $\mathbf{D}$ is a tuple $H = (\mathbf{D}, \mathbf{w})$ that includes the document collection $\mathbf{D}$ and a list of q keywords $\mathbf{w} = (w_1, \ldots, w_q)$.

The concept of q-query history formalizes the information outsourced, in the form of documents and single-keyword queries, to the cloud service provider in a cloud computing setting. As explained above, the access and search patterns respectively refer to the information of which documents have been accessed when processing a query and to the information of whether any two single-keyword trapdoors encode the same keyword or not.

Definition 2.20 (Access Pattern)**.** The *access pattern* induced by a q-query history $H = (\mathbf{D}, \mathbf{w})$ is the tuple $\alpha(H) = (\mathbf{D}(w_1), \ldots, \mathbf{D}(w_q))$, where $\mathbf{D}(w_i)$ denotes all documents in $\mathbf{D}$ matched by keyword w_i.

Definition 2.21 (Search Pattern)**.** The *search pattern* induced by a q-query history $H = (\mathbf{D}, \mathbf{w})$ is the $q \times q$ symmetric matrix $\sigma(H)$ where the (i, j)-th component $\sigma(H)_{i,j}$ is 1 if $w_i = w_j$ and is 0 otherwise.

The notion used in the security definitions in [37] is that of the *trace* of a history. The trace consists of the information about the history that searchable encryption schemes do typically leak.

Definition 2.22 (Trace)**.** The *trace* induced by a q-query history $H = (\mathbf{D}, \mathbf{w})$ is the sequence $\tau(H) = (|D_1|, \ldots, |D_N|, \alpha(H), \sigma(H))$ comprised of the lengths of the documents in $\mathbf{D}$ and the access and search patterns induced by H.

If $\mathbf{w}$ is empty, we denote $\tau(\mathbf{D}, \mathbf{w}) = \tau(\mathbf{D})$.

Next, we describe the *Adaptive Semantic Security* definition presented in [37]. In this definition, Curtmola et al. consider an adaptive stateful adversary, that is, an adversary $\mathcal{A}$ that chooses how to interact with the server by taking into account all the previously held information, including an internal *state variable* $\mathrm{st}_{\mathcal{A}}$.

Definition 2.23 (Adaptive Semantic Security for SSE [37])**.** Let $\mathcal{S} = \{\mathcal{S}.\mathrm{Gen}, \mathcal{S}.\mathrm{Enc}, \mathcal{S}.\mathrm{Trapdoor}, \mathcal{S}.\mathrm{Search}, \mathcal{S}.\mathrm{Dec}\}$ be a searchable symmetric encryption scheme. Let λ be the security parameter and $\mathcal{A} = (\mathcal{A}_0, \ldots, \mathcal{A}_q)$ and $\mathcal{B} = (\mathcal{B}_0, \ldots, \mathcal{B}_q)$ be tuples of PPT algorithms respectively defining the adversary and the simulator, where q is polynomial in λ. Consider the following probabilistic experiments $\mathbf{Real}_{\mathcal{S},\mathcal{A}}(\lambda)$ and $\mathbf{Sim}_{\mathcal{S},\mathcal{A}}(\lambda)$:

$\mathbf{Real}_{\mathcal{S},\mathcal{A}}(\lambda)$:
- $k \leftarrow \mathcal{S}.\mathrm{Gen}(\lambda)$
- $(\mathbf{D}, st_{\mathcal{A}}) \leftarrow \mathcal{A}_0(\lambda)$
- $(I, \mathbf{C}) \leftarrow \mathcal{S}.\mathrm{Enc}_k(\mathbf{D})$
- $(w_1, st_{\mathcal{A}}) \leftarrow \mathcal{A}_1(st_{\mathcal{A}}, I, \mathbf{C})$
- $T_1 \leftarrow \mathcal{S}.\mathrm{Trapdoor}_k(w_1)$
- for $2 \leq i \leq q$,
  - $(w_i, st_{\mathcal{A}}) \leftarrow \mathcal{A}_i(st_{\mathcal{A}}, I, \mathbf{C}, T_1, \ldots, T_{i-1})$
  - $T_i \leftarrow \mathcal{S}.\mathrm{Trapdoor}_k(w_i)$
- let $\mathbf{T} = (T_1, \ldots, T_q)$
- output $\mathsf{v} = (I, \mathbf{C}, \mathbf{T})$ and $st_{\mathcal{A}}$

$\mathbf{Sim}_{\mathcal{S},\mathcal{A},\mathcal{B}}(\lambda)$:
- $(\mathbf{D}, st_{\mathcal{A}}) \leftarrow \mathcal{A}_0(\lambda)$
- $(I, \mathbf{C}, st_{\mathcal{B}}) \leftarrow \mathcal{B}_0(\tau(\mathbf{D}))$
- $(w_1, st_{\mathcal{A}}) \leftarrow \mathcal{A}_1(st_{\mathcal{A}}, I, \mathbf{C})$
- $(T_1, st_{\mathcal{B}}) \leftarrow \mathcal{B}_1(st_{\mathcal{B}}, \tau(\mathbf{D}, (w_1)))$
- for $2 \leq i \leq q$,
  - $(w_i, st_{\mathcal{A}}) \leftarrow \mathcal{A}_i(st_{\mathcal{A}}, I, \mathbf{C}, T_1, \ldots, T_{i-1})$
  - $(T_i, st_{\mathcal{B}}) \leftarrow \mathcal{B}_i(st_{\mathcal{B}}, \tau(\mathbf{D}, (w_1, \ldots, w_i)))$
- let $\mathbf{T} = (T_1, \ldots, T_q)$
- output $\mathsf{v} = (I, \mathbf{C}, \mathbf{T})$ and $st_{\mathcal{A}}$

We say that $\mathcal{S}$ is *adaptively semantically secure* if for all PPT adversaries $\mathcal{A}$ there exists a PPT simulator $\mathcal{B}$ such that for every PPT distinguisher algorithm $\mathcal{D}$,

$$\begin{aligned}&|\Pr\left(\mathcal{D}(\mathsf{v}, st_{\mathcal{A}}) = 1 : (\mathsf{v}, st_{\mathcal{A}}) \leftarrow \mathbf{Real}_{\mathcal{S},\mathcal{A}}(\lambda)\right)\\ &-\Pr\left(\mathcal{D}(\mathsf{v}, st_{\mathcal{A}}) = 1 : (\mathsf{v}, st_{\mathcal{A}}) \leftarrow \mathbf{Sim}_{\mathcal{S},\mathcal{A},\mathcal{B}}(\lambda)\right)|\end{aligned}$$

is negligible, where the probabilities are taken over the random coins of $\mathcal{S}.\mathrm{Gen}$, $\mathcal{S}.\mathrm{Enc}$, $\mathcal{A}$ and $\mathcal{B}$.

The adaptive semantic security definition for SSE states that the only information potentially leaked by the scheme is the trace induced by the outsourced data set and by the queried keywords. However, the leakage of searchable encryption schemes can vary depending on the scheme and on the followed adversary model. More complex schemes may need a different description of leakage. To address this problem, in [114] Chase and Kamara introduced the notion of *leakage function*. Leakage functions generalize the

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

notion of trace: given a history as input, leakage functions output an upper bound on the information that the scheme leaks about the history.

We next introduce the $\mathcal{L}$-*Semantic Security against Adaptive Attacks* definition presented by Cash et al. in [93], which is used in this thesis. This security definition is conditioned to a predefined leakage function, and it is designed for the OXT Boolean SSE scheme defined in [93]. This security definition is conceptually equivalent to adaptive semantic security for $\mathcal{L} = \tau$. We modify the original formulation to fit the notation of Definitions 2.18,2.23.

Definition 2.24 ($\mathcal{L}$-Semantic Security against Adaptive Attacks [93])**.** Let $\mathcal{S} = \{\mathcal{S}.\text{Gen}, \mathcal{S}.\text{Enc}, \mathcal{S}.\text{Trapdoor}, \mathcal{S}.\text{Search}, \mathcal{S}.\text{Dec}\}$ be a searchable symmetric encryption scheme. Let λ be the security parameter and $\mathcal{A} = (\mathcal{A}_0, \ldots, \mathcal{A}_{q+1})$ and $\mathcal{B} = (\mathcal{B}_0, \ldots, \mathcal{B}_q)$ be tuples of PPT algorithms respectively defining the adversary and the simulator, where q is polynomial in λ. Consider the following probabilistic experiments $\mathbf{Real}_{\mathcal{S},\mathcal{A}}(\lambda)$ and $\mathbf{Sim}_{\mathcal{S},\mathcal{A}}(\lambda)$:

$\mathbf{Real}_{\mathcal{S},\mathcal{A}}(\lambda)$:

$k \leftarrow \mathcal{S}.\text{Gen}(\lambda)$
$(\mathbf{D}, st_{\mathcal{A}}) \leftarrow \mathcal{A}_0(\lambda)$
$(I, \mathbf{C}) \leftarrow \mathcal{S}.\text{Enc}_k(\mathbf{D})$
$(w_1, st_{\mathcal{A}}) \leftarrow \mathcal{A}_1(st_{\mathcal{A}}, I, \mathbf{C})$
$T_1 \leftarrow \mathcal{S}.\text{Trapdoor}_k(w_1)$
for $2 \leq i \leq q$,
$(w_i, st_{\mathcal{A}}) \leftarrow \mathcal{A}_i(st_{\mathcal{A}}, I, \mathbf{C}, T_1, \ldots, T_{i-1})$
$T_i \leftarrow \mathcal{S}.\text{Trapdoor}_k(w_i)$
let $\mathbf{T} = (T_1, \ldots, T_q)$
$b \leftarrow \mathcal{A}_{q+1}(st_{\mathcal{A}}, I, \mathbf{C}, \mathbf{T})$
output b

$\mathbf{Sim}_{\mathcal{S},\mathcal{A},\mathcal{B}}(\lambda)$:

$(\mathbf{D}, st_{\mathcal{A}}) \leftarrow \mathcal{A}_0(\lambda)$
$(I, \mathbf{C}, st_{\mathcal{B}}) \leftarrow \mathcal{B}_0(\mathcal{L}(\mathbf{D}))$
$(w_1, st_{\mathcal{A}}) \leftarrow \mathcal{A}_1(st_{\mathcal{A}}, I, \mathbf{C})$
$(T_1, st_{\mathcal{B}}) \leftarrow \mathcal{B}_1(st_{\mathcal{B}}, \mathcal{L}(\mathbf{D}, (w_1)))$
for $2 \leq i \leq q$,
$(w_i, st_{\mathcal{A}}) \leftarrow \mathcal{A}_i(st_{\mathcal{A}}, I, \mathbf{C}, T_1, \ldots, T_{i-1})$
$(T_i, st_{\mathcal{B}}) \leftarrow \mathcal{B}_i(st_{\mathcal{B}}, \mathcal{L}(\mathbf{D}, (w_1, \ldots, w_i)))$
$b \leftarrow \mathcal{A}_{q+1}(st_{\mathcal{A}}, I, \mathbf{C}, \mathbf{T})$
output b

We say that $\mathcal{S}$ is $\mathcal{L}$-*semantically secure against adaptive attacks* if for all PPT adversaries $\mathcal{A}$ there exists a PPT simulator $\mathcal{B}$ such that

$$|\Pr(\mathbf{Real}_{\mathcal{S},\mathcal{A}}(\lambda) = 1) - \Pr(\mathbf{Sim}_{\mathcal{S},\mathcal{A},\mathcal{B}}(\lambda) = 1)|$$

is negligible, where the probabilities are taken over the random coins of $\mathcal{S}.\text{Gen}$, $\mathcal{S}.\text{Enc}$, $\mathcal{A}$ and $\mathcal{B}$.

To conclude this subsection, we describe the second and most secure searchable encryption scheme introduced by Curtmola, Garay, Kamara and Ostrovsky in [37]. This single-keyword static scheme, while being less efficient than the first scheme in [37], is the first scheme to satisfy the strong notion of adaptive semantic security.

The main insight behind the schemes presented in [37] is the use of an inverted index, which is the conceptual basis for nearly all efficient searchable symmetric encryption solutions. The use of inverted indexes in [37] initiated a recurrent trend in searchable encryption, where cryptographic primitives are used on top of structured data to enable efficient searching over encrypted data.

A *forward index* can be understood as a table, where every row contains a different document D and all keywords $w_1, \ldots, w_n$ attached to it. An *inverse index* can also be understood as a table, where every row contains a different keyword w and all documents $\mathbf{D}(w) = \{D_1, \ldots, D_m\}$ it is attached to. The task of searching for all documents attached to a particular keyword w over a forward index is sequential, and takes linear time $O(|\mathbf{D}|)$ in the number of documents. In contrast, this same searching task takes $O(|\mathbf{D}(w)|)$ time when using inverted indexes, by employing a technique called FKS dictionary [115].

Before stating the definition, we follow [37] and give some notation. Given an ordered set $A = (a_1, \ldots, a_n)$, we denote by $A[i]$ the i-th element a_i of A. Given an element $b \in A$, assuming A has no repeated elements, we denote by $\text{addr}_A(b)$ the subindex i such that $A[i] = b$. Therefore, $A[\text{addr}_A(b)] = b$.

Definition 2.25. Let n be the total number of documents to encrypt and let MAX be the bit size of the largest document. Let W denote the keyword space, and assume that the bit size of the keywords in the document collection to encrypt ranges between m and ℓ, and that it includes a keyword of bit size exactly m. Set $\max = \min\left(\max\{i : \sum_{j=1}^{i} 2^{mj} j < \text{MAX}\}, |W|\right)$ and $s = \max \cdot n$.

Define a SSE scheme $\mathcal{S}$ by the five following polynomial-time algorithms:

$\mathcal{S}.\text{Gen}(\lambda)$**:**

Choose parameters k, ℓ non-constant polynomial in λ. Let W be the keyword space. Let $\mathcal{T} = (\mathcal{T}.\text{Gen}, \mathcal{T}.\text{Enc}, \mathcal{T}.\text{Dec})$ be an IND − PCPA-secure symmetric encryption scheme. Instantiate a PRP π with key space $\{0,1\}^k$

$$\pi : \{0,1\}^k \times \{0,1\}^{\ell + \log_2(n+\max)} \to \{0,1\}^{\ell + \log_2(n+\max)}$$

Sample K_1 uniformly at random from $\{0,1\}^k$ and generate $K_2 = \mathcal{T}.\text{Gen}(\lambda)$. Output the public parameters $\text{params} = \{W, k, \ell, m, \max, n, s, \mathcal{T}, \pi\}$ and the secret key $K = (K_1, K_2)$.

$\mathcal{S}.\text{Enc}_K(\mathbf{D})$**:**

Let $\mathbf{D} = (W_i, D_i)_{i=1}^n$, where W_i is a tuple of keywords attached to the document D_i. For every document D in $\mathbf{D}$, denote by $\text{id}(D)$ a unique identifier of D. Define $\delta(\mathbf{D}) = \cup_i W_i$ as the ordered set of distinct keywords in $\mathbf{D}$. For every $w \in \delta(\mathbf{D})$, let $\mathbf{D}(w)$ denote the ordered set of documents attached to keyword w, in the order induced by $\mathbf{D}$.

Compute $\mathbf{C} = (C_1, \ldots, C_n)$, where $C_i = \mathcal{T}.\text{Enc}_{K_2}(D_i)$.

To prepare the index I, for every $w \in \delta(\mathbf{D})$ and for every $D \in \mathbf{D}(w)$, set

$$I[\pi_{K_1}(w || \text{addr}_{\mathbf{D}(w)}(D))] = \text{id}(D).$$

Now let $s' = \sum_{w \in \delta(\mathbf{D})} |\mathbf{D}(w)|$, and let c_i denote the number of entries of I that contain $\text{id}(D_i)$. For security purposes, if $s' < s$ then modify I as follows: for all documents D_i in $\mathbf{D}$ and for all $1 \leq l \leq \max - c_i$, set $I[\pi_{K_1}(0^\ell \| n + l)] = \text{id}(D_i)$.

Output $I, \mathbf{C}$.

$\mathcal{S}.\text{Trapdoor}_K(w)$**:**

Output $T = (t_1, \ldots, t_n)$, where $t_i = \pi_{K_1}(w \| i)$.

$\mathcal{S}$.Search$(I, \mathbf{C}, T)$**:**
Let X be an empty set. For all $t \in T$ such that $I[t]$ exists, add $\mathbf{C}[I[t]]$ to X. Output X.

$\mathcal{S}$.Dec$_K(C)$**:**
Output $D = \mathcal{T}$.Dec$_{K_2}(C)$.

This searchable encryption scheme is seen to be adaptively semantically secure in [37], under the assumptions that π is a pseudo-random permutation and that $\mathcal{T}$ is IND − PCPA-secure.

2.5.2 Public-Key Searchable Encryption with Keyword Search

In searchable encryption there potentially exist two kinds of clients: readers and writers. Readers are able to generate trapdoors and search over the outsourced data, and writers can generate searchable indexes. While in searchable symmetric encryption one assumes that a single user fills both roles, that may not always be the case, and writers may not necessarily be authorized to search over the outsourced data. Moreover, one can consider architectures where there may exist multiple readers or writers.

Public key searchable encryption (also named *public key encryption with keyword search*), or PEKS, realizes searchable encryption in the public-key setting, where there may exist multiple writers that send data over to a single reader. PEKS was firstly proposed by Boneh et al. in [38]. Since their pioneering work, there have appeared several PEKS schemes in the literature [40–43, 47, 48, 65, 74, 116–118], improving the scheme in [38] in terms of efficiency, security or functionality.

The keyword search protocol usually considered in PEKS involves the following entities:

- a set of *data suppliers*, which provides and encrypts the data to be outsourced,
- a storage *server* (e.g. an e-mail gateway or a database), which stores the outsourced data, and
- a *client* of the storage server, who retains the ability to generate queries for the encrypted data.

In the protocol, the client firstly sets up the scheme by generating some public parameters and shares them with the server and the data suppliers. It generates a public and private key pair, shares the public key with the data suppliers and keeps the private key secret. Afterwards, the data suppliers may wish to share a collection of data items with the client, where each of them is indexed by a set of keywords. To do so, they encrypt this collection, and they upload the resulting ciphertexts to the storage server. Afterwards, the client may want to receive from the storage server all the stored data items indexed by keywords in some chosen list. To let the storage server know which encrypted data it should forward, the client can generate a query and send it to the storage server. The storage server is then able to use the received information to select the encrypted stored data items satisfying the query conditions, and it can return those data items to the client. Note that this process can be carried out without direct interaction between the client and the data providers.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

We now give a general model for public-key searchable encryption schemes. Although not stated, every algorithm apart from $\mathcal{S}$.Gen takes the public parameters as input.

Definition 2.26. We define a PEKS scheme $\mathcal{S}$ as consisting of four polynomial-time algorithms:

$\mathcal{S}.\text{Gen}(\lambda)$**:** Probabilistic algorithm run by the client that, given a security parameter λ, returns the private key sk, the public key pk and the public parameters params of the scheme.

$\mathcal{S}.\text{Enc}_{\text{pk}}(\mathbf{D})$**:** Probabilistic algorithm run by data providers. It takes as input a document $\mathbf{D}$, which consists of a tuple of keywords. It returns a corresponding searchable index $\mathbf{I}$.

$\mathcal{S}.\text{Trapdoor}_{\text{sk}}(f)$**:** Algorithm run by the client that takes as input a predicate f on keywords. It returns a corresponding trapdoor $\mathbf{T}$.

$\mathcal{S}.\text{Search}(\mathbf{I}, \mathbf{T})$**:** Deterministic algorithm run by the server and taking as input a searchable index $\mathbf{I}$ and a trapdoor $\mathbf{T}$. It returns either 1 or 0.

Note that our usage of the term *document* here drops the data items (e.g., files, e-mails) and considers only the indexing keywords. This is so because PEKS works exclusively over the indexing keywords, and data items may be protected by other cryptographic means (as explained in the following paragraphs).

Just as in searchable symmetric encryption schemes, PEKS schemes can be classified according to query expressiveness, i.e., according to the class of predicates allowed by the $\mathcal{S}$.Trapdoor algorithm. If the algorithm $\mathcal{S}$.Trapdoor supports only single keywords as predicates, we say that $\mathcal{S}$ is a *single-keyword* SSE scheme. If $\mathcal{S}$.Trapdoor supports arbitrary conjunctions of such predicates, we say that $\mathcal{S}$ is a *conjunctive* SSE scheme.

In most of the proposed applications for single-keyword PEKS, the exchanged ciphertexts take the form

$$\text{Enc}_{\text{pk}}(D) \| \mathbf{I}(w_1) \| \cdots \| \mathbf{I}(w_m),$$

where pk is the public key of the client, Enc is some public-key encryption scheme, the data item D is indexed by keywords $w_1, \ldots, w_m$, and $\mathbf{I}(w_i) = \mathcal{S}.\text{Enc}_{\text{pk}}(w_i)$ are the searchable indexes corresponding to each of the indexing keywords. In a cloud computing setting, this concatenation of ciphertexts is uploaded to the cloud server by the data suppliers. The client can recover all data items indexed with keyword w at *position* (or *field*) $i \in \{1, \ldots, m\}$ by sending the trapdoor $\mathbf{T}(w)$ and the position i to the server.

Following this last strategy, the PEKS literature usually achieves conjunctive queries by using keyword fields. This is done by setting up the $\mathcal{S}$.Trapdoor algorithm so that it admits as input a tuple $\mathbf{L} = (w'_1, \ldots, w'_l)$ of keywords and a set $J = \{j_1, \ldots, j_l\}$ of positions. Then, a data item $\mathbf{D} = \{w_1, \ldots, w_n\}$ satisfies such a conjunctive field keyword query if $w_{j_i} = w'_i$ for all $i \in [l]$. This construction is properly defined in Section 3.2.

In order to define correctness for PEKS schemes, we recur to the notions of consistency defined by Abdalla et al. [119]. Consistency in PEKS refers to the property that a searchable index and a trapdoor should match in the search process exactly when the underlying document and query also match. The consistency notions defined in [119] are, in increasing strength order, *computational*, *statistical* and *perfect*. In this thesis we

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

use the computational consistency definition. Informally, this definition states that the advantage of any polynomial-time adversary in finding a matching searchable index and trapdoor coming from a non-matching document and query is negligible in the security parameter, where the adversary has access to the public parameters and to the public key.

We now present the Computational Consistency definition. Let $\mathcal{S}$ be a PEKS scheme. Given a security parameter λ, we introduce a consistency game in the following three phases:

- *Setup.* The challenger runs $\mathcal{S}$.Gen on input λ, hands over the public parameters and the public key pk to the adversary, and keeps the private key sk secret.
- *Guess.* The adversary outputs a document of the form $\mathbf{D} = (w_1, \ldots, w_m)$ and a predicate f on keywords.
- *Output.* The challenger hands over adversary the trapdoor $\mathbf{T} = \mathcal{S}.\text{Trapdoor}_{\text{sk}}(f)$, and the adversary computes $\mathbf{I} = \mathcal{S}.\text{Enc}_{\text{pk}}(\mathbf{D})$. If $\mathcal{S}.\text{Search}(\mathbf{I}, \mathbf{T}) = 1$ and if predicate f does not match document $\mathbf{D}$, then the adversary outputs a bit $b = 1$. Otherwise, it outputs $b = 0$.

Definition 2.27 (Computational Consistency of PEKS [119])**.** A PEKS scheme $\mathcal{S}$ is *computationally consistent* if the advantage of every PPT adversary $\mathcal{A}$ in the above game

$$\text{Adv}_{\mathcal{A}}(\lambda) = \Pr(b = 1)$$

is negligible in λ.

We next introduce the Semantic Security against Adaptive Chosen Keyword Attacks definition presented by Boneh et al. in [38]. This security definition states that indexes $\mathbf{I}(w)$ do not reveal any information about the underlying keyword w unless the trapdoor $\mathbf{T}(w)$ corresponding to this keyword is available. In a cloud computing setting, this means that that the server does not learn any information from the searchable indexes unless it has the knowledge of a matching trapdoor.

Given a security parameter λ, define a security game in the following five phases:

- *Setup.* The challenger runs $\mathcal{S}$.Gen on input λ and hands over the public parameters to the adversary.
- *Query Phase 1.* The adversary adaptively requests the challenger for encrypted indexes $\mathcal{S}.\text{Enc}_k(w)$ for a polynomial number of keywords w of its own choice.
- *Challenge.* The adversary outputs two challenge candidate keywords w_0, w_1. The challenger throws a fair coin $b \in \{0, 1\}$, and outputs the encrypted index $I_b = \mathcal{S}.\text{Enc}_{\text{k}}(m_b)$ corresponding to keyword w_b.
- *Query Phase 2.* The adversary proceeds just as in Query Phase 1.
- *Guess.* The adversary outputs a guess $b' \in \{0, 1\}$ for b.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Definition 2.28 (Semantic Security against Adaptive Chosen Keyword Attacks [38])**.** The PEKS scheme $\mathcal{S}$ is *semantically secure against adaptive chosen keyword attacks* if the advantage of every PPT algorithm $\mathcal{A}$ in breaking the above game

$$\mathrm{Adv}_{\mathcal{A}}(\lambda) = \left|\Pr(b' = b) - 1/2\right|$$

is negligible in λ.

To conclude this subsection, we describe the PEKS scheme introduced by Boneh et al. in [38]. This scheme is a single-keyword scheme that satisfies the previous security definition under the Bilinear Diffie Hellman computational hardness assumption [103, 120] in the Random Oracle Model.

Definition 2.29. Define a PEKS scheme $\mathcal{S}$ by the four following polynomial-time algorithms:

$\mathcal{S}.\mathrm{Gen}(\lambda)$**:** Given a security parameter $\lambda \in \mathbb{Z}$, fix a symmetric bilinear group $\mathbb{G}$ of prime order $q \geq 2^{\lambda}$ and denote the corresponding pairing by $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Let g be a random generator of $\mathbb{G}$. Let $H_1 : \{0,1\}^* \to \mathbb{G}$ and $H_2 : G_T \to \{0,1\}^{\log q}$ be collision-resistant hash functions. Choose $\alpha \in \mathbb{F}_q^*$ uniformly at random. Output the public parameters $\mathrm{params} = \{\mathbb{G}, \mathbb{G}_T, q, e, g, H_1, H_2\}$, the private key $\mathrm{pk} = g^{\alpha}$ and the public key $\mathrm{sk} = \alpha$.

$\mathcal{S}.\mathrm{Enc}_{\mathrm{pk}}(w)$**:** Given as input a document consisting of a single keyword w, compute $t = e(H_1(w), \mathrm{pk}^r)$ for a random $r \in \mathbb{F}_q^*$. Return the encrypted index $\mathbf{I}(w) = (g^r, H_2(t))$.

$\mathcal{S}.\mathrm{Trapdoor}_{\mathrm{sk}}(w)$**:** Given as input a tuple of keywords consisting of a single keyword w, return a corresponding trapdoor $\mathbf{T}(w) = H_1(w)^{\mathrm{sk}}$.

$\mathcal{S}.\mathrm{Search}(\mathbf{I}, \mathbf{T})$**:** Given an encrypted index $\mathbf{I} = (A, B)$ and a trapdoor $\mathbf{T}$, return 1 if $H_2(e(\mathbf{T}, A)) = B$. Otherwise, return 0.

2.6 Secret Sharing

In this section we give an account of unconditionally secure secret sharing. We formally define some of the terms mentioned in Section 1.4.4, such as access structure, distribution scheme and secret sharing scheme, and we present the complexity measures analyzed in this thesis. For a low-level discussion of secret sharing, see Section 1.4.4 and Chapter 5. For an introduction to secret sharing, see for instance [121, 122].

As explained in Section 1.4.4, secret sharing schemes aim at protecting a secret piece of information by dividing it into shares, in such a way that the secret can only be reconstructed from some combinations of shares. The collection of combinations of shares that allow the reconstruction of the secret form an *access structure*. We now formally define this term.

Definition 2.30 (Access Structure)**.** Let P be a set. A collection $\Gamma \subseteq \mathcal{P}(P)$ is *monotone* if $B \in \Gamma$ and $B \subseteq C \subseteq P$ implies $C \in \Gamma$. An *access structure* is a monotone collection $\Gamma \subseteq \mathcal{P}(P)$ of non-empty subsets of P. The family of minimal subsets in Γ is denoted by $\min \Gamma$.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Secret sharing schemes can be used to distribute a secret among a set of participants, in such a way that the secret can only be recovered when some participants pool together their shares. The distribution phase in this usage is formalized by the notion of distribution scheme.

Definition 2.31 (Distribution Scheme)**.** Let $P = \{1, \ldots, n\}$ and let K be a finite set. A *distribution scheme* on P with domain of secrets K is a pair $\Sigma = (\Pi, \mu)$, where μ is a probability distribution on a finite set R, and Π is a mapping from $K \times R$ to a set of n-tuples $K_1 \times K_2 \times \cdots \times K_n$. The set R is called *the set of random strings* and K_j is called the *domain of shares* of j.

For a distribution scheme (Π, μ) and for any $A \subseteq P$, we denote by $\Pi(s, r)_A$ the projection of $\Pi(s, r)$ to its A-entries. We next formally define secret sharing schemes as distribution schemes with some added reconstruction and confidentiality conditions. This definition is taken from [121].

Definition 2.32 (Secret Sharing)**.** Let K be a finite set of secrets with $|K| \geq 2$. A distribution scheme (Π, μ) on P with domain of secrets K is a *secret-sharing scheme* realizing an access structure Γ if the following two requirements hold for every $A \subseteq P$:

- If $A \in \Gamma$, then there exists a *reconstruction function* $\mathrm{Recon}_A : K_{i_1} \times \ldots \times K_{i_r} \to K$ such that, for every $k \in K$,

$$\Pr\left(\mathrm{Recon}_A(\Pi(k, r)_A) = k \right) = 1.$$

- If $A \notin \Gamma$ then, for every $a, b \in K$ and for every possible vector of shares $(s_j)_{j \in A}$,

$$\Pr\left(\Pi(a, r)_A = (s_j)_{j \in A} \right) = \Pr\left(\Pi(b, r)_A = (s_j)_{j \in A} \right).$$

In a secret sharing scheme, we usually consider that there is an additional player p_0 not in P called the *dealer*. In order to distribute a secret $k \in K$ according to Σ, the dealer first samples a random string $r \in R$ according to μ, it computes a vector of *shares* $\Pi(k, r) = (s_1, \ldots, s_n)$, and it privately communicates each share s_j to party j. The subsets of participants in P satisfying the first condition in the last definition are called *authorized*, and the ones satisfying the second condition are called *forbidden*. In this work we consider only *perfect* secret sharing schemes, that is, schemes in which every subset of participants is either authorized or forbidden.

Among secret sharing schemes, we highlight *linear* secret sharing schemes. These schemes are of special interest, since their linearity is useful for applications in other cryptographic solutions such as attribute-based encryption schemes [24], secure multiparty computation schemes or joint signature schemes.

Definition 2.33 (Linear Secret Sharing Scheme)**.** Let $\mathbb{F}$ be a finite field. A secret sharing scheme $\Sigma = (\Pi, \mu)$ is $(\mathbb{F}, \ell)$-*linear* if $K = \mathbb{F}^\ell$, the sets R, K_1, ..., K_n are vector spaces over $\mathbb{F}$, μ is the uniform distribution on R, and Π is a surjective linear mapping.

As stated in Section 1.4.4, we can associate various complexity measures to secret sharing schemes and to access structures. We next define the complexity measures analyzed in our research.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

For a secret sharing scheme Σ on P, the *information ratio* of Σ is defined as

$$\sigma(\Sigma) = \frac{\max_{1\leq j\leq n} \log|K_j|}{\log|K|},$$

and the *total information ratio* of Σ is defined as

$$\sigma^{\mathrm{T}}(\Sigma) = \frac{\sum_{1\leq j\leq n} \log|K_j|}{\log|K|}.$$

We say that Σ is *ideal* if $\sigma(\Sigma) = 1$. If an access structure Γ admits an ideal secret sharing scheme, we also say that Γ is *ideal*.

For an access structure Γ, we define the *optimal information ratio* $\sigma(\Gamma)$ as the infimum of the information ratio of secret sharing schemes for Γ. Also, we define the *optimal total information ratio* $\sigma^{\mathrm{T}}(\Gamma)$ as the infimum of the total information ratio of the secret sharing schemes for Γ. Analogously, for every prime power q we define $\lambda_{q,\ell}(\Gamma)$ and $\lambda^{\mathrm{T}}_{q,\ell}(\Gamma)$ as the infimum of the information ratios and total information ratios of the $(\mathbb{F}_q, \ell)$-linear secret sharing schemes for Γ, respectively.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Chapter 3

Searchable Encryption

In this chapter, we present our contributions to the field of searchable encryption. This chapter is divided in two independent sections, each containing a separate work.

In Section 3.1, we present our work on two-dimensional range queries over encrypted data. This work is set in the field of Symmetric Searchable Encryption (SSE), which aims to enable efficient search over outsourced encrypted data in a single-user architecture. In this section, we present techniques that enable secure and efficient two-dimensional range queries in SSE.

In Section 3.2, we present our work on Public Key Encryption with Keyword Search (PEKS). PEKS schemes are searchable encryption schemes that enable public key holders to encrypt documents, while the secret key holder is able to generate queries for the encrypted data. In this work, we build two schemes achieving conjunctive and subset queries respectively. We show that our schemes outperform existing ones in terms of efficiency and expressiveness.

3.1 Two-Dimensional Range Queries Over Encrypted Data

3.1.1 Introduction

The main problem we deal with in this section is searching over encrypted data. In particular, we consider the case in which we want to perform queries on an encrypted two-dimensional data set. A possible motivation would be the natural issue of users owning data sets containing spatial geo-referenced data, for instance data about geologic resources, water supplies, critical infrastructures or civil constructions. Typically, this information is stored and indexed locally, but lately, due to economical and practical reasons, data is often outsourced to the cloud. Since the information may be valuable and confidential, users may want to prevent the cloud server and any external attackers from obtaining information about it.

A simple approach for secure storage is to encrypt all the data in a straightforward way, using symmetric-key encryption schemes. However, very few functionalities are preserved when using general encryption techniques, and in particular it is hard to search or to compute on the outsourced data efficiently. Hence, we consider a solution

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

based on *searchable encryption* (SE), a cryptographic primitive that allows to remotely query on outsourced encrypted data in a secure way. Song et al. [25] presented one of the first SE schemes in 2000. Since their pioneering work, the field has expanded in many directions, by enhancing the security and efficiency of the schemes, by improving expressiveness of the queries, and by dealing with multi-user settings.

We are interested in using SE schemes in a client-server architecture. In this setting, the client first creates an index of the data according to certain indexing keywords and stores it along with the data set in an encrypted form. Later, if the client wants to recover the elements of the encrypted data set that are indexed by some chosen keywords, it sends a token associated to these keywords to the server. The server is then able to efficiently retrieve and return the corresponding encrypted documents while learning as little information as possible. We assume that the server behaves in an honest-but-curious manner, that is, it always runs the protocol correctly, but it collects additional information and is able to perform computations using the data it already holds.

Recent works [64–72] present extensions of existing SE schemes that achieve secure and efficient range queries. These solutions generally exploit a special indexation of documents. However, in all these solutions, achieving range queries has an impact on computational and communication costs. Our work here is dedicated to the study of similar techniques for range queries in SE, focusing on two-dimensional range queries. We employ two techniques: *quadtrees* and *over-covers*.

We introduce quadtrees as a new technique for SE that aims to reduce the leakage of information and the communication and computational costs of two-dimensional range queries. Quadtrees induce a decomposition of the two-dimensional grid into quadrants. They allow to improve security by reducing leakage with respect to alternative constructions [66] in the case of two-dimensional data. This technique can also be extended to other multi-dimensional range queries.

In order to reduce the communication costs, Faber et al. [68] introduced the concept of over-covers. By enlarging the queried ranges, over-covers reduce the communication costs and information leakage associated to the scheme, but they induce false positives in the search results. However, the ranges are chosen in such a way that the information sent to the server is reduced. For instance, queries that use the over-covers in [68] always result in three keyword queries, and they have a false-positive rate of at most 66%.

In this work we analyze the use of over-covers, generalizing the constructions in [68] by providing over-covers with a greater number of intervals. Using these over-covers the information sent to the server is still small. Moreover, the accuracy is substantially increased when increasing the number of elements in the over-cover.

Due to restrictions of the setting, we study symmetric-key cryptographic schemes. We provide general techniques that can be used for a broad family of schemes, the ones supporting at least single keyword searches. Moreover, we also provide techniques that take advantage of the properties of more advanced schemes supporting Boolean queries, like the one by Cash et al. [93]. Recently, Faber et al. [68] presented an extension of the scheme in [93] that supports range queries. Following [68], we provide an adaptation of their scheme for two-dimensional range queries. In the public-key setting, there exist schemes supporting range queries [65, 66], but these schemes do not fit our setting because we aim at dealing with the one writer and one reader scenario.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

We note here that very recent works study the cryptanalysis of SE schemes [26–32], exposing attacks that exploit the leakage of SE schemes. Some of these attacks, particularly [30], specialize to the case of range queries. Assuming that the data set is dense and that enough uniformly distributed queries are issued, they allow attackers to reveal all the queried and data set values by using only the leaked access pattern information. To the best of our knowledge, up to date no range SE schemes that avoid this and similar attacks have been proposed, and the solutions proposed in Chapter 3.1 are also vulnerable if the premises stated in [30] hold. The easiest way to prevent such attacks is to apply a PIR scheme instead of a SE scheme in our constructions. A scheme based on binary trees such as the one by Lipmaa [123] would be most suitable to our constructions. Applying PIR as in [124] would effectively achieve query privacy and prevent the attacks described in [30], although it would have an impact both on efficiency and on the security of the outsourced data set.

In Section 3.1.2 we give an introduction to searchable encryption, providing a general description of the primitives involved in our constructions. In Section 3.1.3 we suggest different representations of two-dimensional data, and we provide our two-dimensional range searchable encryption constructions in Sections 3.1.5, 3.1.6, 3.1.7, 3.1.8. The efficiency and the information leakage associated to the proposed constructions are described in Sections 3.1.9 and 3.1.10. Then, some of the proofs are deferred to Section 3.1.11. Finally, the conclusions and future work are sketched in Section 3.1.12.

3.1.2 Problem Definition

3.1.2.1 Searchable Symmetric Encryption

The two-dimensional data-oriented solutions presented in this thesis make a black-box use of an underlying *searchable symmetric encryption* (SSE) scheme. Searchable symmetric encryption schemes are SE schemes set up in a single-client architecture, and they were principally developed by Curtmola et al. in [37]. Such schemes allow a client to outsource an encrypted version of a data set to a semi-trusted, honest-but-curious server, while preserving basic searching capabilities over the encrypted data.

Consider the one writer and one reader setting, i.e. a single-client architecture. Firstly, a single client wants to upload a data set $\mathbf{D}$ to a server. The data set is composed of tuples (W_i, D_i), where the *document* D_i is any kind of plaintext (e.g. text, audio, images) and W_i is a *list of keywords* associated D_i.

A searchable symmetric encryption (SSE) scheme $\mathcal{S}$ first provides a client with a symmetric key k by executing an algorithm $\mathcal{S}$.Gen. Using this key k, an algorithm $\mathcal{S}$.Enc allows the client to build an encryption $\mathbf{C}$ of the data set $\mathbf{D}$. Additionally, this algorithm creates a *searchable index* I in the process, which is uploaded to the server along with the ciphertexts $\mathbf{C}$.

Afterwards, the client may want to retrieve the documents satisfying a particular predicate f on keywords. A predicate is specified by a Boolean formula on a tuple of keywords. For example, the predicate $w_1 \wedge (w_2 \vee w_3)$ is satisfied by documents whose associated keyword list contains the keyword w_1 and either w_2 or w_3. The client can then furnish a predicate f to the algorithm $\mathcal{S}$.Trapdoor to generate a *trapdoor* (or *search token*) T

encoding the given predicate. To query for documents satisfying predicate f, the client sends T to the server.

By combining the trapdoor T and the searchable index I and by using the public algorithm $\mathcal{S}$.Search, the server is able to obtain a tuple of ciphertexts which are the encrypted documents satisfying the predicate that T represents.

Refer to Definition 2.18 for a general model for searchable symmetric encryption schemes. For an introduction to searchable encryption, see [109] and Section 2.5.1.

From all searchable symmetric encryption schemes, we choose the OXT scheme by Cash et al. [93] as the underlying scheme for the instantiations of our proposed schemes. The OXT scheme is suitable for static data sets and it provides highly efficient Boolean search while incurring moderate and quantifiable leakage to the server. Regardless of this choice, we provide general techniques that can also be used with any searchable encryption scheme, even if it does not support conjunctive or Boolean queries.

As of security, queries using searchable encryption induce some leakage on the information that is exchanged or stored. Several works define different models of leakage, depending on the scheme and on the followed adversary model. The notion of security of some SSE schemes is conditioned to a predefined *leakage function* $\mathcal{L}$. In [93], Cash et al. define the concept of $\mathcal{L}$-semantic security against adaptive attacks (which is stated in Section 2.5.1), and in this work we base our security analysis on their security definition. Further, in [93, Section 5.3], they show precise upper bounds on the allowed leakage of the OXT scheme. This enables us to provide a clean security analysis in Section 3.1.9 by describing upper bounds on the leakage incurred by each instantiation of our proposed schemes.

3.1.3 Data Structure for Range Queries

In this section we present the structure for two-dimensional data considered in this work. We use binary tree and quadtree structures in order to describe the geometrical objects used in the proposed schemes. Next, we give the general model for 2-dimensional range searchable symmetric encryption schemes.

3.1.3.1 Data Model

In our model, the client owns a collection of documents, each of which is attached to a particular tuple of values. Without loss of generality, we assume that such tuples always belong to the two-dimensional discrete grid $\Lambda_n = \{0, \ldots, 2^{n+1} - 1\} \times \{0, \ldots, 2^{n+1} - 1\}$ for some fixed bit length $n + 1 > 0$.

We call the tuples $(a, b) \in \Lambda_n$ *points*. By an abuse of notation, we also denote points in Λ_n by $(a_n \cdots a_0, b_n \cdots b_0)$, where $a_n \cdots a_0, b_n \cdots b_0 \in \{0,1\}^{n+1}$ are the *bit representations* of a and b. For instance, we can consider the point $(1001, 0100) \in \Lambda_3$.

We restrict our model to the case where each document is attached to a single point in Λ_n. Since all shapes in Λ_n can be represented as some set of points, the constructions naturally extend to cases where documents are attached to lines and polygons. We further comment this extension in Section 3.1.12.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Let $\mathcal{I}_n = \{[a,b] : 0 \leq a \leq b < 2^{n+1}, a, b \in \mathbb{Z}\}$ denote the set of *intervals* (or *ranges*) with only $n+1$-bit values. Here, $[a,b]$ denotes the discrete interval $\{a, a+1, \ldots, b\}$. We call the elements in $\mathcal{I}_n^2 = \mathcal{I}_n \times \mathcal{I}_n$ *rectangles*. For instance, the rectangle $[0100, 1011] \times [1000, 1111] \in \mathcal{I}_3^2$ is a product of two intervals of length 8.

In our setting, the client first delegates an encrypted version of its data set to a server. Afterwards, it may want to issue a query to the server to retrieve the subset of the outsourced data set whose associated points fall inside a two-dimensional range of values. That is, the client may want to query the server for the set of documents whose associated points lie inside a rectangle of its own choice.

3.1.3.2 Binary Trees and Prefixes

Binary trees have been widely applied in computer science, and applications range from routing protocols to sorting and text searching. In the context of searchable encryption, they have been used to achieve security and efficiency in range queries over encrypted data in works such as [66, 70] and, more recently, in [68, 71].

Let $0 \leq a < 2^{n+1}$ be an integer. Then a can be represented with $n+1$ bits as $a_n \cdots a_0$. It is clear that every such a can be viewed as a leaf of the perfect binary tree of height $n+1$, so that the parent of a at depth i is associated to the i most significant bits of a. In this way, we denote the parent of a at depth $i \leq n$ with the binary representation $a_n \cdots a_{n-i+1}*$, where the wildcard $*$ denotes an arbitrary suffix. We call such representation a *prefix* of *length* i.

We define the *parent tuple* of a non-negative integer a with bit representation $a_n \cdots a_0$ as the tuple of prefixes given by the parents of a in increasing depth, that is,

$$(a_n*,\ a_n a_{n-1}*,\ \ldots,\ a_n a_{n-1} \cdots a_1*,\ a_n a_{n-1} \cdots a_1 a_0).$$

For instance, the parent tuple of 0110 is $(0*, 01*, 011*, 0110)$. We use this tuple to generate the index element associated to a point, as in related works [66, 68].

Any prefix $a_n \cdots a_{n-i+1}*$ of length i defines a *prefix range* of length 2^{n-i+1}

$$\{a_n \cdots a_{n-i+1} b_{n-i} \cdots b_0 \ :\ b_{n-i} \cdots b_0 \in \{0,1\}^{n-i+1}\}.$$

We say that any given interval is a prefix range if it is expressible as a prefix. For example, in a 4-bit field, the prefix $01*$ defines the prefix range $[0100, 0111]$.

3.1.3.3 Quadtrees and Prefixes

As we see in Section 3.1.9 or in works such as [66, 68], the usage of binary trees in 2-dimensional range searchable encryption induces substantial data leakage. The main reason behind this is that the plaintext documents use a binary tree for each coordinate, so the dimensions are decoupled. In an attempt to fix this, we develop the same concepts above by using *quadtrees* instead of binary trees.

We consider *quadtrees* to be perfect 4-ary trees. Most common uses of quadtrees involve two-dimensional data, such as in image representation, spatial indexing or storing matrix

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

information. Geometrically, they are usually interpreted as representing a recursive subdivision of space into quadrants.

Let (a, b) be a point in Λ_n with binary representation $(a_n \cdots a_0, b_n \cdots b_0)$. Then (a, b) can be viewed as a leaf of a quadtree of height $n + 1$, in such a way that the parent of (a, b) at depth i is associated to both the i most significant bits of a and the i most significant bits of b. In this way, we denote the parent of (a, b) at depth $i \leq n$ with the binary representation $(a_n a_{n-1} \cdots a_{n-i+1}*, b_n b_{n-1} \cdots b_{n-i+1}*)$, where the wildcard $*$ denotes an arbitrary suffix. We call such representation a *prefix* of *length* $2i$.

The parent tuple of the point $(a_n \cdots a_0, b_n \cdots b_0)$ is defined just as in the case of binary trees. For instance, the parent tuple of $(001, 100)$ is $((0*, 1*), (00*, 10*), (001, 100))$.

We say that a rectangle $[a, b] \times [c, d]$ is a *prefix rectangle* if it is expressible as a prefix. This happens exactly when both $[a, b]$ and $[c, d]$ are prefix ranges expressible as prefixes of the same length. For example, when using 6-bit fields, the prefix $(110*, 011*)$ defines the prefix rectangle $[110000, 110111] \times [011000, 011111]$.

3.1.3.4 Prefix Decomposition

Given an arbitrary interval, consider an expression of it as a disjoint union of some set of prefix ranges. We call the set of corresponding prefixes a *prefix decomposition* of the interval. For example, the interval $[0011, 1101]$ admits a decomposition into 4 prefixes, namely $\{0011, 01*, 10*, 110*\}$. The *profile* of a prefix decomposition (as defined in [68]) is the set of lengths of each of its prefixes, counting multiplicity. For example, the profile of the prefix decomposition $\{0011, 01*, 10*, 110*\}$ is $\{4, 2, 2, 3\}$.

Any rectangle can be expressed as a disjoint union of some prefix rectangles. We call the set of corresponding prefixes a *prefix decomposition* of the rectangle. For example, the rectangle $[00, 01] \times [00, 10]$ admits a decomposition into 3 prefixes, namely $\{(0*, 0*), (00, 10), (01, 10)\}$. In Section 3.1.6 we state an upper bound on the minimal size of such prefix decompositions.

As observed in [66, Section 5.1] for binary trees, an element belongs to an interval if and only if the intersection between the parent tuple of the element and any prefix decomposition of the interval is nonempty. Similarly in quadtrees, the parent tuple of a point consists exactly of the prefixes whose prefix rectangle contains that point. This is why a point belongs to a rectangle if and only if the intersection between the parent tuple of the point and any prefix decomposition of the rectangle is nonempty. As a result of this property, in Sections 3.1.5, 3.1.6 we use prefix decomposition in order to generate queries associated to rectangles.

3.1.3.5 Over-Covers

Given an arbitrary interval, consider a set of prefix ranges whose union contains the interval (possibly strictly). The set of corresponding prefixes is called an *over-cover* of the given interval. This notion was first defined in [68]. The *profile* of an over-cover is defined just as for prefix decompositions.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

As an example, the interval $[0011, 1101]$ admits the over-cover $\{0011, 01*, 1*\}$, which has profile $\{4, 2, 1\}$. Note that this over-cover has size 3, even though the smallest prefix decomposition the interval admits is $\{0011, 01*, 10*, 110*\}$ and has size 4.

Given an arbitrary rectangle, consider a set of prefix rectangles whose union contains the rectangle. Just as in the case of intervals, we call the set of corresponding prefixes an *over-cover* of the rectangle. For example, the rectangle $[00, 01] \times [00, 10]$ admits an over-cover $\{(0*, 0*), (0*, 1*)\}$.

As observed in [66, Section 5.1] for binary trees, if an element belongs to an interval, the intersection between the parent tuple of the element and any over-cover of the interval is nonempty. Similarly for quadtrees, if a point belongs to a rectangle, then the intersection between the parent tuple of the point and any over-cover of the rectangle is nonempty. As a result of this property, in Sections 3.1.7, 3.1.8 we use prefix over-covers in order to generate a query associated to a rectangle.

3.1.4 Searchable Encryption for Two-Dimensional Data

We start by giving context for our two-dimensional data-oriented solutions. Our aim is to provide schemes that enable a client to delegate an encrypted version of a data set to a semi-trusted, honest-but-curious server, in such a way that searching capabilities over the encrypted data are preserved.

Assume that a client wants to upload a data set $\mathbf{D}$ to a server. The data set is composed of tuples $((w_{i,1}, w_{i,2}), D_i)$, where the document D_i is any kind of plaintext (text, audio, images, etc.) and $(w_{i,1}, w_{i,2})$ is the point in Λ_n where D_i is *located*.

A 2-dimensional range searchable symmetric encryption scheme $\mathcal{R}$ first provides a client with a symmetric key k by executing an algorithm $\mathcal{R}$.Gen. Using this key k, an algorithm $\mathcal{R}$.Enc allows the client to build an encryption $\mathbf{C}$ of the data set $\mathbf{D}$. Additionally, this algorithm creates a *searchable index* I in the process, which is uploaded to the server along with the ciphertexts $\mathbf{C}$.

Afterwards, the client may want to retrieve the documents located in a rectangle $R \in \mathcal{I}_n^2$. The client can furnish this rectangle to the algorithm $\mathcal{R}$.Trapdoor to generate a *trapdoor* (or *search token*) T that encodes the given rectangle. To query for rectangle R, the client sends T to the server.

The server is able to combine a trapdoor T and the searchable index I by using the public algorithm $\mathcal{R}$.Search. In this way, it obtains a tuple of ciphertexts, which correspond to all documents located in the rectangle R.

We now give the formal definition of 2-dimensional range searchable symmetric encryption scheme.

Definition 3.1. A 2-dimensional range searchable symmetric encryption (2-DRSSE) scheme $\mathcal{R}$ is a SSE scheme where

1. The output of $\mathcal{R}.\mathrm{Gen}(\lambda)$ includes a bit-length $n + 1$.
2. The input of $\mathcal{R}.\mathrm{Enc}_k$ is a collection $\mathbf{D}$ of plaintext documents, where each document consists of some plaintext (e.g., the id of a file) and an attached point in Λ_n. That is, $\mathbf{D} = ((w_{i,1}, w_{i,2}), D_i)_i$.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

3. The input of $\mathcal{R}.\text{Trapdoor}_k$ is a rectangle $R \in \mathcal{I}_n^2$, seen as the disjunction of all points of R.

In this work, we use a SSE scheme $\mathcal{S}$ as a primitive for building 2-DRSSE schemes. The solutions, presented sequentially in the next sections, provide different trade-offs between security and efficiency. They use the following techniques:

- Prefix decompositions and binary trees: initial approach with remarkable efficiency.
- Prefix decompositions and quadtrees: sacrifices efficiency in order to achieve a smaller leakage.
- Over-covers and binary trees: induces false positives, but improves on both efficiency and security.
- Over-covers and quadtrees: induces false positives, and sacrifices efficiency with respect to the previous solution in order to achieve even smaller leakage.

Throughout the rest of the section, we fix a positive bit length $n+1$, so that all documents are located at points of Λ_n.

3.1.5 Prefix Decomposition Using Binary Trees

In this section we present our first 2-DRSSE scheme. We first state the required results, and we proceed by describing the construction.

Li and Omiecinski [70] proved that, for $n \geq 2$, any interval in $\mathcal{I}_n$ admits a prefix decomposition using at most $2n$ prefixes. We present an extension of this result. First, we need the following lemma.

Lemma 3.2. *Let t be a positive integer and let $[a, b]$ be an interval of length at most 2^t such that*

1. *the least t significant bits of a are zero, or*
2. *the least t significant bits of b are one.*

Then, $[a, b]$ decomposes into at most $\max(1, t)$ prefixes.

Proof. Set $a = a_n \cdots a_0$, $b = b_n \cdots b_0$ the bit representation of a, b.

We prove this lemma by induction on t. The result holds trivially for $t = 0, 1$. Now assume $t > 0$ and consider the case 1. Then $a = a_n \cdots a_t 0 \cdots 0$ and $b = b_n \cdots b_0$ with $b_i = a_i$ for $i = t \ldots n$. If $b_{t-1} = 0$ then $[a, b]$ decomposes into at most t prefixes by hypothesis. Otherwise, $[a, b]$ splits into $[a, a']$ and $[b', b]$, where

$$a' = a_n \cdots a_t 01 \cdots 1 \quad b' = a_n \cdots a_t 10 \cdots 0$$

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Since $[a, a']$ is a prefix, the case 1 follows by hypothesis. The case 2 is proved analogously. □

The following theorem extends the result in [70] so that the bound on the number of prefixes depends on the length of the given interval rather than on the underlying bit length.

Theorem 3.3. *Let t be a positive integer. Then, any interval $[a, b]$ of length at most 2^t decomposes into at most $\max(t+1, 2t-1)$ prefixes.*

Proof. Set $a = a_n \cdots a_0$, $b = b_n \cdots b_0$ the bit representations of a, b.

The result holds trivially for $t = 0, 1$. For $t > 1$, first suppose that $a_i = b_i$ for all $t \leq i \leq n$. If $a = b$ then $[a, b]$ decomposes into one prefix, and the result holds. Otherwise, let k be the most significant bit such that $a_k < b_k$. We tackle the cases $k < t$ and $k \geq t$ separately.

First assume that $k < t$. Then, $[a, b]$ splits into $[a, a']$ and $[b', b]$, where

$$a' = a_n \cdots a_{k+1} a_k 1 \cdots 1 \quad b' = a_n \cdots a_{k+1} b_k 0 \cdots 0.$$

By Lemma 3.2, $[a, b]$ decomposes into at most $2k \leq 2(t-1)$ prefixes.

Now suppose that $k \geq t$. Let

$$a' = a_n \cdots a_t 11 \cdots 1 \quad b' = b_n \cdots b_t 00 \cdots 0.$$

Since $b - a + 1 \leq 2^t$ observe that $a', b' \in [a, b]$ satisfy $b' = a' + 1$. Thus, $[a, b] = [a, a'] \cup [b', b]$.

Furthermore, if $a_{t-1} = 0$, then $[a, a']$ has length at least $2^{t-1} + 1$. Then $[b', b]$ must have length at most $2^{t-1} - 1$, and so $b_{t-1} = 0$. Similarly, if $b_{t-1} = 1$ then $a_{t-1} = 1$. By Lemma 3.2, one interval decomposes into at most $t-1$ prefixes and the other in at most t prefixes, and so the result follows. □

The proof of Theorem 3.3 describes a recursive procedure to compute the prefix decomposition of an interval $[a, b] \in \mathcal{I}_n$. This procedure follows directly from the constructive argument in the proof, and we describe it as preDec in Algorithm 1.

Algorithm 1: preDec($[a, b], n+1$) [70]

1. If some of a, b are larger or equal than 2^{n+1}, then halt.
2. If $a = b$, return a.
3. Let $a_n \cdots a_0, b_n \cdots b_0$ be the bit representations of a, b. Find the most significant bit k for which $a_k < b_k$.
4. If $a_i < b_i$ for all $0 \leq i \leq k$, return $a_n \cdots a_{k+1} *$ (return $*$ if $k = n$).
5. Append the prefixes returned by preDec($[a_n \cdots a_0, a_n \cdots a_k 1 \cdots 1], n+1$) to the output string.
6. Append the prefixes returned by preDec($[b_n \cdots b_k 0 \cdots 0, b_n \cdots b_0], n+1$) to the output string.
7. Return the output string.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

When encrypting a document located at a point (w_1, w_2), the idea behind the first proposed scheme we propose is to attach the document to every possible tuple (w'_1, w'_2) for w'_i in the parent tuple of w_i $(i = 1, 2)$. Then, to generate a trapdoor for a rectangle $R = I_1 \times I_2$, we query all tuples in the Cartesian product of the prefix decomposition of both intervals I_1, I_2. Note that this implicitly defines a partition in rectangles of R.

Definition 3.4. Let $\mathcal{S}$ be a Boolean SSE scheme (see Definition 2.18). We define a 2-dimensional range searchable symmetric encryption scheme $\mathcal{R}_1$ by means of the following five polynomial-time algorithms:

$\mathcal{R}_1.\mathrm{Gen}(\lambda)$**:**
Return the output of $\mathcal{S}.\mathrm{Gen}(\lambda)$, including the secret key k, and the bit-length $n+1$.

$\mathcal{R}_1.\mathrm{Enc}_k(\mathbf{D})$**:**
Given an input collection of plaintext documents with associated points $\mathbf{D} = ((w_{i,1}, w_{i,2}), D_i)_{i=1}^N$, let $P_{i,j}$ denote the parent tuple of $w_{i,j}$ for $i = 1, \ldots, N$ and $j = 1, 2$. Let

$$\mathbf{D}' = \left(\left(w'_{i,1}, w'_{i,2}\right), D_i\right)_{\substack{w'_{i,1} \in P_{i,1}, w'_{i,2} \in P_{i,2} \\ i \in \{1,\ldots,N\}}}.$$

Return the output of $\mathcal{S}.\mathrm{Enc}_k(\mathbf{D}')$.

$\mathcal{R}_1.\mathrm{Trapdoor}_k(R)$**:**
Given the rectangle $R = I_1 \times I_2$ as input, denote by Q_j the output of the algorithm $\mathrm{preDec}(I_j, n+1)$ for $j = 1, 2$. Denote the Boolean formula satisfied only by the tuples in $Q_1 \times Q_2$ by $\Phi = \bigvee_{w_1 \in Q_1} \left(w_1 \wedge \left(\bigvee_{w_2 \in Q_2} w_2\right)\right)$. Return $T = \mathcal{S}.\mathrm{Trapdoor}_k(\Phi)$.

$\mathcal{R}_1.\mathrm{Search}(I, \mathbf{C}, T)$**:**
Return the output of $\mathcal{S}.\mathrm{Search}(I, \mathbf{C}, T)$.

$\mathcal{R}_1.\mathrm{Dec}_k(C)$**:**
Return the output of $\mathcal{S}.\mathrm{Dec}_k(C)$.

This scheme can be seen as a two-dimensional adaptation of the scheme defined in [68], where only the one-dimensional case is described. However, the schemes in [68] choose to reveal the length of all prefixes. This slightly speeds up the searching process with respect to our construction, but also increases the leakage. We further discuss this issue in Section 3.1.9.

3.1.6 Prefix Decomposition Using Quadtrees

In this section we present our second 2-DRSSE scheme.

We carry out the idea of the previous construction by using quadtrees. As we see in Sections 3.1.9, 3.1.10, this has the effect of reducing the leakage incurred by the scheme, while increasing the trapdoor size and search time.

We start by giving an analog to Theorem 3.3 for quadtrees. The proof follows a similar argument as in Theorem 3.3, and it is given in Section 3.1.11.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Theorem 3.5. *Let t be a positive integer and let $[a,b]\times[c,d]$ be a rectangle such that both $[a,b]$ and $[c,d]$ have length greater than 2^{t-1} and at most 2^t. Then, it decomposes into at most $7\cdot 2^{t+1}-18t-3$ prefixes.*

The procedure preDec in Algorithm 2 computes the prefix decomposition of a rectangle $R=[a,b]\times[c,d]$, where $[a,b],[c,d]\in\mathcal{I}_n$ satisfy the restrictions of this theorem. It follows directly from the constructive argument in the proof of Theorem 3.5.

Algorithm 2: preDec$([a,b],[c,d],n+1)$

1 If some of a,b,c,d are larger or equal than 2^{n+1}, then halt.
2 Let $a=a_n\cdots a_0$, $b=b_n\cdots b_0$, $c=c_n\cdots c_0$, $d=d_n\cdots d_0$ be the bit representations of a, b, c and d.
3 If $a=b$, return $\{(a_n\cdots a_0, e_n\cdots e_0)\}_{e_n\cdots e_0\in[c,d]}$.
4 If $c=d$, return $\{(e_n\cdots e_0, c_n\cdots c_0)\}_{e_n\cdots e_0\in[a,b]}$.
5 Find k_1,k_2 the most significant bits such that $a_{k_1}<b_{k_1}$ and $c_{k_2}<d_{k_2}$.
6 If $k_1=k_2$ and for all $0\le i\le k_1$ we have $a_i<b_i$ and $c_i<d_i$, return $(a_n\cdots a_{k_1+1}*, c_n\cdots c_{k_1+1}*)$.
7 If $k_1=k_2$ but the previous condition is not satisfied, let $A_1=[a,a_n\cdots a_{k_1+1}01\cdots 1]$, $A_2=[a_n\cdots a_{k_1+1}10\cdots 0,b]$, $C_1=[c,c_n\cdots c_{k_1+1}01\cdots 1]$, $C_2=[c_n\cdots c_{k_1+1}10\cdots 0,d]$. Return the union of the outputs of preDec$(A_i\times C_j,n+1)$ for $i,j=1,2$.
8 If $k_1<k_2$, return the union of the outputs of preDec$([a,b]\times[c,c_n\cdots c_{k_2+1}01\cdots 1],n+1)$ and preDec$([a,b]\times[c_n\cdots c_{k_2+1}10\cdots 0,d],n+1)$
9 If $k_1>k_2$, return the union of the outputs of preDec$([a,a_n\cdots a_{k_1+1}01\cdots 1]\times[c,d],n+1)$ and preDec$([a_n\cdots a_{k_1+1}10\cdots 0,b]\times[c,d],n+1)$

When encrypting a document located at a point (w_1,w_2), the idea behind the second proposed scheme is to attach the document to every element in the parent tuple of $(w_1.w_2)$. Then, to generate a query for a rectangle $R=I_i\times I_2$, we query all elements in its prefix decomposition.

Definition 3.6. Let $\mathcal{S}$ be a SSE scheme (see Definition 2.18). We define a 2-DRSSE scheme $\mathcal{R}_2$ by means of the following five polynomial-time algorithms:

$\mathcal{R}_2.\mathrm{Gen}(\lambda)$**:**
Return the output of $\mathcal{S}.\mathrm{Gen}(\lambda)$, including the secret key k, and the bit-length $n+1$.

$\mathcal{R}_2.\mathrm{Enc}_k(\mathbf{D})$**:**
Given as input a collection of plaintext documents with associated points $\mathbf{D}=((w_{i,1},w_{i,2}),D_i)_{i=1}^N$, let P_i denote the parent tuple of the point $(w_{i,1},w_{i,2})$.
Let $\mathbf{D}'=(w_i',D_i)_{\substack{w_i'\in P_i\\ i\in\{1,\ldots,N\}}}$.
Return the output of $\mathcal{S}.\mathrm{Enc}_k(\mathbf{D}')$.

$\mathcal{R}_2.\mathrm{Trapdoor}_k(R)$**:**
Given the rectangle R, denote by Q the output of the function preDec$(R,n+1)$. Let $\Phi=\bigvee_{w\in Q}w$ denote the Boolean formula satisfied only by the prefixes of Q.
Return $T=\mathcal{S}.\mathrm{Trapdoor}_k(\Phi)$.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

$\mathcal{R}_2$.Search$(I, \mathbf{C}, T)$**:**
Return the output of $\mathcal{S}$.Search$(I, \mathbf{C}, T)$.

$\mathcal{R}_2$.Dec$_k(C)$**:**
Return the output of $\mathcal{S}$.Dec$_k(C)$.

3.1.7 Over-Covers Using Binary Trees

In this section we present our third 2-DRSSE scheme. We first state the required results, and we proceed by describing the construction.

There are mainly two drawbacks in using prefix decomposition in binary trees. The first one is that the size of the trapdoor associated to a rectangle R usually increases in the size of the partition of R that the prefix decomposition defines. Consequently, by Theorem 3.3, larger rectangles can have larger trapdoors. The second one is that the size of such trapdoors reveals information about the queried rectangle, since larger trapdoors also correspond to larger rectangles.

These concerns are addressed by Faber et al. in [68], where they accordingly propose the concept of over-cover. As defined earlier in Section 3.1.3.2, an over-cover of an interval is a set of prefixes such that the union of the associated prefix ranges contains the interval.

As the following theorem states, the relaxation to over-covers allows prefix representations of constant size 3, and so it solves the concerns mentioned in the previous paragraph.

Theorem 3.7 ([68])**.** *Let t be an integer with $1 \leq t \leq n+1$, and let s, n' be positive integers with $s < t$ and $n'+2 < 2^s$. Then, any interval $[a, b] \in \mathcal{I}_n$ of length $2^t+2^s+n'+2$ admits an over-cover of size 3, with profile $\{n-t+1, n-s+1, \min\{n-s, n-t+2\}\}$.*

However, Faber et al. note that the interval defined by the corresponding prefix ranges is on average about 40% larger than the original interval (and at most 66% larger), and so it may contain values outside the given range. This translates into possibly having a large amount of false positives in the search results.

In order to mitigate this effect, we extend their result and reduce the false-positive rate, while still being able to produce constant-sized over-covers. Given an interval length L, we associate to it a parameter $h \in \{1, \ldots, \lfloor \log(L) \rfloor\}$ prior to the generation of the over-covers. For $h = \lfloor \log(L) \rfloor$ our over-covers are similar to the ones built in [68]. Our results show that for smaller values of h we can produce over-covers covering a smaller range and having more prefixes, thus reducing the false-positive rate at the cost of slightly larger communication complexity and search time. The next lemma provides a first approximation to building these over-covers.

Lemma 3.8. *Let $L < 2^{n-1}$ and $h \leq \lfloor \log(L) \rfloor$ be positive integers. Then, every interval of length L admits an over-cover with $\lfloor L/2^h \rfloor + 2$ disjoint prefixes, where $\lfloor L/2^h \rfloor + 1$ prefixes have length $n-h+1$ and one prefix has length $n-h+2$.*

Proof. To prove this, let $[a, b] = [a_n \cdots a_0, b_n \cdots b_0]$ be an interval of length L. Consider first the prefix $a_n \cdots a_h*$ of length $n-h+1$ covering a, and consider also the $\lfloor L/2^h \rfloor + 1$ next consecutive prefixes of length $n-h+1$. Since the corresponding prefix ranges sum

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

to a length of $2^h(\lfloor L/2^h \rfloor + 2)$, which is between $L + 2^h$ and $L + 2^{h+1}$, the last of such prefix ranges contains b.

Now, split the first prefix $a_n \cdots a_h*$ and the last prefix $b_n \cdots b_h*$ into two prefixes of length $n - h + 2$ each

$$\begin{array}{ll} a_n \cdots a_h 0* & a_n \cdots a_h 1*, \\ b_n \cdots b_h 0* & b_n \cdots b_h 1 * . \end{array}$$

Then, the interior prefixes (that is, all but the first and last prefixes of length $n-h+2$) sum to a length of $2^h\lfloor L/2^h \rfloor + 2 \cdot 2^{h-1} > L$. Therefore at least one of the outer prefixes of the over-cover can be removed while still having an over-cover of $[a, b]$, and the theorem follows. □

Note that the number of prefixes in the over-cover in the Lemma 3.8 is $\lfloor L/2^h \rfloor + 2$. We now look for a reduction in the number of prefixes of the over-cover, since this quantity has an impact on the communication complexity of our scheme. This is achieved by the following lemma

Lemma 3.9. *Given a positive integer r, all intervals admitting a prefix decomposition in r consecutive prefixes of the same length admit a prefix decomposition in less than $2\lceil \log(r + 1) \rceil$ prefixes and with the same profile.*

Proof. The lemma is trivially satisfied for $r = 1$.

Now, consider r prefixes of the same length l. When r is even, at least $r - 2$ prefixes of length l can be merged pairwise into prefixes of length $l - 1$. And when r is odd, at least $r - 1$ prefixes can be merged pairwise. Therefore, at least $2\lfloor (r - 1)/2 \rfloor$ prefixes can be merged pairwise, giving either one or two prefixes of length l and $\lfloor (r - 1)/2 \rfloor < (r + 1)/2 - 1$ prefixes of length $l - 1$.

By applying the above argument recursively on r, we obtain a set of prefixes whose profile depends exclusively on r and l. Furthermore, at step i we get either one or two prefixes of length $l - i + 1$ and at most $(r + 1)/2^i - 1$ prefixes of length $l - i$. This argument can be repeated at most $\lceil \log(r+1) \rceil$ times, giving a decomposition of at most $2\lceil \log(r + 1) \rceil$ prefixes. □

By applying Lemma 3.9 to Lemma 3.8 we obtain an exponential reduction on the amount of prefixes in the over-cover. Note that the resulting over-covers have the same profile for intervals of equal length. The next theorem summarizes this, and gives upper bounds on the size and false-positive rate of the resulting over-cover. We compile some particular cases in Table 3.1.

Theorem 3.10. *Let $L < 2^{n-1}$ and $h \leq \lfloor \log(L) \rfloor$ be positive integers, and set $s = L/2^h$. Then, every interval $[a, b] \in \mathcal{I}_n$ of length L admits an over-cover with the same profile, of the same size σ and covering the same length $\rho \cdot L$. Moreover σ depends exclusively on $\lfloor s \rfloor$, and*

$$\begin{aligned} \sigma &\leq 2\lceil \log(\lfloor s \rfloor + 2) \rceil + 1, \\ \rho &\leq 3/2s + \lfloor s \rfloor / s. \end{aligned}$$

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Proof. The bound on σ is a direct consequence of Lemmas 3.8, 3.9. Since the over-cover given in Lemma 3.8 covers a total length of $2^h\lfloor L/2^h \rfloor + 3 \cdot 2^{h-1}$ and the original interval has length L, the theorem follows. □

A procedure to compute the over-cover of an interval $[a, b] \in \mathcal{I}_n$ follows directly from the constructive arguments in the proofs of Lemmas 3.8 and 3.9. We define it as the procedure preCov in Algorithm 3. Note that preCov uses a recursive sub-algorithm reduceCov, which realizes the reduction provided by Lemma 3.9. The description of this sub-algorithm can be found in Algorithm 4.

In practice, we associate to each interval of length L a parameter $h \in \{1, \ldots, \lfloor \log(L) \rfloor\}$, which is adjusted a priori as the largest value for which the false-positive rate achieves a desirable bound. Then, when building an over-cover of any interval of length L, the interval and the corresponding h parameter are fed into the preCov algorithm.

Algorithm 3: preCov$([a, b], h, n + 1)$

1 If some of a, b are larger or equal than 2^{n+1}, then halt.
2 Let $L = b - a + 1$ and $t = \lfloor \log(L) \rfloor$. Halt if $L >= 2^{n-1}$, $h < 1$ or $h > t$.
3 Let $a = a_n \cdots a_0$, $b = b_n \cdots b_0$ denote the bit representations of a, b.
4 Initialize two empty sets S, S_1 and two variables i, s.
5 If $a_{h-1} = 1$, set $i = 1$ and $s = \lfloor L/2^h \rfloor + 1$, and add the prefix $a_n \cdots a_{h-1}*$ to S_1. Otherwise, set $i = 0$ and $s = \lfloor L/2^h \rfloor$, and add $b_n \cdots b_{h-1}*$ to S_1.
6 Let $\alpha = a_n \cdots a_h$.
7 While $i \leq s$, add to S the prefix of length $n - h + 1$ defined by $\alpha + i$, and increment i by 1.
8 Output the union of S_1 and reduceCov$(S, n + 1)$.

Algorithm 4: reduceCov$(P, n + 1)$

1 If $|P| = 1$, output P and halt.
2 Parse $P = (p_i)_{i=1}^t$ as an ordered tuple of n-bit prefixes $p_i = a_n^{(i)} \cdots a_{n-l+1}^{(i)}*$ of length l, sorted so that $a_n^{(i)} \cdots a_{n-l+1}^{(i)} + 1 = a_n^{(i+1)} \cdots a_{n-l+1}^{(i+1)}$ for every i. If P is not of this form, then halt.
3 Initialize two empty ordered tuples S_1 and S and a variable i.
4 If $a_{n-l+1}^{(1)} = 0$, set $i = 1$. Otherwise, append p_1 to S_1 and set $i = 2$.
5 If $a_{n-l+1}^{(t)} = 0$, append p_t to S_1.
6 While $i < t$, append the prefix $a_n^{(i)} \cdots a_{n-l+2}^{(i)}*$ to S and set $i = i + 2$.
7 Output S_1 and the output of reduceCov$(S, n + 1)$.

The third proposed scheme follows the same idea as the $\mathcal{R}_1$ scheme defined in Section 3.1.5. The difference is that, in the process of generating trapdoors for a rectangle $R = I_1 \times I_2$, we use over-covers of the corresponding intervals instead of prefix decompositions. This implicitly defines a set of rectangles whose union may strictly contain R, and so there can be documents in the result set falling outside of the queried rectangle.

Definition 3.11. Let $\mathcal{S}$ be a Boolean searchable symmetric encryption scheme (see Definition 2.18). We define a 2-dimensional range searchable symmetric encryption scheme $\mathcal{R}_3$ by means of the following five polynomial-time algorithms:

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

$\lfloor L/2^h \rfloor$	σ	upper bound on ρ
1	3	150%
2	3	75.0%
3	4	50.0%
4	4	37.5%
5	5	30.0%
6	4	25.0%
7	5	21.4%
8	5	18.7%
9	6	16.6%
10	5	15.0%

TABLE 3.1: Number of binary tree prefixes σ and upper bound on the false-positive rate ρ of an over-cover, related to the value $L/2^h$.

$\mathcal{R}_3.\mathrm{Gen}(\lambda)$**:**
Return the output of $\mathcal{S}.\mathrm{Gen}(\lambda)$, including the secret key k, and the bit-length $n+1$.

$\mathcal{R}_3.\mathrm{Enc}_k(\mathbf{D})$**:**
Given as input a collection of plaintext documents with associated points $\mathbf{D} = ((w_{i,1}, w_{i,2}), D_i)_{i=1}^N$, let $P_{i,j}$ denote the parent tuple of $w_{i,j}$ for $i = 1, \ldots, N$ and $j = 1, 2$. Let

$$\mathbf{D}' = \left(\left(w'_{i,1}, w'_{i,2}\right), D_i\right)_{\substack{w'_{i,1} \in P_{i,1}, w'_{i,2} \in P_{i,2} \\ i \in \{1,\ldots,N\}}}.$$

Return the output of $\mathcal{S}.\mathrm{Enc}_k(\mathbf{D}')$.

$\mathcal{R}_3.\mathrm{Trapdoor}_k(R)$**:**
Given the rectangle $R = I_1 \times I_2$, let L_j denote the length of I_j and choose $h_j \in \{1, \ldots, \lfloor \log(L_j) \rfloor\}$ as a function of L_j for $j = 1, 2$. Denote by Q_j the output of $\mathrm{preCov}(I_j, h_j, n+1)$ for $j = 1, 2$. Let

$$\Phi = \bigvee_{w_1 \in Q_1} \left(w_1 \wedge \left(\bigvee_{w_2 \in Q_2} w_2 \right) \right)$$

denote the Boolean formula satisfied only by the tuples in $Q_1 \times Q_2$.

Return $T = \mathcal{S}.\mathrm{Trapdoor}_k(\Phi)$.

$\mathcal{R}_3.\mathrm{Search}(I, \mathbf{C}, T)$**:**
Return the output of $\mathcal{S}.\mathrm{Search}(I, \mathbf{C}, T)$.

$\mathcal{R}_3.\mathrm{Dec}_k(C)$**:**
Return the output of $\mathcal{S}.\mathrm{Dec}_k(C)$.

3.1.8 Over-Covers Using Quadtrees

In this section we present our fourth 2-DRSSE scheme. We first state the required results, and we proceed by describing the construction.

Since, when using quadtrees, the trapdoor size depends on the size of the queried rectangle by Theorem 3.5, both the efficiency and the privacy concerns stated at the beginning

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

of the previous section hold in the $\mathcal{R}_2$ scheme. The concept of over-cover, which is naturally adapted to quadtrees, mitigates both concerns. As defined earlier in Section 3.1.3.3, an over-cover of a rectangle is a set of prefixes such that the union of the associated prefix rectangles contains the rectangle.

The following theorem is an adaptation of Theorem 3.10. Applying over-covers allows prefix representations of constant size and variable false-positive rate. When building an over-cover of a rectangle with sides of length L, we consider a parameter $h \in \{1, \ldots, \lfloor \log(L) \rfloor\}$ taken as a function of L. The parameter h is adjusted as the largest value for which the false-positive rate achieves an acceptable bound. For $h = \lfloor \log(L) \rfloor$ this over-cover has size 13 and a false-positive rate of at most 525%. Decreasing h also decreases the area covered by the over-cover, an thus the false-positive rate. We compile some particular cases in Table 3.2.

Theorem 3.12. *Let $L < 2^{n-1}$ and $h \leq \lfloor \log(L) \rfloor$ be positive integers, and set $s = L/2^h$. Then, every rectangle $[a,b] \times [c,d] \in \mathcal{I}_n^2$ with sides $[a,b], [c,d]$ of length L admits an over-cover with the same profile, of the same size σ and covering the same area $\rho \cdot L^2$. Moreover σ depends exclusively on $\lfloor s \rfloor$, and*

$$\sigma \leq 4^{\lceil \log(\lfloor s \rfloor + 2) \rceil + 3},$$
$$\rho \leq (3/2s + \lfloor s \rfloor / s)^2.$$

Proof. Consider the over-covers given by Lemmas 3.8, 3.9. By decomposing in quadtree prefixes each of the elements in the Cartesian product of these two over-covers, we find a set of quadtree prefixes that have the same lengths for all intervals $[a,b]$ and $[c,d]$ as considered.

Note that for each interval $[a,b], [c,d]$ we have a prefix of size $n - t + h + 1$ and at most two prefixes of length $n - t + h - i$ for every $i \in \{0, \ldots, 2\lceil \log(s+2) \rceil\}$, where $s = \lfloor L/2^h \rfloor$. Therefore, the number of quadtree prefixes found in this way is at most $1 + 4 \cdot 2\lceil \log(s+2) \rceil + 4 \sum_{i,j=0}^{2\lceil \log(s+2) \rceil} 2^{|i-j|}$. Since it can be proved by induction that $\sum_{i,j=0}^{n} 2^{|i-j|} = 2^{n+3} - 3n - 7$, this quantity amounts to $2^{2\lceil \log(s+2) \rceil + 5} - 4\lceil 2 \log(s+2) \rceil - 27$.

Since the original area is L^2 and the area covered by the quadtree prefixes is $(2^h \lfloor L/2^h \rfloor + 3 \cdot 2^{h-1})^2$, the theorem follows. □

The procedure preCov in Algorithm 5 computes an over-cover of a rectangle $[a,b] \times [c,d] \in \mathcal{I}_n^2$ according to Theorem 3.10. It follows directly from the constructive argument in the proof of Theorem 3.12.

The following fourth scheme follows the same idea as the $\mathcal{R}_2$ scheme defined in Section 3.1.6. The only difference is that, in the process of generating trapdoors for a rectangle $R = [a,b] \times [c,d]$, we use an over-cover of R instead of a prefix decomposition. Also, note that the queried two-dimensional ranges are restricted so that both intervals have equal length.

As in the previous case, this implicitly defines a set of two-dimensional ranges whose union strictly contains R, and so there can be documents in the result set falling outside the queried domain.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

$\lfloor L/2^h \rfloor$	σ	upper bound on ρ
1	13	525%
2	19	206%
3	30	125%
4	34	89.0%
5	49	69.0%
6	48	56.2%
7	67	47.4%
8	67	41.0%
9	90	36.1%
10	79	32.2%

TABLE 3.2: Number of quadtree prefixes σ and upper bound on the false-positive rate ρ of an over-cover, related to the value $L/2^h$.

Algorithm 5: preCov$([a,b],[c,d],h,n+1)$

If some of a,b,c,d are larger or equal than 2^{n+1}, then halt.
Halt if $[a,b],[c,d]$ have different length, i.e., if $b-a+1 \neq d-c+1$.
Let $L=b-a+1$ and $t=\lfloor \log(b-a+1) \rfloor$. Halt if $L \geq 2^{n-1}$, $h<1$ or $h>t$.
Let $a=a_n \cdots a_0$, $b=b_n \cdots b_0$ denote the bit representations of a, b, c and d.
Initialize an empty set S.
Set $S_1=\text{preCov}([a,b],h,n+1)$ and $S_2=\text{preCov}([c,d],h,n+1)$.
For every $(p,q) \in S_1 \times S_2$, denote $p=p_n \cdots p_r*$ and $q=q_n \cdots q_s*$, and let $p_1=p_n \cdots p_r 0 \cdots 0$, $p_2=p_n \cdots p_r 1 \cdots 1$, $q_1=q_n \cdots q_s 0 \cdots 0$, $q_2=q_n \cdots q_s 1 \cdots 1$. Add preDec$([p_0,p_1],[q_0,q_1],n+1)$ to S.
Output S.

Definition 3.13. Let $\mathcal{S}$ be a searchable symmetric encryption scheme (see Definition 2.18). We define a 2-dimensional range searchable symmetric encryption scheme $\mathcal{R}_4$ by means of the following five polynomial-time algorithms:

$\mathcal{R}_4.\text{Gen}(\lambda)$**:**
Return the output of $\mathcal{S}.\text{Gen}(\lambda)$, including the secret key k, and the bit-length $n+1$.

$\mathcal{R}_4.\text{Enc}_k(\mathbf{D})$**:**
Given as input a collection of plaintext documents with associated points $\mathbf{D}=(D_i,(w_{i,1},w_{i,2}))_{i=1}^N$, let P_i denote the parent tuple of the point $(w_{i,1},w_{i,2})$.
Let $\mathbf{D}'=(w_i',D_i)_{\substack{w_i' \in P_i \\ i \in \{1,\ldots,N\}}}$.
Return the output of $\mathcal{S}.\text{Enc}_k(\mathbf{D}')$.

$\mathcal{R}_4.\text{Trapdoor}_k(R)$**:**
Given the rectangle $R=[a,b]\times[c,d]$, let L_1 and L_2 denote the length of $[a,b]$ and $[c,d]$, respectively. If $L_1 \neq L_2$, the algorithm halts.
Choose $h \in \{1,\ldots,\lfloor \log(L_1) \rfloor\}$ as a function of L, and denote by Q the output of the algorithm preCov$(R,h,n+1)$. Let $\Phi=\bigvee_{w \in Q} w$ denote the Boolean formula satisfied only by those prefixes in Q.
Return $T=\mathcal{S}.\text{Trapdoor}_k(\Phi)$.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

$\mathcal{R}_4$.Search$(I, \mathbf{C}, T)$**:**
Return the output of $\mathcal{S}$.Search$(I, \mathbf{C}, T)$.

$\mathcal{R}_4$.Dec$_k(C)$**:**
Return the output of $\mathcal{S}$.Dec$_k(C)$.

3.1.9 Security Analysis

In this section we use the Boolean OXT scheme by Cash et al. [93] as the underlying SSE scheme in our constructions. By following the security analysis carried out in [93, Section 5.3], we characterize the leakage profile under the $\mathcal{L}$*-semantic-security against adaptive attacks* security definition by Cash et al. for each of our schemes. We refer the reader to the article [93] or to Section 2.5.1 for the security definition statement. Also, note that the information leakage of searchable encryption schemes can often be reduced by combining them with private information retrieval or oblivious random access memory schemes. In this work we do not discuss these techniques. For more details, see [68].

The $\mathcal{L}$*-semantic-security against adaptive attacks* security definition is stated in the real-ideal paradigm, and it is parametrized by a *leakage function* $\mathcal{L}$. The input of $\mathcal{L}$ is the whole outsourced data set and all exchanged queries.

If the scheme is $\mathcal{L}$-semantically-secure, $\mathcal{L}$ outputs an upper bound for the leakage incurred by the scheme in the security game. In other words, if the scheme is $\mathcal{L}$-semantically-secure, any probabilistic polynomial-time adversary having access to the whole data set and to all queries exchanged during the game is not able to deduce any extra information apart from what it is able to deduce from the output of $\mathcal{L}$. In this way, the output of $\mathcal{L}$ models what can potentially be leaked to the server in a protocol execution that follows the security game.

We introduce some terminology and notation in order to describe the output of the leakage function. This notation is inspired by spatial data, although our schemes deal with general quantitative two-dimensional data sets. Suppose that the client uploads the encryption of the collection of N plaintext documents $((w_{i,1}, w_{i,2}), D_i)_{i=1}^N$. For each plaintext document $((w_1, w_2), D)$ we say that (w_1, w_2) is its *location*, w_1 is its *latitude*, and w_2 is its *longitude*.

Also suppose that, during the game, the client adaptively sends M queries to the server for documents located in rectangles $(R_i)_{i=1}^M$. In the process of generating trapdoors, any such rectangle $R_i = I_{i,1} \times I_{i,2}$ is implicitly expressed as a collection of other rectangles. This is done in different ways depending on the scheme:

- In $\mathcal{R}_1$, we consider the tuples of prefixes in the Cartesian product of preDec$(I_{i,1}, n+1)$ and preDec$(I_{i,2}, n+1)$.
- In $\mathcal{R}_2$, we consider every prefix in preDec$(R_i, n+1)$.
- In $\mathcal{R}_3$, we consider every tuple of prefixes in the Cartesian product of the over-covers preCov$(I_{i,1}, h_1, n+1)$ and preCov$(I_{i,2}, h_2, n+1)$.
- In $\mathcal{R}_4$, we consider every prefix in preCov$(R_i, h, n+1)$.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Since every prefix expresses a prefix range (in $\mathcal{R}_1$, $\mathcal{R}_3$) or a prefix rectangle (in $\mathcal{R}_2$, $\mathcal{R}_4$), this defines a collection of rectangles. We denote such collection by $(R_{i,1}, \ldots, R_{i,m_i})$ in all cases. Also, for any rectangle $R = I_1 \times I_2$, we say that I_1 is its *latitude* and I_2 is its *longitude*.

We now characterize the leakage profile $\mathcal{L}$ by following the upper bounds on the allowed leakage of the OXT scheme given in [93, Section 5.3].

The output of a leakage function $\mathcal{L}$, such that all of the schemes $\mathcal{R}_1$, $\mathcal{R}_2$, $\mathcal{R}_3$ and $\mathcal{R}_4$ are $\mathcal{L}$-semantically-secure, is determined by:

- Total number of encrypted documents N.
- *Results pattern* RP: Encrypted documents corresponding to points located in $R_{i,j}$, for each i, j.
- *Profile pattern* Φ: Whether we are using binary trees or quadtrees, and m_i for all $i = 1, \ldots, M$.

In the cases of the $\mathcal{R}_1$ and $\mathcal{R}_3$ schemes, define the rest of the output of $\mathcal{L}$ by:

- *Equality pattern* $\bar{s}$: Whether the latitude of two different $R_{i,j}$ coincides.
- *Size pattern* SP: Number of documents in the latitude of $R_{i,j}$ for each i, j.
- *Intersection pattern* IP: Encrypted documents whose latitude is contained in the latitudes of two rectangles $R_{i,j}$ having the exact same longitude.

And in the case of the $\mathcal{R}_2$ and $\mathcal{R}_4$ schemes, define the remainder of the output of $\mathcal{L}$ by:

- *Equality pattern* $\bar{s}$: Whether $R_{i,j} = R_{k,l}$ for any i, j, k, l.

Then, as a corollary to the analysis done in [93, Section 5.3], the schemes $\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3, \mathcal{R}_4$ are $\mathcal{L}$-semantically secure against adaptive attacks with $\mathcal{L}$ defined as above.

It is clear that N leaks the same information in all cases. We also deem the information leaked by the results pattern RP as roughly equivalent in all cases, although it is dependent on the expression $(R_{i,1}, \ldots, R_{i,m_i})$ of the queried rectangle R_i.

As to the profile pattern Φ, the situation is presumably very different depending on whether one uses prefix decomposition or over-covers. As observed in [68, Section 3], knowledge about m_i can potentially determine the exact size of the queried rectangle. When using over-covers, m_i can be considered as constant for queries with equal length according to Theorems 3.10 and 3.12, so Φ reveals at most the length of the latitude and longitude of the queried rectangles. However, this leakage is reduced for larger values of the parameter h, or when h is chosen so that m_i does not exceed a small enough constant. For instance, this leakage vanishes for $h = \lfloor \log(L) \rfloor$. Thus, a stronger security guarantee comes at the cost of a higher false-positive rate.

As an aside, observe that, if we had included the prefix lengths in the queries (as done for one-dimensional range queries in [68]), then Φ would definitely reveal the exact lengths

of the latitude and the longitude of the queried rectangles. Knowledge about prefix lengths would also allow to distinguish between rectangles of the same size. This is fixed in [68, Section 3] by introducing the concept of *universal covers*.

Regarding the equality pattern $\bar{s}$, when using binary trees the leakage seems much larger. The reason is that, for large and sparse enough data sets, knowledge about the equality pattern and the result pattern can be used to decide whether the rectangles in consideration are equal or not. When using quadtrees, $\bar{s}$ is roughly equivalent to having deterministic trapdoors, which is standard leakage in SSE.

Finally, when using binary trees, the size pattern SP and intersection pattern IP leak additional information about documents in the same latitude or longitude than queried rectangles.

3.1.10 Efficiency Analysis

In this section we provide the worst-case complexity analysis of the proposed schemes and of some of the related schemes in the literature. As in the previous section, we use the Boolean OXT scheme by Cash et al. [93] in the underlying SSE scheme for our constructions.

3.1.10.1 Spatial Data

As stated in the introduction, we study the problem of searching over encrypted data in the context of the project *CLARUS: User-Centered Privacy and Security in the Cloud* [1]. In *CLARUS*, we focus on the two-dimensional range queries that ask for the collection of features in the data set intersecting a given rectangle.

We consider the requirements derived from the use of the OGC® Web Feature Service Interface Standard (WFS) interface. WFS defines web interface operations for querying and editing vector geographic features. In this case the user requests geographical features across the web using platform-independent calls. These features can be described by points, paths or polygons. Our queries can be described using PostGIS / PostgreSQL. PostGIS is a spatial database extender that adds support for geographic objects allowing location queries. We here focus on enhancing the security of `SELECT` queries with clause `WITH` of the form `ST_MakeEnvelope(X,Y,*)`, which creates a rectangle described by the two opposite vertexes `X` and `Y` and returns the elements intersecting this rectangle.

3.1.10.2 Worst-Case Complexity Analysis

The complexity analysis in Table 3.3 is stated in terms of the *resolution* $L = 2^{n+1}$. Thus, L is the number of possible values in each dimension of the grid Λ_n, where the data is located.

Index size is thought as the total size of the index. Trapdoor size and search processing time are taken to be the size of a trapdoor associated to a rectangle and the time needed to retrieve the corresponding results.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Scheme	Index	Trapdoor	Search
SBCSP [66]	$O(L^2 \log L)$	$O(\log L)$	$O(L^2 \log^2 L)$
BW [65]	$O(L^3)$	$O(1)$	$O(L^2)$
$\mathcal{R}_1$	$O(L^2)$	$O(\log^2 L)$	$O(L \log^2 L)$
$\mathcal{R}_2$	$O(L^2)$	$O(L \log L)$	$O(L^3)$
$\mathcal{R}_3$	$O(L^2)$	$O(\log L)$	$O(L)$
$\mathcal{R}_4$	$O(L^2)$	$O(\log L)$	$O(L^2)$

TABLE 3.3: Worst-case efficiency comparison between 2-DRSSE schemes.

In writing the index size and the search processing time complexity analysis, we assume that every point in the grid Λ_n holds up to one document. This assumption is reasonable since the OXT scheme is suitable for static data sets, and plaintext documents are initially thought to be identifiers (see [93, Section 2.1]). More precisely, when outsourcing a data set with our schemes, the client encrypts documents of the form $((w_{i,1}, w_{i,2}), \mathrm{id}_i)$ with the SSE scheme, and then associates outsourced ciphertexts to the particular id_i by other means.

We also assume that the h parameters in the trapdoor generation algorithm are taken so that, when using over-covers, the client issues constant-sized trapdoors. This is indeed the case when $h = \lfloor \log(L) \rfloor$, or when h is chosen so that over-covers do not exceed a small chosen constant size σ.

It is interesting to note that, even if in $\mathcal{R}_3$ and $\mathcal{R}_4$ the client issues a constant number of trapdoors per queried rectangle, the size of the trapdoors is logarithmic in L. The use of pseudo-random functions in the trapdoor generation algorithm of the OXT scheme forces the size of every trapdoor to scale in $\log L$, and thus it may be possible to obtain constant-sized trapdoors by changing the underlying searchable encryption scheme.

3.1.11 Proof of Theorem 3.5

For presentation purposes, in this section we provide a proof of Theorem 3.5. We start with the following two lemmas.

Lemma 3.14 (Side rectangles)**.** *Let t be a positive integer and let $[a, b] \times [c, d]$ be a rectangle. Suppose that $[a, b]$ and $[c, d]$ have both length at most 2^t and*

1. *$[a, b]$ is a $(n - t + 1)$-bit prefix and the least t significant bits of c are zero, or*
2. *$[a, b]$ is a $(n - t + 1)$-bit prefix and the least t significant bits of d are one, or*
3. *$[c, d]$ is a $(n - t + 1)$-bit prefix and the least t significant bits of a are zero, or*
4. *$[c, d]$ is a $(n - t + 1)$-bit prefix and the least t significant bits of b are one.*

Then $[a, b] \times [c, d]$ decomposes in at most $\max(1, 2^{t+1} - 2)$ prefixes.

Proof. Denote by p_t^{SIDE} the size of the largest prefix decomposition of rectangles of the form in the statement.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Note that $p_0^{\text{SIDE}} = 1$ and $p_1^{\text{SIDE}} = 2$. Now consider the case 1) and assume that $t > 1$. Then $a = a_n \cdots a_t 0 \cdots 0$, $b = a_n \cdots a_t 1 \cdots 1$, $c = c_n \cdots c_t 0 \cdots 0$ and $d = c_n \cdots c_t d_{t-1} \cdots d_0$.

Note that $[a, b] = a_n \cdots a_t *$ splits into $[a, a'] = a_n \cdots a_t 0 *$ and $[b', b] = a_n \cdots a_t 1 *$. We have two cases:

If $d_{t-1} = 0$, since $[a, a'] \times [c, d]$ and $[b', b] \times [c, d]$ both satisfy the conditions of the lemma for $t - 1$, they admit a decomposition into at most p_{t-1}^{SIDE} prefixes each.

If $d_{t-1} = 1$, then $[c, d]$ decomposes into $[c, c']$ and $[d', d]$, where

$$c' = c_n \cdots c_t 01 \cdots 1 \quad d' = c_n \cdots c_t 10 \cdots 0.$$

By the same argument as in the previous point, $[a, b] \times [d', d]$ decomposes into at most $2p_{t-1}^{\text{SIDE}}$ prefixes. Since $[a, b] \times [c, c']$ clearly decomposes into two prefixes, we have $p_t^{\text{SIDE}} \leq 2p_{t-1}^{\text{SIDE}} + 2$, and the result for 1) follows.

We omit the cases 2), 3), 4), since their proof is similar to that of 1). □

Lemma 3.15 (Edge rectangles)**.** *Let t be a positive integer and let $[a, b] \times [c, d]$ be a rectangle. Suppose that both $[a, b]$ and $[c, d]$ have length at most 2^t and*

1. *the least t significant bits of a and c are zero, or*
2. *the least t significant bits of a and d are zero and one respectively, or*
3. *the least t significant bits of b and c are one and zero respectively, or*
4. *the least t significant bits of b and d are one.*

Then $[a, b] \times [c, d]$ decomposes in at most $\max(1, 2^{t+2} - 3t - 3)$ prefixes.

Proof. Denote by p_t^{EDGE} the size of the largest prefix decomposition of rectangles of the form in the statement, and denote by p_t^{SIDE} the size of the largest prefix decomposition of rectangles of the form in the statement of Lemma 3.14.

Note that $p_0^{\text{EDGE}} = 1$ and $p_1^{\text{EDGE}} = 2$. Now assume $t > 1$ and consider the case 1). Then, the $n + 1$-bit representations of a, b, c, d are

$$\begin{array}{ll} a = a_n \cdots a_t 0 \cdots 0 & b = b_n \cdots b_0 \\ c = c_n \cdots c_t 0 \cdots 0 & d = d_n \cdots d_0 \end{array}$$

with $a_i = b_i$ and $c_i = d_i$ for all $i \geq t$.

We consider four different cases.

Firstly, if $b_{t-1}, d_{t-1} = 0$, there is a prefix decomposition of size at most p_{t-1}^{EDGE} by induction.

If $b_{t-1} = 0$ and $d_{t-1} = 1$, then $[c, d]$ decomposes into $[c, c'] \cup [d', d]$ where

$$c' = c_n \cdots c_t 01 \cdots 1 \quad d' = d_n \cdots d_t 10 \cdots 0.$$

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

This gives two sub-rectangles: one $[a,b] \times [c,c']$ which is decomposable into at most p_{t-1}^{SIDE} prefixes by Lemma 3.14, and the other $[a,b] \times [d',d]$ which is decomposable into at most p_{t-1}^{EDGE} prefixes.

If $b_{t-1} = 1$ and $d_{t-1} = 0$, a similar argument also exhibits a decomposition of size $p_{t-1}^{\text{SIDE}} + p_{t-1}^{\text{EDGE}}$.

Finally, if $b_{t-1}, d_{t-1} = 1$ then the intervals $[a,b]$ and $[c,d]$ split into $[a,a'] \cup [b',b]$ and $[c,c'] \cup [d',d]$ respectively, where

$$\begin{aligned} a' &= a_n \cdots a_t 01 \cdots 0 \quad & b' &= b_n \cdots b_t 10 \cdots 0 \\ c' &= c_n \cdots c_t 01 \cdots 1 \quad & d' &= c_n \cdots c_t 10 \cdots 0. \end{aligned}$$

By a similar argument to the previous point, there is a prefix decomposition of size at most $1 + 2p_{t-1}^{\text{SIDE}} + p_{t-1}^{\text{EDGE}}$.

Since the last bound is the largest one, we get the result from $p_t^{\text{EDGE}} \leq 1 + 2p_{t-1}^{\text{SIDE}} + p_{t-1}^{\text{EDGE}}$.

We omit the cases 2), 3), 4), since their proof is similar to that of 1). □

We now restate and prove Theorem 3.5.

Theorem 3.16 (Theorem 3.5)**.** *Let t be a positive integer and let $[a,b] \times [c,d]$ be a rectangle such that both $[a,b]$ and $[c,d]$ have length greater than 2^{t-1} and at most 2^t. Then, it decomposes into at most $7 \cdot 2^{t+1} - 18t - 3$ prefixes.*

Proof. Denote by p_t^{EDGE} the size of the largest prefix decomposition of rectangles of the form in the statement of Lemma 3.15.

The result holds trivially for $t = 0, 1$, so assume $t > 1$. Write the $n+1$-bit representations of a, b, c, d as

$$\begin{aligned} a &= a_n \cdots a_0 \quad & b &= b_n \cdots b_0 \\ c &= c_n \cdots c_0 \quad & d &= d_n \cdots d_0 \end{aligned}$$

If k_1, k_2 are the most significant bits such that $a_{k_1} < b_{k_1}$ and $c_{k_2} < d_{k_2}$, then $k_1, k_2 \geq t-1$ by hypothesis. We consider four different cases

Firstly, if $k_1, k_2 = t - 1$, then the intervals $[a,b]$ and $[c,d]$ split into $[a,a'] \cup [b',b]$ and $[c,c'] \cup [d',d]$ respectively, where

$$\begin{aligned} a' &= a_n \cdots a_t 01 \cdots 1 \quad & b' &= b_n \cdots b_t 10 \cdots 0 \\ c' &= c_n \cdots c_t 01 \cdots 1 \quad & d' &= d_n \cdots d_t 10 \cdots 0. \end{aligned}$$

This divides $[a,b] \times [c,d]$ into four sub-rectangles, each of which is decomposable into at most p_{t-1}^{EDGE} prefixes by Lemma 3.15, and so we have a decomposition into at most $4p_{t-1}^{\text{EDGE}}$ prefixes.

Secondly, suppose $k_1 = t - 1$ and $k_2 \geq t$. Let

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

$$a' = a_n \cdots a_t 01 \cdots 1 \quad b' = b_n \cdots b_t 10 \cdots 0$$
$$c' = c_n \cdots c_t 11 \cdots 1 \quad d' = d_n \cdots d_t 00 \cdots 0.$$

Since $d-c+1 \leq 2^t$ observe that $c', d' \in [c, d]$ satisfy $d' = c'+1$. Thus, $[c, d] = [c, c'] \cup [d', d]$.

Furthermore, if $c_{t-1} = 0$, then $[c, c']$ has length at least $2^{t-1} + 1$. Then $[d', d]$ has length at most $2^{t-1} - 1$, and so $d_{t-1} = 0$. Similarly, if $d_{t-1} = 1$ then $c_{t-1} = 1$. Without loss of generality, assume that $c_{t-1} = 0$ and $d_{t-1} = 0$. By Lemma 3.15, $[a, b] \times [c, c']$ is decomposable into at most p_{t-1}^{EDGE} prefixes, and $[a, a'] \times [d', d]$ and $[b', b'] \times [d', d]$ are decomposable into at most p_{t-2}^{EDGE} prefixes each. This yields a prefix decomposition of $[a, b] \times [c, d]$ into at most $p_{t-1}^{\text{EDGE}} + 2p_{t-2}^{\text{EDGE}}$ prefixes.

Thirdly, if $k_1 \geq t$ and $k_2 = t - 1$, we proceed similarly to the last point to get a prefix decomposition of at most $p_{t-1}^{\text{EDGE}} + 2p_{t-2}^{\text{EDGE}}$ prefixes.

Lastly, if $k_1, k_2 \geq t$, by the same reasons as in the second point, we have

$$a' = a_n \cdots a_t 11 \cdots 1 \quad b' = b_n \cdots b_t 00 \cdots 0$$
$$c' = c_n \cdots c_t 11 \cdots 1 \quad d' = d_n \cdots d_t 00 \cdots 0$$

with $b' = a' + 1$ and $d' = c' + 1$, so $[c, d] = [c, c'] \cup [d', d]$. Also, it is not possible to have both $a_{t-1} = 0$ and $b_{t-1} = 1$, or to have both $c_{t-1} = 0$ and $d_{t-1} = 1$. Without loss of generality, assume that $a_{t-1}, b_{t-1}, c_{t-1}, d_{t-1} = 0$. Then $[a, a']$ and $[c, c']$ split in $[a, a''] \cup [a''', a']$ and $[c, c''] \cup [c''', c']$, where

$$a'' = a_n \cdots a_t 01 \cdots 1 \quad a''' = a_n \cdots a_t 10 \cdots 0$$
$$c'' = c_n \cdots c_t 01 \cdots 1 \quad c''' = c_n \cdots c_t 10 \cdots 0.$$

By Lemma 3.15, we have that $[a, a'] \times [c, c']$ is decomposable in at most p_t^{EDGE} prefixes, and each of $[b', b] \times [d', d]$, $[a, a''] \times [d', d]$, $[a''', a'] \times [d', d]$, $[a, a'] \times [c, c'']$ and $[a, a'] \times [c''', c']$ are decomposable in at most p_{t-1}^{EDGE} prefixes each.

We thus get a decomposition of $[a, b] \times [c, d]$ into at most $p_t^{\text{EDGE}} + 5p_{t-1}^{\text{EDGE}}$ prefixes.

Since the last bound is the largest one, we get a prefix decomposition of at most the claimed size. □

A recursive procedure to compute the prefix decomposition of a rectangle $R = [a, b] \times [c, d]$, where $[a, b], [c, d] \in \mathcal{I}_n$ have just $n + 1$-bit values, follows directly from this proof. See Algorithm 2.

3.1.12 Conclusion

In this section we propose various techniques for searchable encryption allowing two-dimensional queries on encrypted data. We analyze the trade-off between performance, security and communication overhead of the presented options. Some solutions we provide take advantage of the Boolean search and inverted index properties of [93]. We also present a technique based on over-covers that notably reduces the communication cost of the queries at the expense of increasing the false-positive rate. It provides a convenient

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

trade-off between the searching time and the covered region, thus giving flexibility in reducing false positives at the cost of efficiency.

It is known that, when using searchable encryption, the server can infer statistical information about the stored data. This leakage depends on the election of keywords used to classify the data. It would be interesting to measure the information leakage in the case of range queries, and to find an election of the keywords minimizing it in a way that prevents some of the recent attacks to range searchable symmetric encryption [26–32].

In the case of two-dimensional spatial data, we deal with the case of documents indexed by single locations. However, it would be interesting to deal with the case of data indexed by paths and polygons. Although it is possible to index a particular document by many points using our solutions, the server could potentially distinguish which documents are attached to single points and which are attached to paths or polygons. A naive and inefficient solution to prevent this leakage would be to store these documents separately for every location.

A next step in this line of work is to deal with the single-writer multi-reader setting, in which a single client uploads an encrypted data set and delegates search capabilities to multiple clients. In this case, we are interested in providing access control on the query results. More precisely, we can consider the case where each reader is authorized to search only over data that is located within a region specified by an access policy. This situation requires an extension of the searchable encryption scheme supporting access control in the multi-client setting.

Finally, it would be interesting to find cryptographic schemes that enable efficient range queries over encrypted data while protecting against leakage-abuse attacks [26–32], especially against [30]. While there exist ways to avoid [30] by using exclusively SSE schemes, the leakage of the access pattern makes it difficult to guard against frequency analysis attacks. The most direct way to prevent these attacks is to apply a PIR scheme such as [123] instead of a SE scheme in our constructions. Applying PIR as in [124] effectively prevents access pattern leakage, but it can have an impact both on efficiency and on the security of the outsourced data set.

3.2 Public-Key Searchable Encryption with Conjunctive and Subset Keyword Search

3.2.1 Introduction

External storage servers allow users to retrieve outsourced data selectively. However, when such servers are managed by untrusted third parties, users are usually reluctant to outsource their sensitive data in the clear.

Encrypting the data prior to outsourcing it is a good approach to overcome this security concern. Nevertheless, traditional encryption schemes fail to provide selective retrieval in an efficient and secure way. Searchable encryption deals with this problem by allowing data owners to issue queries for encrypted outsourced data. Much like traditional encryption schemes, searchable encryption schemes generally come in two distinct types, serving different purposes: *symmetric-key searchable encryption* (SSE) [37] and *public key encryption with keyword search* (PEKS).

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Public key encryption with keyword search schemes are searchable encryption schemes set in the public key setting, where there may exist multiple data suppliers that send data over to a single client. Public key encryption with keyword search was initially proposed by Boneh et al. [38] in 2004. See Sections 1.4.2 and 2.5.2 for a brief introduction to PEKS.

The keyword search protocol usually considered in PEKS involves the following entities:

- a set of *data suppliers*, which provides and encrypts the data to be outsourced,
- a storage *server* (e.g. an e-mail gateway or a database), which stores the outsourced data and executes the search process, and
- a *client* of the storage server, who retains the ability to generate queries for the encrypted data.

In the protocol, the client sets up the scheme, keeps the secret key private and distributes the public key to the data providers. By using the public key, the data providers can generate searchable encrypted data and upload it to the storage server. In turn, the client retains the ability to generate queries for the encrypted data. By sending such queries to the storage server, the client empowers the storage server to select the encrypted data items satisfying the query, and to return those data items to the client.

Regarding query expressiveness, the scheme in [38] achieves *single-keyword* queries. Single-keyword PEKS schemes enable data providers to generate an encryption of a keyword w, which is to be uploaded to the storage server. We call this encryption a *searchable index* and we denote it by $\mathbf{I}(w)$. The client, holding the secret key, can build a *trapdoor* $\mathbf{T}(w')$ corresponding to some keyword w'. By sending $\mathbf{T}(w')$ to the storage server, the client empowers the storage server to learn whether any searchable index $\mathbf{I}(w)$ satisfies $w = w'$ or not, but no other information about $\mathbf{I}(w)$ is revealed in this process.

One of the most common enhancements of PEKS is *conjunctive PEKS*, which enables *conjunctive field keyword queries* [42, 65, 73, 74]. Typically, in conjunctive PEKS data providers can encrypt a tuple of keywords $(w_1, \ldots, w_m)$ by using the public key of the client, producing a searchable index $\mathbf{I}(w_1, \ldots, w_m)$. The client can use its secret key to generate a trapdoor associated to a tuple of keywords $(w'_1, \ldots, w'_l)$, along with a set of *keyword fields* or *positions* $\{j_1, \ldots, j_l\} \subseteq \{1, \ldots, m\}$. When receiving this trapdoor, the storage server is able to check whether the predicate $\bigwedge_{i=1}^{l}(w_{j_i} = w'_i)$ holds or not by using the index $\mathbf{I}(w_1, \ldots, w_m)$ and the trapdoor.

Another enhancement of PEKS is *subset PEKS*, first defined in [65], which enables *subset queries*. In subset PEKS, data providers can encrypt a tuple of keywords $(w_1, \ldots, w_m)$ by using the public key of the client, producing a searchable index $\mathbf{I}(w_1, \ldots, w_m)$. The client can use its secret key to generate a trapdoor associated to m arbitrary sets of keywords $(A_1, \ldots, A_m)$. When receiving this trapdoor, the storage server is able to check whether the conjunctive subset query predicate $\bigwedge_{i=1}^{m}(w_i \in A_i)$ holds or not by using the index $\mathbf{I}(w_1, \ldots, w_m)$ and the trapdoor.

In most of the proposed applications for single-keyword PEKS, the ciphertexts that are uploaded to the server by the data suppliers take the form

$$\mathrm{Enc}_{\mathrm{pk}}(D) \| \mathbf{I}(w_1) \| \cdots \| \mathbf{I}(w_m),$$

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

where pk is the public key of the client, Enc is some public-key encryption scheme, the data item D is indexed by keywords $w_1, \ldots, w_m$, and $\mathbf{I}(w_1), \ldots, \mathbf{I}(w_m)$ are the corresponding searchable indexes. The client can recover the data items that are indexed by the keyword w in position $i \in \{1, \ldots, m\}$ by sending the trapdoor $\mathbf{T}(w)$ and the position i to the server. The basic security property of PEKS schemes guarantees that the server does not learn any information about the searchable indexes unless it has the knowledge of a matching trapdoor.

Note that one could achieve conjunctive queries by using single-keyword PEKS schemes, simply by querying for particular keywords as stated above, and computing the intersection of the results locally or in the storage server. When doing so, the server learns which data items are indexed by each of the keywords. The benefits of using conjunctive PEKS against using single-keyword PEKS in this way can mainly be in efficiency and security, since trapdoors are usually much shorter, the intersection predicate is embedded in the trapdoor and the intersection is computed at the storage server.

The earliest application scenario for PEKS, as suggested by Boneh et al. in [38], is e-mail gateways. In this scenario, a user Alice wishes to read her e-mails, which are stored in an untrusted e-mail gateway in an encrypted form. When retrieving her e-mails, she may want the e-mail gateway to forward her just e-mails satisfying certain conditions, e.g. containing the tag "urgent" or having a particular sender "Bob". To enable the e-mail gateway to filter the e-mail messages correctly, she sends the trapdoors corresponding to these keywords to the gateway, e.g. $\mathbf{T}(\text{“tag:urgent”})$ and $\mathbf{T}(\text{“sender:Bob”})$.

Now, suppose user Bob wishes to send Alice an e-mail message D. He may encrypt D by using Alice's public key pk and then attach to it the sender information and the "urgent" tag in the form of searchable indexes of the form $\mathbf{I}(\text{“sender:Bob”})$ and $\mathbf{I}(\text{“tag:urgent”})$. Thus, Alice's gateway would receive the message

$$\text{Enc}_{\text{pk}}(D) \| \mathbf{I}(\text{“sender:Bob”}) \| \mathbf{I}(\text{“tag:urgent”}).$$

By matching the attached searchable indexes with the stored trapdoors, the e-mail gateway is able to learn which e-mail messages should be forwarded to Alice. However, it learns nothing else about the messages in the process. This example illustrates an application covered by PEKS that seems, *a priori*, hard to cover by using exclusively symmetric-key mechanisms such as SSE.

Another natural application of PEKS schemes involves secure audit logs, and it was devised by Waters et al. in [125]. In their proposed application, audit logs are stored in an untrusted storage server in an encrypted form by using an Identity-Based Encryption (IBE) scheme (e.g. [102, 120, 126–128]), and PEKS searchable indexes are attached to it. Attributes for IBE and keywords for PEKS are related to the audit record at hand, e.g. date and time. An investigator Bob may wish to be granted access to audit logs recorded, for instance, on a particular date. To do so, Bob asks the key escrow agent, say Alice, for the trapdoors and decryption keys corresponding to this particular date. If Alice authorized Bob to issue this particular search, she would serve Bob the requested IBE decryption keys and PEKS trapdoor, and Bob would then be able to retrieve the audit logs of interest from the untrusted storage server.

Other applications include secure cloud storage [44], decryption key delegation systems [73] and context-based forwarding [129]. Although SSE represents a much more efficient solution than PEKS for cloud storage in the symmetric setting, PEKS schemes can

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

be useful in applications involving asymmetric architectures, such as when sending or sharing outsourced data.

The first PEKS scheme was presented by Boneh et al. [38] in 2004. This was immediately followed by the first conjunctive searchable encryption scheme in the symmetric setting by Golle et al. [130], and by an extension of PEKS to conjunctive PEKS by Park et al. [74]. Many authors then presented alternative PEKS schemes that allow decryption of indexes [65, 73, 118], multi-dimensional range queries [66], reduction of communication and storage costs [40, 41], extension to multi-user systems [42] or improvements of security in various ways [43–48]. Among them, one of the most relevant is the work by Boneh and Waters [65], in which they define the general notion of a Hidden-Vector Encryption (HVE) scheme, providing an enhancement of expressiveness of the queries that allows for conjunctive field keyword search, and also decryption of indexes.

Also, the relationship between PEKS schemes and Anonymous Identity-Based Encryption (AIBE), was first established in [38]. Most AIBE schemes (e.g., see [102, 120, 126–128]) can be easily translated to PEKS schemes and vice-versa via a generic blackbox transformation [38, 119].

We propose two PEKS schemes. The first one achieves conjunctive field keyword search. We prove the computational consistency of this scheme, and we prove security under the asymmetric DBDH assumption. Under the proposed security definition, our scheme does not provide any security enhancement against using a single-keyword PEKS scheme to issue conjunctive field keyword queries. Nevertheless, as we see in Section 3.2.7, it improves all previous related schemes in terms of efficiency in the most critical operations. In addition, the trapdoors generated by using this first scheme consist of just one group element, and the index size improves all previous conjunctive PEKS schemes.

The second proposed scheme enables a class of generalized subset queries, which includes subset queries as defined in [65]. We prove the computational consistency of our scheme, and we prove our scheme secure under the $p-$BDHI assumption. The proposed security definition guarantees that nothing is leaked from searchable indexes apart from the output of the search process. To the best of our knowledge, apart from the scheme in [65] no other subset PEKS schemes have been proposed in the literature. The proposed scheme improves [65] in terms of efficiency and expressiveness, and it does not assume that keywords are taken from a finite keyword space.

The remainder of this section is structured as follows. In Section 3.2.2, we outline the preliminaries needed in this work. Our constructions for conjunctive and subset PEKS are described in Sections 3.2.3 and 3.2.5. Sections 3.2.4 and 3.2.6 feature the consistency and security proofs for our conjunctive and subset PEKS schemes, respectively. In Section 3.2.7, we analyze the efficiency of our schemes. We give some final remarks and future work directions in Section 3.2.8.

3.2.2 Preliminaries

We start this section by defining some general notation and by stating the general model for the proposed PEKS schemes. We then give the consistency and security definitions, providing the hardness assumptions on which we base the security of our schemes. We finally give some implementation remarks.

3.2.2.1 Notation

We start by giving some standard notation and definitions used in searchable encryption. In this work, a *keyword* denotes a binary string $w \in \{0,1\}^*$. We here define a *document* as a tuple of keywords $\mathbf{D} = (w_1, \ldots, w_m)$, and we say that keyword w_i is in *keyword field* (or *position*) i of $\mathbf{D}$.

If $\mathbf{D}_0, \mathbf{D}_1$ are two documents, we denote by $\mathbf{D}_0 \Delta \mathbf{D}_1$ the set of keywords appearing in either $\mathbf{D}_0$ or $\mathbf{D}_1$, but not in both at the same time. So if, for instance, $\mathbf{D}_0 = (\text{“a”}, \text{“b”}, \text{“c”})$ and $\mathbf{D}_1 = (\text{“a”}, \text{“d”}, \text{“e”})$, we would then have that $\mathbf{D}_0 \Delta \mathbf{D}_1 = \{\text{“b”}, \text{“c”}, \text{“d”}, \text{“e”}\}$. We also name $\mathbf{D}_0 \Delta \mathbf{D}_1$ as the set of keywords *distinguishing* $\mathbf{D}_0$ and $\mathbf{D}_1$.

Given a positive integer m, we denote by $[m]$ the set $\{1, \ldots, m\}$.

3.2.2.2 General Model for PEKS Schemes

We here give the model used for the proposed conjunctive and subset public-key searchable encryption schemes. Although not stated, every algorithm apart from Gen takes the public parameters as input.

Definition 3.17. We define a PEKS scheme $\mathcal{S}$ as consisting of four polynomial-time algorithms:

$\mathcal{S}.\text{Gen}(\lambda)$**:** Probabilistic algorithm run by the client that, given a security parameter λ, returns the private key sk, the public key pk and the public parameters params of the scheme.

$\mathcal{S}.\text{Enc}_{\text{pk}}(\mathbf{D})$**:** Probabilistic algorithm run by data providers. It takes as input a document $\mathbf{D}$ and returns a corresponding searchable index $\mathbf{I}$.

$\mathcal{S}.\text{Trapdoor}_{\text{sk}}(\mathbf{L}, J)$**:** Algorithm run by the client that takes as input a tuple $\mathbf{L}$ of keywords and a set J of positions. It returns a corresponding trapdoor $\mathbf{T}$.

$\mathcal{S}.\text{Search}(\mathbf{I}, \mathbf{T})$**:** Deterministic algorithm run by the server and taking as input a searchable index $\mathbf{I}$ and a trapdoor $\mathbf{T}$. It returns either 1 or 0.

3.2.2.3 Consistency Definition

The consistency property relates to the correctness of the scheme, in the sense that a searchable index and a trapdoor should match in the search process exactly when the underlying document and query also match. If a document and a query match, then by construction of our schemes the corresponding searchable index and trapdoor match in the search process. However, the converse does not necessarily hold. In this regard, the usage of hash functions in the proposed schemes induces the existence of false positives in the search process. Therefore, we must analyze the extent to which false positives can be produced, and we recur to a notion of consistency defined by Abdalla et al. in [119].

The consistency notions defined by Abdalla et al. in [119] are, in increasing strength order, *computational*, *statistical* and *perfect*. We prove consistency in the random oracle

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

model and under an adaptation of the weakest definition of consistency in [119], namely computational consistency. We here state their definition for the particular cases of conjunctive and subset PEKS schemes.

Let $\mathcal{S}$ be a PEKS scheme. Given a security parameter λ, we introduce a consistency game in the following three phases:

- *Setup.* The challenger runs $\mathcal{S}$.Gen on input λ, hands over the public parameters and the public key pk to the adversary, and keeps the private key sk secret.
- *Guess.* The adversary outputs a document of the form $\mathbf{D} = (w_1, \ldots, w_m)$ and a tuple of keywords of the form $\mathbf{L} = (w'_1, \ldots, w'_l)$ together with a set of positions $J = \{j_1, \ldots, j_l\} \subseteq [m]$.
- *Output.* The challenger hands over the trapdoor $\mathbf{T} = \mathcal{S}.\mathrm{Trapdoor}_{\mathrm{sk}}(\mathbf{L}, J)$ to the adversary, and the adversary computes $\mathbf{I} = \mathcal{S}.\mathrm{Enc}_{\mathrm{pk}}(\mathbf{D})$. If $\mathcal{S}.\mathrm{Search}(\mathbf{I}, \mathbf{T}) = 1$ and if there exists a $j_i \in J$ such that $w_{j_i} \neq w'_i$, then the adversary outputs a bit $b = 1$. Otherwise, it outputs $b = 0$.

Definition 3.18 (Computational Consistency of Conjunctive PEKS [119])**.** A PEKS scheme $\mathcal{S}$ is *computationally consistent* if the advantage of every PPT adversary $\mathcal{A}$ in the above game

$$\mathrm{Adv}_{\mathcal{A}}(\lambda) = \Pr(b = 1)$$

is negligible in λ.

3.2.2.4 Hardness Assumptions

Here we provide the hardness assumptions used in the security analysis of the proposed schemes.

The first scheme we propose is proved secure under the asymmetric Decisional Bilinear Diffie-Hellman assumption (asymmetric DBDH). This assumption is proposed in the work [102] by Boneh and Boyen as a generalization of the DBDH assumption (see [103]) to the asymmetric setting. The DBDH assumption is easily seen to imply DDH in the target group $\mathbb{G}_T$.

Definition 3.19 (Asymmetric DBDH Assumption)**.** Let $\mathbb{G}_1 = \langle g \rangle$, $\mathbb{G}_2 = \langle h \rangle$ be asymmetric bilinear groups deterministically generated according to a security parameter λ. We say the *asymmetric* DBDH *assumption* holds in $\mathbb{G}_1$ and $\mathbb{G}_2$ if for every PPT algorithm $\mathcal{B}$,

$$\begin{aligned}\mathrm{Adv}_{\mathcal{B}}(\lambda) = \Big|&\Pr\left(\mathcal{B}(g, g^a, g^b, h, h^a, h^c, e(g,h)^{abc}) = 1\right) \\ &-\Pr\left(\mathcal{B}(g, g^a, g^b, h, h^a, h^c, e(g,h)^{r}) = 1\right)\Big|\end{aligned}$$

is negligible in λ, where the probabilities are taken over a, b, c, r uniformly distributed in $\mathbb{F}_q$ and over the random bits of $\mathcal{B}$.

The second scheme we propose is proved secure under the Bilinear Diffie Hellman Inversion Assumption ($p-$BDHI). This assumption is also proposed in [102]. According to [100], the best known algorithm breaking $p-$BDHI is to solve the Discrete Logarithm Problem (DLP) in $\mathbb{G}$.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Definition 3.20 ($p-$ BDHI Assumption)**.** Let $\mathbb{G} = \langle g \rangle$ denote a symmetric bilinear group deterministically generated according to a security parameter λ, and let p be a positive integer. We say the $p-$BDHI *assumption* holds in $\mathbb{G}$ if for every PPT algorithm $\mathcal{B}$,

$$\mathrm{Adv}_{\mathcal{B}}(\lambda) = \Pr\left(\mathcal{B}(g, g^a, g^{a^2}, \ldots, g^{a^p}) = e\,(g,g)^{1/a}\right)$$

is negligible in λ, where the probabilities are taken over uniformly distributed $a \in \mathbb{F}_q$ and over the random bits of $\mathcal{B}$.

In the proposed schemes and in the definitions above, bilinear groups are generated according to the security parameter λ. We choose bilinear groups to have exponential order in λ. See the security and consistency proofs for more details about this choice.

3.2.2.5 Security Definition

We now introduce the security definition used in this work. We adapt the security definition introduced by Boneh et al. in [38] to the conjunctive and subset case of Definition 3.17. All proofs in this work are set in the random oracle model (see [104]).

The used security definition is a semantic-security style definition that guarantees searchable index indistinguishability in the face of an adversary with access to the public key and to trapdoors not containing any keyword distinguishing the challenge candidate documents. Therefore, in the security definition we propose, the adversary is not allowed to obtain trapdoors associated to any word that appears in one of the challenge candidate documents, but not in both.

Let $\mathcal{S}$ be a PEKS scheme. Given a security parameter λ, we introduce a security game in the following five phases:

- *Setup.* The challenger runs Gen on input λ, hands over the public parameters and the public key pk to the adversary, and keeps the private key sk secret.
- *Query Phase 1.* The adversary adaptively requests the challenger for q_T trapdoors of its own choice, where q_T is a polynomial value in the security parameter λ. We denote the set of all keywords queried in this phase by $\mathcal{W}_1$.
- *Challenge.* The adversary outputs two challenge candidate documents $\mathbf{D}_0, \mathbf{D}_1$ subject to the restriction that keywords appearing in $\mathbf{D}_0 \Delta \mathbf{D}_1$ have not been queried in Query Phase 1. That is, $(\mathbf{D}_0 \Delta \mathbf{D}_1) \cap \mathcal{W}_1 = \emptyset$. The challenger throws a fair coin $b \in \{0,1\}$, and outputs the searchable index $\mathcal{S}.\mathrm{Enc}(\mathbf{D}_b)$ corresponding to $\mathbf{D}_b$.
- *Query Phase 2.* The adversary proceeds just as in Query Phase 1, but it is not allowed to ask for trapdoors containing keywords in $\mathbf{D}_0 \Delta \mathbf{D}_1$. That is, if the set of all keywords queried in this phase is $\mathcal{W}_2$, we impose $(\mathbf{D}_0 \Delta \mathbf{D}_1) \cap \mathcal{W}_2 = \emptyset$.
- *Guess.* The adversary outputs a guess $b' \in \{0,1\}$ for b.

Definition 3.21 (Semantic Security against Adaptive Chosen Keyword Attacks)**.** We say that a PEKS scheme $\mathcal{S}$ is *semantically secure against adaptive chosen keyword attacks*

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

if the advantage of every PPT adversary $\mathcal{A}$ in distinguishing b in the above game

$$\begin{aligned}\mathrm{Adv}_{\mathcal{A}}(\lambda) =& |\Pr(b' = b) - 1/2| \\ =& |\Pr(\mathcal{A}(X) = b | X = b) - \Pr(\mathcal{A}(X) = b | X = 1 - b)|\end{aligned}$$

is negligible in λ.

For conjunctive PEKS, the security definition we consider is slightly weaker than in other related works, in the following sense. Works such as [40, 42, 65, 116, 130, 131] impose the natural restriction of serving the adversary just trapdoors coming from queries with equal search outcome over the two challenge candidate documents. In the case of conjunctive queries, the restriction we pose is stronger, since served trapdoors can not contain any keywords distinguishing the challenge candidate documents. This implies that trapdoors could leak which searchable indexes contain some of the keywords in the underlying query, even if there is not a match.

In addition, the considered security definition does not provide trapdoor unlinkability or remove the need for a secure channel for trapdoors, as studied for instance in [43, 44, 48, 118].

3.2.2.6 Implementation Remarks

We refer the reader to [132] for remarks and references on the following statements and for a recent review on the state of the art of pairing computation.

Implementation of secure asymmetric bilinear groups for elliptic curve cryptography is often based on BN, BLS, KSS or MNT elliptic curves. In turn, symmetric bilinear groups are implemented in practice on supersingular elliptic curves.

Supersingular elliptic curves are well known to require large prime order groups for the DLP to be intractable (since they have a small MOV exponent) and this would enlarge the size of exchanged information in the proposed schemes. Moreover, recent results on the discrete logarithm problem [133] have rendered symmetric bilinear groups effectively obsolete for cryptographic purposes. Nevertheless, as in [134], we note that the second scheme we propose can be implemented in asymmetric bilinear groups as well, thus reducing the group order and increasing efficiency and security. In this context, symmetric bilinear groups are used just to facilitate the construction of the formal security proof.

We should note that asymmetric bilinear groups guarantee that we can securely and efficiently hash onto $\mathbb{G}_1$. In particular, it is possible to efficiently and uniformly sample from $\mathbb{G}_1$ without computing multiples of the generator g. The fact that we prove security under the random oracle model forces the use of such hash functions in the proposed schemes. See [107] for an explicit solution on secure hashing for BN curves.

3.2.3 Conjunctive PEKS Scheme

The proposed scheme can be seen as an analog to Boneh et al.'s scheme [38] by replacing the symmetric computational-type hardness assumption with an asymmetric decisional-type one. This replacement by a stronger assumption allows one to take advantage of

the bilinearity of pairings and build a conjunctive PEKS scheme with small trapdoors and indexes and efficient search process.

Following [40, 42, 73, 74, 130], we assume that the documents to be encrypted satisfy that

1. two different keyword fields never hold the same keyword, and
2. every keyword field is defined.

As noted in the literature [74], this can be effectively achieved by appending a keyword field identifier to every keyword. For instance, when encrypting a document of the form $(w_1, \ldots, w_n)$, one can assume that $w_i = i \| w_i'$ for some keyword w_i' (which could be `NULL` or $\perp$) and for all $i \in [m]$. We implicitly assume keywords in documents and trapdoors to be of this form.

We now describe the proposed conjunctive PEKS scheme. Although not stated, in the following every algorithm apart from $\mathcal{S}_1$.Gen takes the public parameters as input.

Definition 3.22. We define a public-key encryption with conjunctive keyword search scheme $\mathcal{S}_1$ by means of the following four polynomial-time algorithms:

$\mathcal{S}_1$.Gen(λ)**:** Given a security parameter $\lambda \in \mathbb{Z}$, fix two asymmetric bilinear groups $\mathbb{G}_1, \mathbb{G}_2$ of prime order $q \geq 2^\lambda$ and denote the corresponding pairing by $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. Let g, h be random generators of $\mathbb{G}_1, \mathbb{G}_2$ respectively. Let $H : \{0,1\}^* \to \mathbb{G}_1$ be a collision-resistant hash function. Define $m \in \mathbb{Z}$ as the fixed number of keywords in every document, which we assume constant in λ and satisfying $m \leq (1 + \log q)/2$. Choose $\beta \in \mathbb{F}_q$ uniformly at random. Output the public parameters $\text{params} = \{\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, e, g, h, H, m\}$, the private key β and the public key $\alpha = h^\beta$.

$\mathcal{S}_1$.Enc$_\alpha$($\mathbf{D}$)**:** Denote by $\mathbf{D} = (w_1, \ldots, w_m)$ the input document consisting of a tuple of m keywords $w_i \in \{0,1\}^*$. Then, uniformly generate a random nonce $r \in \mathbb{F}_q$ and set

$$I_0 = h^r$$
$$I_i = e\left(H(w_i), \alpha^r\right) \quad \text{for } i \in [m].$$

Output the index $\mathbf{I} = (I_0, I_1, \ldots, I_m)$.

$\mathcal{S}_1$.Trapdoor$_\beta$($\mathbf{L}, J$)**:** Denote by $\mathbf{L} = (w_1, \ldots, w_l)$ the input tuple of keywords (where $l \leq m$ and $w_i \in \{0,1\}^*$) and by $J = \{j_1, \ldots, j_l\} \subseteq [m]$ the input set of positions. Set

$$T_0 = \left(\prod_{i=1}^{l} H(w_i)\right)^\beta.$$

Output the trapdoor $\mathbf{T}$, consisting of T_0 along with the fields J to be queried.

$\mathcal{S}_1$.Search($\mathbf{I}, \mathbf{T}$)**:** Given the index $\mathbf{I} = (I_0, I_1, \ldots, I_m)$ and the trapdoor $\mathbf{T} = (T_0, J)$, where $J = \{j_1, \ldots, j_l\}$, output 1 if

$$e\left(T_0, I_0\right) = \prod_{i=1}^{l} I_{j_i}.$$

Otherwise output 0.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

We next give the consistency and security theorems for our scheme. The proofs are deferred to Sections 3.2.4.1 and 3.2.4.2, respectively.

Theorem 3.23. *The proposed conjunctive PEKS scheme $\mathcal{S}_1$ is computationally consistent under the random oracle model.*

Theorem 3.24. *Assume that the* DBDH *assumption holds. Then, the proposed conjunctive PEKS scheme $\mathcal{S}_1$ is semantically secure against adaptive chosen keyword attacks under the random oracle model.*

3.2.4 Consistency and Security Proofs for Conjunctive PEKS Scheme $\mathcal{S}_1$

In this section we present the consistency and security proofs of the conjunctive PEKS scheme $\mathcal{S}_1$.

3.2.4.1 Consistency Proof for the Conjunctive PEKS Scheme $\mathcal{S}_1$

We dedicate this section to the proof of Theorem 3.23. By proceeding in a similar way than in the proof by Abdalla et al. in [119], we prove consistency of the scheme $\mathcal{S}_1$ in the random oracle model.

Let $\mathcal{A}$ be a PPT adversary in the consistency game defined in Section 3.2.2.3, having access to the public parameters, to the public key pk and to the hash oracle H modeled as a random oracle. Let WSet be the set of keywords queried to the hash oracle H throughout the game, whose size q_H is polynomial in λ. Let $\mathbf{D} = (w_1, \ldots, w_m)$, $\mathbf{L} = (w'_1, \ldots, w'_l)$ and $J = \{j_1, \ldots, j_l\} \subseteq [m]$ denote the guess of $\mathcal{A}$ in the *Guess* phase, where keywords are taken from WSet, and let $\tilde{J}$ be the set of positions $j_i \in J$ such that $w_{j_i} \neq w'_i$. Without loss of generality, we rule out adversaries choosing $\tilde{J} = \emptyset$ in the *Guess* phase. Let $r \in \mathbb{F}_q$ denote the random nonce generated by $\mathcal{A}$ in the searchable index generation of the *Output* phase.

Denote $X = e\left(\prod_{i\in J} H(w_i), h^{\beta r}\right)$ and also denote $X' = e\left(\prod_{i=1}^{l} H(w'_i), h^{\beta r}\right)$. Now note that the output of $\mathcal{A}$ in the consistency game is 1 if and only if $X = X'$. We proceed to bound the probability of this event, which is $\mathrm{Adv}_{\mathcal{A}}(\lambda)$ by definition.

Let E be the event that there exist $\mathbf{D} = (w_1, \ldots, w_m)$, $\mathbf{L} = (w'_1, \ldots, w'_l)$ and $J = \{j_1, \ldots, j_l\} \subseteq [m]$, among all possible guesses taking words in WSet, in such a way that the equality $\prod_{i=1}^{l} H(w_{j_i}) = \prod_{i=1}^{l} H(w'_i)$ is satisfied. If $r\beta = 0$, then $\mathcal{A}$ always outputs 1. Otherwise, notice that $X = X'$ happens only when E happens. Therefore,

$$\mathrm{Adv}_{\mathcal{A}}(\lambda) \leq \frac{(q-1)^2}{q^2}\Pr(E) + \frac{2}{q}$$

Since $q \geq 2^{\lambda}$, it suffices to see that $\Pr(E)$ is negligible in λ.

Since H is modeled as a random oracle and since inversion permutes group elements, by using Lemma 3.25 we see that $\Pr(E) \leq q_H^{2m}\frac{m2^{2m}}{q}$. This bound is negligible in λ, since $q \geq 2^{\lambda}$, and m, q_H are assumed to be constant and polynomial in λ, respectively.

As a consequence of this result, we conclude the proof of Theorem 3.23. We next state the lemma used above.

Lemma 3.25. *Let $\mathbb{G}$ be a finite group of order q and neutral element 1. Let m, n be positive integers with $m \leq (1 + \log q)/2$ and set $X_1, \ldots, X_n$ independent and identically distributed uniform random variables with support $\mathbb{G}$.*

Let $A_{n,2m}$ denote the event that there exists a $S \subseteq [n]$ with $|S| \leq 2m$ such that $\prod_{i \in S} X_i = 1$. Then we have

$$\Pr(A_{n,2m}) \leq n^{2m} \frac{m 2^{2m}}{q}.$$

Proof. To make the notation simpler, denote $A_{t,t}$ by A_t and set $t = 2m$. Notice that $A_{n,t}$ happens for $X_1, \ldots, X_n$ exactly when it happens for some subset of $X_1, \ldots, X_n$ with $\min(n, t)$ terms. Therefore, by the union bound

$$\Pr(A_{n,t}) \leq \binom{n}{t} \Pr(A_t) \leq n^t \Pr(A_t).$$

We now lower bound the probability of the complementary event A_t^c.

We first prove $\Pr(A_t^c) \geq \prod_{i=0}^{t-1} \frac{q-2^i}{q}$ by induction on t over positive integers. For $t = 1$ we have $\Pr(A_1^c) = \frac{q-1}{q}$. For $t > 1$ note that, for A_t^c to happen with $X_1, \ldots, X_t$, the event A_{t-1} must happen with $X_1, \ldots, X_{t-1}$ and X_t can not take as a value any of the inverses of the subproducts of $X_1, \ldots, X_{t-1}$. Therefore, there are at least $q - 2^{t-1}$ possible values for X_t such that A_t happens and we get that $\Pr(A_t^c) \geq \frac{q-2^{t-1}}{q} \Pr(A_{t-1}^c)$ as claimed.

Now we have

$$\Pr(A_t) \leq 1 - \prod_{i=0}^{t-1} \frac{q - 2^i}{q} \leq 1 - \left(1 - \frac{2^{t-1}}{q}\right)^t.$$

Since $t \leq 1 + \log q$, we can bound this last expression by using the binomial inequality, obtaining $\Pr(A_t) \leq \frac{t 2^{t-1}}{q}$, and the result is proved. □

3.2.4.2 Security Proof for the Conjunctive PEKS Scheme $\mathcal{S}_1$

We dedicate this section to the proof of Theorem 3.24. As in [65], security is proved in the random oracle model by means of a sequence of hybrid games.

Given two documents $\mathbf{D}_0 = (w_{0,1}, \ldots, w_{0,m})$ and $\mathbf{D}_1 = (w_{1,1}, \ldots, w_{1,m})$, let $\Delta \subseteq [m]$ denote the positions corresponding to keywords in $\mathbf{D}_0 \Delta \mathbf{D}_1$. For $j \in [m]$ let Δ_j denote the first $\min(j, |\Delta|)$ elements of Δ.

Let G_0 be the security game defined in Section 3.2.2.5. Given $j \in [m]$ we define a hybrid game G_j, differring from G_0 only in that the keywords in positions in Δ_j of the challenge index are chosen uniformly at random by the challenger.

Specifically, we introduce the security game G_j for $j \in [m]$, consisting of the following five phases:

- *Setup.* The challenger runs Gen, hands over the public parameters and the public key pk to the adversary, and keeps the private key sk secret.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

- *Query Phase 1.* The adversary adaptively asks the challenger for q_T trapdoors of its own choice, where q_T is a polynomial value in the security parameter λ. We denote the set of all keywords queried in this phase by $\mathcal{W}_1$.
- *Challenge.* The adversary outputs two challenge candidate documents $\mathbf{D}_0$, $\mathbf{D}_1$, subject to the restriction that keywords appearing in $\mathbf{D}_0 \Delta \mathbf{D}_1$ have not been queried in Query Phase 1. That is, $(\mathbf{D}_0 \Delta \mathbf{D}_1) \cap \mathcal{W}_1 = \emptyset$. The challenger throws a fair coin $b \in \{0,1\}$ and computes the index $\mathbf{I} = (I_0, I_1, \ldots, I_m)$ corresponding to $\mathbf{D}_b$. Then, for every $i \in \Delta_j$, the challenger replaces I_i with uniformly sampled random elements from $\mathbb{G}_1$ and hands over this modified index to the adversary as the challenge.
- *Query Phase 2.* The adversary proceeds just as in Query Phase 1, but it is not allowed to ask for trapdoors containing keywords in $\mathbf{D}_0 \Delta \mathbf{D}_1$. That is, if the set of all keywords queried in this phase is $\mathcal{W}_2$, we impose $(\mathbf{D}_0 \Delta \mathbf{D}_1) \cap \mathcal{W}_2 = \emptyset$.
- *Guess.* The adversary outputs a guess $b' \in \{0,1\}$ for b.

Let $\mathrm{Adv}_{\mathcal{A},G_j}(\lambda)$ denote the advantage of the PPT adversary $\mathcal{A}$ in guessing b in the game G_j. It is clear that $\mathrm{Adv}_{\mathcal{A},G_m}(\lambda)$ is negligible in λ for every PPT adversary $\mathcal{A}$ because in G_m the two challenge candidate documents share the same information with the challenge index.

Note that G_0 is the security game defined in Section 3.2.2.5. We prove through Proposition 3.26 that the proposed conjunctive PEKS scheme $\mathcal{S}_1$ is semantically secure against adaptive chosen keyword attacks provided the DBDH assumption holds.

Proposition 3.26. *Assume that the* DBDH *assumption holds. For any $j \in \{0, \ldots, m-1\}$ and for any PPT adversary $\mathcal{A}$, the advantages of $\mathcal{A}$ in the games G_j and G_{j+1}, when using the scheme $\mathcal{S}_1$, are negligibly close in λ. That is,*

$$|\mathrm{Adv}_{\mathcal{A},G_j}(\lambda) - \mathrm{Adv}_{\mathcal{A},G_{j+1}}(\lambda)|$$

is negligible in λ.

Proof. Let $\mathcal{A}$ be a PPT adversary. For every $j \in \{0, \ldots, m-1\}$, we build a PPT DBDH distinguisher $\mathcal{B}_j$ taking a DBDH challenge tuple $(g, g^a, g^b, h, h^a, h^c, v)$ as input and interacting with $\mathcal{A}$ as the challenger in the security game of the scheme.

The distinguisher $\mathcal{B}_j$ is built in such a way that, for tuples with $v = e(g,g)^{abc}$, $\mathcal{A}$ is playing the game G_j, and for tuples with v random $\mathcal{A}$ is playing the game G_{j+1}. The output of the DBDH distinguisher $\mathcal{B}_j$ depends on the output of $\mathcal{A}$.

- *Setup.* The challenger $\mathcal{B}_j$ runs $\mathcal{S}_1.\mathrm{Gen}(\lambda)$ to obtain the public parameters of the scheme $\mathrm{params} = \{\mathbb{G}_1, \mathbb{G}_2, q, e, g, h, H, m\}$, where H is the hash oracle described below. $\mathcal{B}_j$ hands over the public parameters and the public key h^a to $\mathcal{A}$.
- *Hash Oracle.* The hash oracle H is operated by $\mathcal{B}_j$, and it maintains a list of tuples of the form $\langle w, s, c \rangle$ with $w \in \{0,1\}^*$, $s \in \mathbb{F}_q$ and $c \in \{0,1\}$. The list is initially empty. On input a keyword $w \in \{0,1\}^*$, the oracle H operates as follows:
  1. If there is an item in the list whose first element is keyword w, denote it by $\langle w, s, c \rangle$. Then:

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

(a) If $c = 0$, the oracle returns g^s.
(b) If $c = 1$, the oracle returns $\left(g^b\right)^s$.

2. If there is no item in the list whose first element is keyword w, then the oracle flips a coin $c \in \{0,1\}$ with $\Pr(c = 1) = 1/(2q_T m + 1)$, samples $s \in \mathbb{F}_q$ uniformly at random and inserts $\langle w, s, c\rangle$ into the list. Then, it proceeds to give an output as in the previous point.

- *Query Phase 1.* When $\mathcal{A}$ requests a trapdoor for keywords $\mathbf{L} = (w_1, \ldots, w_l)$ in keyword fields $J = \{j_1, \ldots, j_l\}$, the algorithm $\mathcal{B}_j$ first calls the oracle on input each keyword w_i and retrieves the associated oracle list tuples $\langle w_i, s_i, c_i\rangle$. Then, if some coin flip $c_i = 1$, $\mathcal{B}_j$ halts. Otherwise, $\mathcal{B}_j$ hands over to $\mathcal{A}$ the trapdoor $\mathbf{T}$ consisting of $T_0 = \prod_{i=1}^{l} (g^a)^{s_i}$ and J.

- *Challenge.* In this phase, the adversary $\mathcal{A}$ outputs a couple of documents $\mathbf{D}_0 = (w_{0,1}, \ldots, w_{0,m})$ and $\mathbf{D}_1 = (w_{1,1}, \ldots, w_{1,m})$ with the restrictions stated in the security game defined in Section 3.2.2.5 and above, and $\mathcal{B}_j$ throws a fair coin $b \in \{0,1\}$.

  Then, $\mathcal{B}_j$ calls the hash oracle on every keyword $w_{b,i}$ to fill the H-list with tuples $\langle w_{b,i}, s_{b,i}, c_{b,i}\rangle$. The algorithm $\mathcal{B}_j$ halts if:

  – For some $i \in [m]\backslash\Delta_{j+1}$ we have $c_{b,i} = 1$, or
  – $c_{b,t} = 0$, where $\{t\} = \Delta_{j+1}\backslash\Delta_j$.

  Then $\mathcal{B}_j$ samples a value $r \in \mathbb{F}_q$ uniformly at random, and computes the challenge $\mathbf{I} = (I_0, I_1, \ldots, I_m)$ in the following way

$$I_0 = h^r,$$
$$I_i = \begin{cases} \text{unif. sampled from } \mathbb{G}_T & \text{if } i \in \Delta_j \\ v^{r s_{b,i}} & \text{if } i \in \Delta_{j+1}\backslash\Delta_j \neq \emptyset \\ e\left((g^a)^r, (h^c)^{s_{b,i}}\right) & \text{if } i \in [m]\backslash\Delta_{j+1} \neq \emptyset \end{cases}$$

  and hands over $\mathbf{I}$ to $\mathcal{A}$.

- *Query Phase 2.* $\mathcal{B}_j$ proceeds as in Query Phase 1.

- *Guess.* The adversary $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$ for b. If $b = b'$, $\mathcal{B}_j$ outputs 1, and if $b \neq b'$, $\mathcal{B}_j$ outputs 0.

Since the DBDH assumption holds, $\mathrm{Adv}_{\mathcal{B}_j}(\lambda)$ must be negligible in λ. But

$$\begin{aligned} \mathrm{Adv}_{\mathcal{B}_j}(\lambda) =& |\Pr(\mathcal{B}_j(X) = 1 | X = 1) - \Pr(\mathcal{B}_j(X) = 1 | X = 0)| \\ =& \Pr(\mathcal{B}_j \text{ does not halt}) \cdot |\mathrm{Adv}_{\mathcal{A},G_j}(\lambda) - \mathrm{Adv}_{\mathcal{A},G_{j+1}}(\lambda)|. \end{aligned}$$

By Lemma 3.27, $\Pr(\mathcal{B}_j$ does not halt) is non-negligible in λ, and the result is proved. □

As a consequence of this result, we conclude the proof of Theorem 3.24. We next state and prove the lemma referenced in the proof of Proposition 3.26, which is an adaptation of a result in [38].

Lemma 3.27 ([38])**.** *The probability that algorithm $\mathcal{B}_j$ does not halt is non-negligible in the security parameter λ.*

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Proof. We split the calculations between the query phases and the challenge phase.

In each of the query phases, we allow $\mathcal{A}$ to ask for a polynomial amount q_T (in λ) of trapdoor queries. This amounts to throwing at most $2mq_T$ coins c with $\Pr(c = 1) = 1/(2q_T m + 1)$. Since $\mathcal{B}_j$ does not halt exactly when each and every one of these throws outcome is 0, we have

$$\Pr(\mathcal{B}_j \text{ does not halt in query phases}) \geq \left(1 - \frac{1}{2mq_T + 1}\right)^{2mq_T} \geq 1/e,$$

which is non-negligible in λ.

For the challenge phase, $\mathcal{B}$ does not halt exactly when the coin throw corresponding to the keyword in position $\Delta_{j+1} \backslash \Delta_j$ (if nonempty) of the chosen challenge document is 1 and the coin throws corresponding to the keywords in positions in $[m] \backslash \Delta_{j+1}$ of the chosen challenge document are all 0. Since, if $\mathbf{D}_0 \neq \mathbf{D}_1$ then $|[m] \backslash \Delta_{j+1}| \leq m - 1$, we have:

$$\Pr(\mathcal{B}_j \text{ does not halt in the challenge phase}) \geq \left(1 - \frac{1}{2mq_T + 1}\right)^{m-1} \frac{1}{2mq_T + 1} \geq \frac{1}{e} \cdot \frac{1}{2mq_T + 1},$$

which is non-negligible in λ since m is constant in λ and q_T is polynomial in λ, and we get the stated lemma. □

3.2.5 Subset PEKS Scheme

The second PEKS scheme we propose enables a class of subset queries. This class includes subset queries as defined in [65].

Subset queries, as understood in [65], are specified by an ordered tuple of m sets of keywords $(A_1, \ldots, A_m)$. Then, a document $\mathbf{D} = (w_1, \ldots, w_m)$ satisfies such a query if and only if the predicate $(w_1 \in A_1) \wedge \cdots \wedge (w_m \in A_m)$ holds. The scheme we propose considers subsets in a partition of $\mathbf{D}$ instead of keywords w_i in this last predicate.

More concretely, in the Gen algorithm we fix a partition $J_1, \ldots, J_m$ of $[m]$. Given a document $\mathbf{D} = (w_1, \ldots, w_m)$, write $B_i = \{w_j\}_{j \in J_i}$ for every $i \in [m]$. Given a query $\mathbf{L} = (w'_1, \ldots, w'_l)$, $J = \{j_1, ..., j_l\}$, where J is written in increasing order, consider $A_i = \{w'_k\}_{j_k \in J_i}$ for every $i \in [m]$. Then, the document $\mathbf{D}$ satisfies the query $\mathbf{L}, J$ if and only if the predicate $(B_1 \subseteq A_1) \wedge \cdots \wedge (B_m \subseteq A_m)$ holds. Note that we also admit empty keyword fields in documents, which are denoted by keywords $\perp$.

For the sake of clarity, before formally stating the proposed construction, we give a brief example illustrating the internal workings of the scheme.

The Gen algorithm of the scheme fixes a tuple of possibly repeated *field identifiers* $(f_1, \ldots, f_m)$. We take $m = 8$ and $(f_1, \ldots, f_8) = (1, 1, 1, 2, 2, 2, 3, 3)$ as an example.

When encrypting documents in Enc, the documents $\mathbf{D} = (w_1, \ldots, w_m)$ can be thought of as a collection of sets of keywords, where keywords in positions having the same field identifier belong to the same set. Also, the keyword $\perp$ is allowed, and it stands for

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

a null entry. For instance, following the example above, the document $\mathbf{D}$ defined by $\mathbf{D} = (w_1, w_2, w_3, w_4, w_5, \perp, w_7, \perp)$ can be thought of as the following collection of sets $(\{w_1, w_2, w_3\}, \{w_4, w_5\}, \{w_7\})$.

When generating trapdoors, we input a query consisting of a tuple of keywords $\mathbf{L} = (w_1, \ldots, w_l)$ and a set of positions $J = \{j_1, \ldots, j_l\}$ written in increasing order. As above, keywords in positions having the same field identifier are thought to belong to the same set. Thus, in the example above, the query for words $\mathbf{L} = (w'_1, w'_2, w'_3, w'_4, w'_5, w'_6, w'_7)$ at positions $J = \{1, 2, 3, 4, 5, 6, 8\}$ can be thought of as the collection of sets composed by $(\{w'_1, w'_2, w'_3\}, \{w'_4, w'_5, w'_6\}, \{w_7\})$.

Now, a document matches a query in the Search algorithm exactly when the sets of keywords defined by the document are contained in the sets of keywords defined by the query, in a sequential way. That is, following the example above, a match happens exactly when

$$(\{w_1, w_2, w_3\} \subseteq \{w'_1, w'_2, w'_3\}) \wedge (\{w_4, w_5\} \subseteq \{w'_4, w'_5, w'_6\}) \\ \wedge (\{w_7\} \subseteq \{w'_7\}).$$

We now describe the proposed subset PEKS scheme. Although not stated, in the following every algorithm apart from $\mathcal{S}_2$.Gen takes the public parameters as input.

Definition 3.28. We define a public-key encryption with subset keyword search scheme $\mathcal{S}_2$ by means of the following four polynomial-time algorithms:

$\mathcal{S}_2$.Gen(λ)**:** Given a security parameter $\lambda \in \mathbb{Z}$, fix a symmetric bilinear group $\mathbb{G}$ of prime order $q \geq 2^\lambda$ and denote the corresponding pairing by $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Let g be a random generator of $\mathbb{G}$. Let $H : \{0,1\}^* \to \mathbb{G}$ and $H_1 : \mathbb{G}_T \to \{0,1\}^*$ be collision-resistant hash functions. Set $m \in \mathbb{Z}$ the maximum number of keywords in every document, which we assume constant in λ and satisfying $m \leq (1 + \log q)/2$. Define a tuple $(f_1, \ldots, f_m)$ of possibly repeated field identifiers describing which field does each word in the documents belong to, where each $f_i \in [m]$. Choose $a \in \mathbb{F}_q$ uniformly at random. Output the public parameters $\text{params} = \{\mathbb{G}, q, e, g, H, H_1, m, (f_1, \ldots, f_m)\}$, the private key $\beta = (\beta_i)_{i=1}^m$ and the public key $\alpha = (\alpha_i)_{i=1}^m$, where $\beta_i = a^{-i}$ and $\alpha_i = g^{a^i}$.

$\mathcal{S}_2$.Enc$_\alpha(\mathbf{D})$**:** Given as input $\mathbf{D} = (w_1, \ldots, w_m)$ the document consisting of a tuple of m keywords w_i in the domain $\{0,1\}^* \cup \{\perp\}$, generate $r_1, \ldots, r_m \in \mathbb{F}_q$ uniform random nonces in such a way that $f_i = f_j$ implies $r_i = r_j$.

Set

$$I_0 = H_1\left(e\left(\textstyle\prod_{i \in [m]: w_i \neq \perp} H(w_i)^{r_i}, g\right)\right) \\ I_i = \alpha_i^{r_i} \quad \text{for } i \in [m].$$

Output the index $\mathbf{I} = (I_0, I_1, \ldots, I_m)$.

$\mathcal{S}_2$.Trapdoor$_\beta(\mathbf{L}, J)$**:** Given $\mathbf{L} = (w_1, \ldots, w_l)$ the input tuple of keywords with $l \leq m$, and the set of keyword fields $J = \{j_1, \ldots, j_l\} \subseteq [m]$ written in increasing order, set

$$T_i = H(w_i)^{\beta_{j_i}} \quad \text{for } i \in \{1, \ldots, l\}.$$

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Output the trapdoor $\mathbf{T}$, consisting of $T_1, \ldots, T_l$ along with the fields J to be queried.

$\mathcal{S}_2$.Search$(\mathbf{I}, \mathbf{T})$**:** Parse the index $\mathbf{I}$ as $\mathbf{I} = (I_0, I_1, \ldots, I_m)$ and the trapdoor $\mathbf{T}$ as $\mathbf{T} = (T_0, \{j_1, \ldots, j_l\})$. For every $t \in [m]$, let J_t denote the set of elements $i \in [l]$ such that $f_{j_i} = t$. For every $i \in [l]$, compute $v_i = e(T_i, I_{j_i})$.
Output 1 if there exists subsets $J'_t \subseteq J_t$ for $t \in [m]$ such that

$$I_0 = H_1\left(\prod_{t=1}^{m}\prod_{i\in J'_t} v_i\right).$$

Otherwise output 0.

We next give the consistency and security theorems for our scheme. The proofs are deferred to Sections 3.2.6.1 and 3.2.6.2, respectively.

Theorem 3.29. *The proposed subset PEKS scheme $\mathcal{S}_2$ is computationally consistent under the random oracle model.*

Theorem 3.30. *Assume that the $(m+1)-$*BDHI *assumption holds. Then, the proposed subset PEKS scheme $\mathcal{S}_2$ is semantically secure against adaptive chosen keyword attacks under the random oracle model.*

3.2.6 Consistency and Security Proofs for the Subset PEKS Scheme $\mathcal{S}_2$

In this section, we give the consistency and security proofs for the subset PEKS scheme $\mathcal{S}_2$.

3.2.6.1 Consistency Proof for the Subset PEKS Scheme $\mathcal{S}_2$

We dedicate this section to the proof of Theorem 3.29.

We prove consistency in the random oracle model of the scheme $\mathcal{S}_2$ in a similar fashion than in the proof by Abdalla et al. in [119].

Let $\mathcal{A}$ be a PPT adversary in the consistency game defined in Section 3.2.2.3, which has access to the public parameters, to the public key pk and to the hash oracles H, H_1, which are modeled as random oracles. Let WSet, TSet be the sets of polynomial (in λ) size q_H, q_{H_1} which consist of keywords queried to the hash oracles H, H_1 throughout the game, respectively. Write $(f_1, \ldots, f_m)$ the tuple of field identifiers in the public parameters. Let $\mathbf{D} = (w_1, \ldots, w_m)$ and $\mathbf{L} = (w'_1, \ldots, w'_l)$ and $J = \{j_1, \ldots, j_l\} \subseteq [m]$ (written in increasing order) denote the guess of $\mathcal{A}$ in the *Guess* phase.

For $j \in [m]$, let $\mathbf{D}_{f_j}$ denote the set of keywords in $\mathbf{D}$ at positions having field identifier f_j. Let $\tilde{J}$ be the set of positions $j_i \in J$ such that $w'_i \not\subseteq \mathbf{D}_{f_{j_i}}$. Without loss of generality, we rule out adversaries choosing $\tilde{J} = \emptyset$ in the *Guess* phase. Let $r_1, \ldots, r_m \in \mathbb{F}_q$ denote the random nonces generated by $\mathcal{A}$ in the searchable index generation of the *Output* phase.

Denote $X = e\left(\prod_{i\in[m]:w_i\neq\perp} H(w_i)^{r_i}, g\right)$ and, given a subset $J' \subset J$, denote $X' = e\left(\prod_{j_i\in J'} H(w'_i)^{r_{j_i}}, g\right)$. Now note that the output of $\mathcal{A}$ in the consistency game is 1 if and only if $X = X'$ or $H_1(X) = H_1(X')$ for some $J' \subseteq J$ with $J' \cap \tilde{J} \neq \emptyset$.

Let E denote the event that there exist $\mathbf{D} = (w_1, \ldots, w_m)$, $\mathbf{L} = (w'_1, \ldots, w'_l)$ and $J = \{j_1, \ldots, j_l\} \subseteq [m]$, among all possible guesses taking words in WSet in such a way that we have $\prod_{i\in[m]:w_i\neq\perp} H(w_i)^{r_i} = \prod_{i\in[l]} H(w'_i)^{r_{j_i}}$. Likewise, let E_1 be the event that there exist $T, T' \in$ TSet in such a way that $H_1(T) = H_1(T')$.

If all $r_1, \ldots, r_m \neq 0$, then $X = X'$ has nonzero probability of happening only when E happens (note that by ranging over all possible J we remove the need to include the J' above in the argument). Likewise $H_1(X) = H_1(X')$ has nonzero probability of happening only when E_1 happens. Therefore

$$\mathrm{Adv}_{\mathcal{A}}(\lambda) \leq \left(1 - \frac{m}{q}\right)(\Pr(E) + \Pr(E_1)) + \frac{m}{q}.$$

Since $q \geq 2^\lambda$ and m is constant in λ, it suffices to prove that $\Pr(E)$ and $\Pr(E_1)$ are negligible in λ.

By computing the probability of the complementary and using the binomial inequality, we see that $\Pr(E_1) \leq q_{H_1}^2/q$. Now, since H is modeled as a random oracle and since inversion permutes group elements, by using Lemma 3.25 we see that $\Pr(E) \leq q_H^{2m}\frac{m2^{2m}}{q}$. The obtained bounds are indeed negligible in λ, since $q \geq 2^\lambda$, and m, q_H are assumed to be constant and polynomial in λ, respectively.

As a consequence of this result, we conclude the proof of Theorem 3.29.

3.2.6.2 Security Proof for the Subset PEKS Scheme $\mathcal{S}_2$

We dedicate this section to the proof of Theorem 3.30.

We prove security in the random oracle model by proceeding similarly to [38].

Suppose that there exists a PPT adversary $\mathcal{A}$ breaking the security game defined in Section 3.2.2.5 with advantage not negligible in λ. We then build a successful PPT $(m+1)-$BDHI distinguisher $\mathcal{B}$ taking an $(m+1)-$BDHI challenge tuple $(g, g^a, \ldots, g^{a^{m+1}})$ as input. By interacting with $\mathcal{A}$ as the challenger in the security game defined in Section 3.2.2.5, $\mathcal{B}$ computes $e(g,g)^{1/a}$ with non-negligible advantage in λ.

- *Setup.* The challenger $\mathcal{B}$ runs $\mathrm{Gen}(\lambda)$ and generates the public parameters of the scheme $\mathrm{params} = \{\mathbb{G}, q, e, g, H, H_1, m, (f_1, \ldots, f_m)\}$, where H, H_1 are handles to the hash oracles described below. $\mathcal{B}$ hands over the public parameters and the public key $(g^a, \ldots, g^{a^m})$ to $\mathcal{A}$.
- *Hash Oracle H.* The oracle is operated by $\mathcal{B}$, which maintains a list of tuples of the form $\langle w, s, c\rangle$ with $w \in \{0,1\}^*$, $s \in \mathbb{F}_q$ and $c \in \{0,1\}$. The list is initially empty. On input a keyword $w \in \{0,1\}^*$, the oracle H operates as follows:
  1. If there is an item in the list whose first element is keyword w, denote it by $\langle w, s, c\rangle$. Then:

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

(a) If $c = 0$, the oracle returns g^s.

(b) If $c = 1$, the oracle returns $\left(g^{a^{m+1}}\right)^s$.

2. If there is no item in the list whose first element is keyword w, then the oracle flips a coin $c \in \{0,1\}$ with $\Pr(c = 1) = 1/(2q_T m + 1)$, samples $s \in \mathbb{F}_q$ uniformly at random and inserts $\langle w, s, c\rangle$ into the list. Then, it proceeds to give an output as in the previous point.

- *Hash Oracle* H_1. The oracle is operated by $\mathcal{B}$, which maintains a list of tuples of the form $\langle t, V\rangle$ with $t \in \mathbb{G}_T$ and $V \in \{0,1\}^*$. The list is initially empty. On input an element $t \in \mathbb{G}_T$, the oracle H_1 operates as follows:

  1. If there is an item in the list whose first element is t, denote it by $\langle t, V\rangle$. The oracle returns V.
  2. If there is no item in the list whose first element is t, then the oracle samples $V \in \mathbb{G}_T$ uniformly at random and inserts $\langle t, V\rangle$ into the list. Then, it proceeds to give an output as in the previous point.

- *Query Phase 1.* When $\mathcal{A}$ requests a trapdoor for keywords $\mathbf{L} = (w_1, \ldots, w_l)$ in positions $J = \{j_1, \ldots, j_l\}$ written in increasing order, the algorithm $\mathcal{B}$ first calls the H oracle on input each keyword w_i and retrieves the associated oracle list tuples $\langle w_i, s_i, c_i\rangle$. Then, if some coin flip $c_i = 1$, $\mathcal{B}$ halts. Otherwise, $\mathcal{B}$ hands over to $\mathcal{A}$ the trapdoor $\mathbf{T}$ consisting of $T_i = (g^{a^{m-j_i+1}})^{s_i}$ for $i \in \{1, \ldots, l\}$, and J.

- *Challenge.* The adversary outputs two documents $\mathbf{D}_0 = (w_{0,1}, \ldots, w_{0,m})$, $\mathbf{D}_1 = (w_{1,1}, \ldots, w_{1,m})$ with the restrictions stated in the security game defined in Section 3.2.2.5, and $\mathcal{B}$ throws a fair coin $b \in \{0,1\}$.

  Then, $\mathcal{B}$ calls the hash oracle on every keyword $w_{b,i}$ to fill the H-list with tuples $\langle w_{b,i}, s_{b,i}, c_{b,i}\rangle$. The algorithm $\mathcal{B}$ halts if some $c_{b,i} = 1$.

  Then $\mathcal{B}$ uniformly chooses $J \in \mathbb{G}_T$ and random nonces $r_1, \ldots, r_m \in \mathbb{F}_q$ in such a way that if $f_i = f_j$ then $r_i = r_j$, and it computes the challenge $\mathbf{I} = (I_0, I_1, \ldots, I_m)$ in the following way

  $$I_0 = J, \quad I_i = \left(g^{a^{i-1}}\right)^{r_i}$$

  and hands over $\mathbf{I}$ to $\mathcal{A}$. In addition, $\mathcal{B}$ halts if $\sum_i s_{b,i} r_i \equiv 0 \pmod q$ and, if not, it stores $C = (\sum_i s_{b,i} r_i)^{-1} \pmod q$.

- *Query Phase 2.* $\mathcal{B}$ proceeds as in Query Phase 1.

- *Guess.* The adversary $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$ for b. Then, $\mathcal{B}$ picks a random element $\langle t, V\rangle$ from the list in H_1, and returns t^C.

Note that the challenge is well-formed, and that it implicitly imposes the equality $J = H_1\left(e\left(g^{\sum_i s_{b,i} r_i}, g\right)^{1/a}\right)$. If $\mathcal{B}$ does not halt, then it perfectly simulates a real attack game up until the moment when $\mathcal{A}$ issues an H_1 oracle query for $t_0 = e\left(g^{\sum_i s_{0,i} r_i}, g\right)^{1/a}$ or $t_1 = e\left(g^{\sum_i s_{1,i} r_i}, g\right)^{1/a}$.

Let $\mathcal{E}$ denote the event that $\mathcal{A}$ issues a query for t_0 or t_1 in a real attack game. We now lower bound the probability of $\mathcal{E}$. Under the random oracle model, if $\mathcal{E}$ does not

happen, then $\mathcal{B}$ does not reveal any information about b. Therefore,

$$\begin{aligned}\Pr(b' = b) &= \Pr(\mathcal{E})\Pr(b' = b|\mathcal{E}) + \frac{1}{2}\Pr(\neg\mathcal{E}) \\ &= \frac{1}{2} + \Pr(\mathcal{E})\left(\Pr(b' = b|\mathcal{E}) - \frac{1}{2}\right)\end{aligned}$$

so we can express the advantage of $\mathcal{A}$ by

$$\begin{aligned}\mathrm{Adv}_{\mathcal{A}}(\lambda) &= |\Pr(b' = b) - \frac{1}{2}| \\ &= \Pr(\mathcal{E}) \cdot \left|\Pr(b' = b|\mathcal{E}) - \frac{1}{2}\right| \leq \frac{1}{2}\Pr(\mathcal{E})\end{aligned}$$

and $\Pr(\mathcal{E}) \geq 2\mathrm{Adv}_{\mathcal{A}}(\lambda)$.

Now, suppose that

1. $\mathcal{B}$ does not abort,
2. $\mathcal{A}$ eventually issues an H_1 oracle query for either t_0 or t_1, i.e. $\mathcal{E}$ happens, and
3. $\mathcal{B}$ chooses b such that $\mathcal{A}$ queries the H_1 oracle for t_b (this is well defined, since $\mathcal{A}$ does not receive any information about b until $\mathcal{E}$ happens).

Then, if $\mathcal{B}$ calls the hash oracle H_1 on input t_b in the Guess phase, it successfully computes $e\,(g,g)^{1/a}$. Since $\mathcal{B}$ uniformly samples an element from all inputs processed by the hash oracle H_1 to generate its output in the *Guess* phase, the probability of $\mathcal{B}$ breaking the $(m+1)-\mathrm{BDHI}$ assumption in the above situation is at least $1/q_{H_1}$, where q_{H_1} is the polynomial amount of queries issued to the H_1 oracle. This implies

$$\begin{aligned}\mathrm{Adv}_{\mathcal{B}}(\lambda) &= \Pr(\mathcal{B}(g,\ldots,g^{a^{m+1}}) = e\,(g,g)^{1/a}) \\ &\geq \frac{1}{q_{H_1}}\Pr(\mathcal{B} \text{ does not abort})\Pr(\mathcal{E})\frac{1}{2} \\ &\geq \frac{1}{q_{H_1}}\Pr(\mathcal{B} \text{ does not abort})\mathrm{Adv}_{\mathcal{A}}(\lambda).\end{aligned}$$

By following the same argument as in Lemma 3.27, we see that $\Pr(\mathcal{B}$ does not abort) is non-negligible in λ. Since $\mathcal{A}$ breaks the security game defined in Section 3.2.2.5 with non-negligible advantage and q_{H_1} is polynomial in λ, we conclude that $\mathrm{Adv}_{\mathcal{B}}(\lambda)$ is non-negligible as well.

As a consequence of this result, we conclude the proof of Theorem 3.30.

3.2.7 Efficiency Analysis

We next lay out the efficiency measures of the proposed schemes and of other similar searchable encryption schemes. We state the size and the time needed to generate an encrypted index and a trapdoor, and also the time taken to perform a search operation.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

We omit multiplication time, hash evaluation time, key setup time, field identifiers size and key storage size in the efficiency analysis. Notice that the search time refers to performing a search operation for a single encrypted index. This is the case of the application examples stated in Section 3.2.1, where the search time over a set of encrypted indexes scales linearly in the number of encrypted indexes.

To analyze performance, we implement our schemes $\mathcal{S}_1$ and $\mathcal{S}_2$ and the schemes in [74] by using the PBC library [135], and we provide the estimated running times for each algorithm. All simulations ran on an Intel® Core™ i7-4510U CPU at 2.00GHz and 8GB memory under Ubuntu 16.04.1 LTS.

For the sake of comparison, we use symmetric bilinear groups (*type A* pairings in the PBC library documentation) in all implementations. Also, as suggested in the PBC library documentation [135], we fix a 512-bit base field order and a 160-bit group order to instantiate the bilinear group. In our implementations, we did not use pre-processing or any of the functions that the PBC library provides in order to speed up computations.

3.2.7.1 Conjunctive PEKS Scheme

Since we restrict the analysis to conjunctive PEKS schemes, the schemes [43, 47, 66, 118, 131] lie out of the scope of the analysis. However, we include the single-keyword PEKS scheme [38] by Boneh et al. in our analysis for the sake of comparison, extending it to the conjunctive case by considering the concatenation of indexes and trapdoors and the sequential evaluation of the Search algorithm. Note that, for the single-keyword case $m = l = 1$, our scheme and [38] have equivalent efficiency marks.

We also restrict the analysis to the public-key setting, so we leave out schemes such as [40, 44, 116]. Other schemes such as [41, 130] are omitted due to security considerations.

The size and time efficiency measures can be found in Tables 3.4 and 3.5 respectively.

Scheme	Index size	Trapdoor size
[38] BDOP	$mE + mF$	lE
[74] PKL I	$2E + mF$	E
[74] PKL II	$2mE$	$2E$
[73] PCL	$(2m + 1)E$	$2E + F$
[65] BW	$(2m + 2)E$	$(2l + 1)E$
[42] HL	$(m + 2)E$	$3E$
$\mathcal{S}_1$	$E + mF$	E

m: number of keywords in the index
l: number of keywords in the trapdoor
E: size of elliptic curve point
F: size of finite field element

TABLE 3.4: Size efficiency comparison of conjunctive PEKS schemes.

Note that the proposed scheme achieves the lowest efficiency marks for index size, trapdoor size, trapdoor generation time and search time. This is not so for the index generation time, which is constrained by the computing power of the senders. However, in many applications the search time and the index and trapdoor size measures are far more critical, since they are constrained by the throughput of the network and by the

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Scheme	Index time	Trapdoor time	Search time
[38] BDOP	$me + 2mG$	lG	le
[74] PKL I	$me + (m+2)G$	G	$e + G$
[74] PKL II	$(4m+1)G + X$	$2G$	$2e + G$
[73] PCL	$(4m+1)G + 2X$	$2G$	$2e$
[65] BW	$(6m+2)G$	$(5l+1)G$	$(2l+1)e$
[42] HL	$(2m+2)G$	$3G$	$3e$
$\mathcal{S}_1$	$me + 2G_2$	G_1	e

m: number of keywords in the index
l: number of keywords in the trapdoor
e: pairing evaluation time
G: exponentiation time in symmetric bilinear group $\mathbb{G}$
G_1: exp. time in asymmetric bilinear group $\mathbb{G}_1$
G_2: exp. time in asymmetric bilinear group $\mathbb{G}_2$
X: exponentiation time in finite field

TABLE 3.5: Time efficiency comparison of conjunctive PEKS schemes.

computing power of the storage server, and search operations may be executed several times per encrypted index.

In Table 3.6, we give the estimated running times in milliseconds for Table 3.5 by arbitrarily fixing $m = 8$ and $l = 8$, and using 48-bit keywords. For implementation reasons, we do not give the performance analysis for the schemes in [42, 73], mainly because they require the evaluation of multiple independent hash functions. We also omit the analysis of the conjunctive PEKS scheme in [65] for efficiency reasons.

Scheme	Index time	Trapdoor time	Search time
[38] BDOP	58.7ms	36.5ms	7.33ms
[74] PKL I	53.1ms	29.7ms	2.73ms
[74] PKL II	59.8ms	3.08ms	3.26ms
$\mathcal{S}_1$	35.9ms	26.5ms	0.94ms

TABLE 3.6: Performance analysis of conjunctive PEKS schemes.

We observe that $\mathcal{S}_1$ achieves the best index computation and search time in the studied case. Also, the second scheme in [74] gives the best trapdoor computation time. This is achieved in [74] by removing the need for an *admissible encoding scheme*, thus replacing products in the bilinear group by sums in the underlying finite field.

3.2.7.2 Subset PEKS Scheme

We now give the efficiency measures for the proposed subset PEKS scheme and the related subset PEKS scheme by Boneh and Waters [65]. For the sake of comparison, we assume that subset queries are always as defined in [65]. See the beginning of Section 3.2.5 for more details.

One of the main differences between both schemes is that in the proposed scheme the keyword space is an arbitrary exponential-sized keyword space $\{0,1\}^*$, while in [65]

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

keywords are taken from a small polynomial-sized keyword space. We denote by n the size of this keyword space in the efficiency analysis. Another difference is that, in the scheme we propose, the number of keywords in queries is limited at $\mathcal{S}_2$.Setup. We denote by L the maximum number of keywords in a trapdoor. The size and time efficiency measures can be found in Tables 3.7 and 3.8 respectively.

Scheme	Index size	Trapdoor size
[65] BW	$(2nm+2)E$	$(2nm+1)E$
$\mathcal{S}_2$	$LE+H$	lE

n: size of the keyword space
m: number of keywords in the index
l: number of keywords in the trapdoor
L: maximum number of keywords in a trapdoor
E: size of elliptic curve point
F: size of finite field element
H: size of strings in $\mathrm{Im}(H_1) \subseteq \{0,1\}^*$

TABLE 3.7: Size efficiency comparison between subset PEKS schemes.

Scheme	Index time	Trapdoor time	Search time
[65] BW	$(6nm+2)G$	$(5nm+1)G$	$(2nm+1)e$
$\mathcal{S}_2$	$e+2LG$	lG	le

n: size of the keyword space
m: number of keywords in the index
l: number of keywords in the trapdoor
L: maximum number of keywords in a trapdoor
e: pairing evaluation time
G: exponentiation time in symmetric bilinear group $\mathbb{G}$
X: exponentiation time in finite field

TABLE 3.8: Time efficiency comparison between subset PEKS schemes.

We now give the estimated running times in milliseconds for the $\mathcal{S}_2$ scheme. Since the performance of the subset PEKS scheme in [65] depends strongly on the size of the keyword space, it is difficult to choose parameters allowing a sensible comparison. Therefore, we omit the performance analysis of [65].

We arbitrarily fix $m = 8$, $l = 8$, and use 48-bit keywords. We also instantiate the index and the trapdoor so that the search operation takes the longest possible. In the proposed scheme, the computation of an index and a trapdoor takes an estimated time of 39.5ms and 36.6ms respectively, and the search time is approximately 7.57ms.

3.2.8 Conclusion

PEKS schemes enable public key holders to encrypt documents, while the secret key holder is able to generate queries for the encrypted data. In our work we present two PEKS schemes enabling conjunctive and subset queries. We propose a security notion for PEKS and we prove the proposed schemes secure under the asymmetric DBDH assumption and the $p-$BDHI assumption, respectively. We also prove the computational

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

consistence of the given constructions. The main strength of our schemes lies in their efficiency since, as shown in the provided efficiency analysis, they improve all previous related schemes in some of the most critical operations.

The proposed schemes could possibly admit various extensions. For example, we believe it is possible to extend our subset PEKS scheme to allow decryption of searchable indexes by embedding messages in the target group, as done in works such as [65]. Such an extension would simplify the retrieval of messages in the search process.

In [119], Abdalla et al. prove computational consistency for the PEKS scheme [38] by Boneh et al., and give a modified scheme achieving the stronger notion of statistical consistency. Since the conjunctive PEKS scheme we propose here can be seen as a natural extension to the scheme in [38] to the conjunctive case, it would be interesting to find similar modifications that improve consistency.

It would also be interesting to maintain a good efficiency and security trade-off while improving the security notion. For example, by providing tight security proofs in the standard model, or by removing the need for a secure channel for trapdoors.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Chapter 4

Outsourcing Kriging Interpolation Computations

In this chapter, we present our contributions to secure outsourced computation. Due to requirements in the *CLARUS* project, we aim to securely outsource Kriging interpolation computations.

Kriging is a spatial interpolation algorithm which provides the best unbiased linear prediction of an observed phenomena by taking a weighted average of samples within a neighborhood. It is widely applied in areas such as geo-statistics where, for example, it may be used to predict the quality of mineral deposits in a location based on previous sample measurements. In this chapter, we present a method for the private outsourcing of Kriging interpolation using a tailored modification of the Kriging algorithm in combination with homomorphic encryption. Our solution allows crucial information relating to measurement values to be hidden from the cloud service provider.

4.1 Introduction

Kriging [18–21] is a well-recognized form of linear interpolation that gives a prediction z_0^* of the value z_0 of some phenomena at an unobserved location (x_0, y_0) in a two-dimensional region. The quality of a Kriging prediction relies on some *variogram parameters*, which reflect the assumption that measurements taken at nearby locations are more likely to be similar than measurements taken far apart. Such parameters must be carefully selected prior to interpolation. The prediction is then formed as a weighted sum of measurement values, where measurements taken close to (x_0, y_0) are given a greater weight than those far away. Kriging was designed with geo-statistical applications in mind, for instance to predict the best location to mine based on the mineral deposits found at previous boreholes within a region, but it has also found applications in a variety of settings including remote sensing, real-estate appraisal and computer simulations.

Kriging has been identified as a good candidate process to be outsourced, based on the practical and legislative requirements of industrial users (for instance, [1, 75]). Many users may need access to a Kriging prediction service since, for example, legal frameworks may require such data to be shared among relevant authorities [136]). A secured storage

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

server may be preferable to distributing copies of the entire data set to each authorized user, especially when data sets are large and/or user devices are constrained. Further, Kriging might need to be performed over data owned by multiple organizations, with an independent cloud service provider performing processing duties on behalf of all concerned parties. Centralized outsourcing also makes sense when remote sensors take frequent measurements and push the results to a central database.

Consider a client $\mathcal{C}$ that owns a Kriging data set, consisting of measurements taken at various locations, which it wishes to outsource to an honest-but-curious cloud service provider $\mathcal{S}$. Client $\mathcal{C}$ would like to make use of both the storage and computational power of $\mathcal{S}$ to make a Kriging service on its data set available to multiple users. Further, other *data generating nodes* may be authorized by $\mathcal{C}$ to add and remove data to and from the outsourced data set.

A trivial solution consists of encrypting all data using a symmetric encryption scheme and using the server only for Storage-as-a-Service. To compute a Kriging prediction, all relevant data is retrieved, decrypted and computed on locally. Unfortunately, this solution may not be efficient, particularly if client devices have limited computational power or storage capacity, and require a high bandwidth during queries. This may be an issue if, for example, a surveyor in the field requires an on-line Kriging prediction service. Mobile data services may be expensive, intermittently available or slow.

An alternative is to compute the entire Kriging process on encrypted data by encrypting all data using Fully Homomorphic Encryption (FHE)[1]. Unfortunately, Kriging involves several computations that are currently challenging when using FHE, including computing square roots and natural exponentiations. It is possible to outsource the Kriging process and protect all information using FHE. However this results in prohibitively high encryption and decryption costs, as well as a large amount of interactivity and local computations, which may diminish the benefits of cloud computing. Preliminary experiments using the SEAL library [137], admittedly without optimization of code or parameter choices, did not yield promising results when computing a Kriging prediction using a data set of more than three measurements. While the use of FHE schemes should be explored further in future works, particularly to reflect advances in FHE schemes, we show in this work that such schemes are not strictly required in this setting.

Our proposed solution uses additive homomorphic encryption to outsource Kriging interpolation efficiently. We make a trade-off by protecting only the most sensitive parameters. That is, we protect the measurement values in the data set, the generated Kriging predictions and the variogram parameters chosen by the client. We do not hide locations of measurements or queries, noting that measurement locations may well be externally observable (e.g. if measurements come from previous mining operations).

Our main contribution is to show that the Kriging process can be adapted in such a way that the sensitive variogram parameters may be factored out from the online computation, while the remainder of the Kriging computation may be performed on encrypted measurement values using an additively homomorphic encryption scheme. We thus gain a practical, efficient and secure solution to outsourced, private Kriging interpolation. An outline of our protocol is as follows:

[1]In fact, it suffices to consider Somewhat Homomorphic Encryption rather than FHE as the functionality is fixed and has a reasonably low multiplicative depth.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

1. The client $\mathcal{C}$ uploads an encrypted data set, comprising n measurements, to the server $\mathcal{S}$. The cost of this step is $O(n)$ due to encryption of the measurement values.

2. The server $\mathcal{S}$ prepares the Kriging data set for future queries. This process comprises plaintext operations that are also necessary in an unprotected outsourced Kriging scheme.

3. The client $\mathcal{C}$ makes a query to $\mathcal{S}$ requesting a Kriging prediction at a location (x_0, y_0). This is done in plaintext with virtually no cost.

4. The server $\mathcal{S}$ computes the interpolation on encrypted measurements. The cost with respect to an unprotected outsourced Kriging scheme is increased by $O(n)$, due to operations over encrypted data.

5. The client $\mathcal{C}$ decrypts the result.

Cryptographically-secured Kriging was previously studied in a different setting, where a server owns a data set and clients may query the data set at a previously unsampled location [76]: the queried location and the resulting prediction should remain protected from the server, while the data set held by the server should be protected from the client. Two solutions are proposed in [76] which, unlike our solution, support only one variogram model and require high communication complexity, interactivity and local computation. The first is based on creating random dummy queries to hide the queried location, and using an oblivious transfer protocol to hide predictions for all but the legitimate query location. The second solution uses the Paillier encryption scheme in an interactive protocol requiring multiple round-trips between client and server. In [77] collaborative private Kriging was investigated, where users combine their data sets to achieve more accurate Kriging predictions.

The remainder of this chapter is structured as follows. In Section 4.2 we describe the Kriging interpolation process. In Section 4.3 we define our system model and we analyze the required security properties of each piece of data in our setting. In Section 4.4 we introduce the idea of a *canonical* variogram, which is used in our construction to allow the server to compute a Kriging prediction without relying on the sensitive parameters. Our construction is given in Section 4.5 and we discuss its performance in Section 4.6. Finally, we give some final remarks and outline some potential directions for future work in Section 4.7.

4.2 Kriging Interpolation

This section outlines the background theory of Kriging interpolation. For more detail, see [18–21]. There are many variants of Kriging, but we focus on the widely used *Ordinary Kriging* variant.

The Kriging process starts with a set of *measurements* taken at some *locations* in a spatial region, and it produces *predictions* of the measurements at unsampled locations. We denote the spatial region by $R \subset \mathbb{R}^2$ and the locations of sampled measurements by $P = (r_1, r_2, \ldots, r_n)$, where each $r_i = (x_i, y_i) \in R$. The Euclidean distance between two locations $r_i, r_j \in R$ is denoted by $d(r_i, r_j)$. We refer to the set of taken measurements

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

by $S = (z_1, z_2, \ldots, z_n)$, where z_i is measured at the location $r_i \in P$. The *Kriging data set* is then the tuple (P, S).

The Kriging process allows a client to query an arbitrary location $r_0 \in R$ in order to receive a prediction z_0^* of the true value z_0 that would be measured at r_0.

4.2.1 Random Fields and Stationarity Assumptions

In order to apply the Kriging interpolation technique, the observed phenomena is viewed as a realization of a *random field* which satisfies certain properties related to the observed measurements. A random field generalizes the notion of stochastic process, by allowing the underlying parameter to take values other than real numbers. In the case of spatial interpolation, a random field Z is defined as a collection of real-valued random variables $\{Z(r)\}_{r \in R}$, all defined in the same probability space, and indexed by locations r in a fixed planar region $R \subseteq \mathbb{R}^2$.

Given a set of n samples $S = (z_1, \ldots, z_n)$ taken at positions $P = (r_1, \ldots, r_n)$, every sample $z_i \in S$ can be viewed as a realization of the random variable $Z(r_i)$, indexed by the position $r_i \in P$ in a random field Z. Given such realizations, a *linear predictor* Z^* of the random field Z is defined as a random field of the form

$$Z^*(r) = \lambda_0 + \sum_{i=1}^{n} \lambda_i Z(r_i), \qquad \text{where } \lambda_i \in \mathbb{R}.$$

We say a linear predictor Z^* is *unbiased* if the expectation $\mathbb{E}(Z(r) - Z^*(r)) = 0$ for all $r \in R$. Moreover, we say that a linear predictor Z^* is *best* or *optimal* if, for every location $r \in P$, it minimizes the prediction variance $\mathrm{Var}(Z(r) - Z^*(r))$ among all unbiased linear predictors.

The Kriging interpolation technique aims at finding a best unbiased linear predictor for the random field Z derived from a Kriging data set (P, S). In this sense, note that Kriging deals with the same problem as linear least squares does in random fields. However, in order to derive such a predictor from sampled values, additional assumptions are usually made on the *stationarity* of the random field. The most widely applied Kriging process is *Ordinary Kriging*. This form of Kriging stems from two stationarity assumptions: the *second-order stationarity assumption* and the *intrinsic stationarity assumption*.

The second-order stationarity assumption states that the first and second-order moments of the random variables in the random field are shift invariant.

Definition 4.1. A random field Z parametrized by elements of a region $R \subseteq \mathbb{R}^2$ is defined to be *second-order stationary* if the following conditions are satisfied:

- The mean $\mathrm{E}(Z(r))$ does not depend on $r \in R$, and
- The covariance $\mathrm{Cov}(Z(r), Z(r+h))$ is a function of only the separating vector h for every $r, r+h \in R$.

The intrinsic stationarity assumption considers variance of increments in place of the covariance.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Definition 4.2. A random field Z parametrized by elements of a region $R \subseteq \mathbb{R}^2$ is defined to be *intrinsic stationary* if the following conditions are satisfied:

- The mean $\mathrm{E}(Z(r))$ does not depend on $r \in R$, and
- The variance of the increments $\mathrm{Var}(Z(r+h) - Z(r))$ is a function of only the separating vector h for every $r, r+h \in R$.

Second-order stationarity implies intrinsic stationarity [21] and thus we restrict our attention to the more general intrinsic stationarity assumption. Our techniques are, however, applicable to Ordinary Kriging in general.

The intrinsic stationarity assumption naturally leads to the notion of theoretical variogram [19, 138] which models the spatial dependency between the random variables $Z(r)$. Given an intrinsic stationary random field Z, the *theoretical variogram* $\hat{\gamma} : R \to \mathbb{R}$ is defined as the function $\hat{\gamma}(h) = \mathrm{Var}(Z(r+h) - Z(r))$. Under the intrinsic assumption, $\hat{\gamma}(h)$ depends only on the norm of h [21]. Hence, we may view $\hat{\gamma}$ as a function defined over the positive real numbers.

4.2.2 The Kriging Prediction

Informally, the Kriging interpolation technique consists of three phases:

1. *Computing the experimental variogram*: One of the underlying assumptions of the Kriging process is that two measurements of a phenomenon will be similar when measured in nearby locations. Using the sampled data set, one can plot the *experimental variogram* to show the dependence between measurements sampled at locations at certain distances h.
2. *Fitting a variogram model*: Nevertheless, the experimental variogram is not usually sufficient to use directly in the Kriging prediction, since sampled data at every required distance may not be available. Therefore, one selects a parametric *variogram model* and empirically chooses model parameters to fit a curve to the points of the experimental variogram.
3. *Computing the prediction*: Using the variogram, one can determine appropriate weights for each measurement by taking into account the distance between each measurement and the queried location). The Kriging prediction is then computed as a weighted sum of the measured samples.

Let $N(h) = \{(z_i, z_j) : d(r_i, r_j) \in (h - \Delta, h + \Delta)\}$ be the set of all pairs of measurements taken approximately distance h apart[2]. The *experimental variogram* γ^* plots, for every distance h such that $N(h) \neq \emptyset$:

$$\gamma^*(h) = \frac{1}{2N(h)} \sum_{(z_i, z_j) \in N(h)} (z_i - z_j)^2.$$

[2]The approximation tolerance Δ can be increased when the Kriging data set does not include enough sample points at a close enough distance.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

A suitable variogram function $\gamma : \mathbb{R}^{\geq 0} \to \mathbb{R}$, in phase 2, must satisfy a set of conditions [18, 19]. The most commonly used models require that $\gamma(0) = 0$, that $\gamma(h)$ is positive and bounded, and the existence of the limits $\lim_{h\to 0^+} \gamma(h)$ and $\lim_{h\to\infty} \gamma(h)$. These models are parametrized by the following three variables:

- The *nugget effect* η: The limit of $\gamma(h)$ as $h \to 0^+$.
- The *sill* ν: The limit of $\gamma(h)$ as $h \to \infty$.
- The *range* ρ: Controls how fast $\gamma(h)$ approaches ν as h increases.

Typically, one chooses a variogram model from a set of standard parametric variogram models, and then fits the model to the experimental variogram by empirically adjusting the nugget effect, sill and range parameters. A selection of the most common choices of bounded variogram models are, for $h > 0$:

- The *bounded linear model*: $\gamma(h) = \nu - (\nu - \eta)\left(1 - \frac{h}{\rho}\right) 1_{(0,\rho)}(h)$.
- The *exponential variogram model*: $\gamma(h) = \nu - (\nu - \eta)e^{-h/\rho}$.
- The *spherical variogram model*: $\gamma(h) = \nu - (\nu - \eta)\left(1 - \frac{3h}{2\rho} + \frac{h^3}{2\rho^3}\right) 1_{(0,\rho)}(h)$.
- The *Gaussian variogram model*: $\gamma(h) = \nu - (\nu - \eta)e^{-h^2/\rho^2}$.

where $1_I(x) = 1$ if $x \in I$, and $1_I(x) = 0$ otherwise.

Let γ be one of the above variogram models instantiated with empirically chosen parameters. To construct the best unbiased linear predictor of the phenomenon at a queried location $r_0 = (x_0, y_0) \in R$, we first form the *Kriging matrix* $K \in \mathbb{R}^{(n+1)\times(n+1)}$ with elements

- $K_{i,j} = \gamma(d(r_i, r_j))$ for $1 \leq i, j \leq n$,
- $K_{n+1,i} = K_{i,n+1} = 1$ for $i \neq n + 1$, and
- $K_{n+1,n+1} = 0$.

Next, define a real vector $v \in \mathbb{R}^{n+1}$ with $v_i = \gamma(d(r_0, r_i))$ for $1 \leq i \leq n$, and $v_{n+1} = 1$. Let $\lambda = (\lambda_i)_{i=1}^{n+1}$ satisfy $K\lambda = v$. The *(Ordinary) Kriging prediction* z_0^* of the value z_0 of the phenomena at the unmeasured location r_0 is computed as the weighted sum of the sampled measurements, with the weights defined by λ. That is,

$$z_0^* = \sum_{i=1}^{n} \lambda_i z_i.$$

The set of linear equations defined by K and v are known as the *Normal Equations*. They are derived by imposing that the induced linear predictor is unbiased (by ensuring that the first n weights sum to one, i.e., $\sum_{i=1}^{n} \lambda_i = 1$) while minimizing the variance of the induced linear predictor [21].

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

The resulting minimized variance σ_0^{*2} is called the *(Ordinary) Kriging variance*, and it is described by the following expression

$$\sigma_0^{*2} = \lambda_{n+1} + \sum_{i=1}^{n} \lambda_i \gamma(d(r_0, r_i)).$$

The Kriging variance allows the construction of confidence intervals for each prediction, and thus it describes the error associated to the prediction. For a reference on the computation of confidence intervals in this context, see [19].

We define a variogram function to be *non-degenerate* if $\eta \neq \nu$, i.e. if γ is non-constant for $h > 0$. We restrict our attention to non-degenerate variogram functions. It is easy to see that using the degenerate variogram (also called the *nugget effect* variogram [21]) results in the average Kriging predictor $z_0^* = \sum_{i=1}^{n} z_i/n$ at all unsampled locations $r_0 \notin P$, with Kriging variance $\sigma_0^{*2} = n + 1$.

4.3 Private Outsourced Kriging Interpolation

Consider a system comprising a client $\mathcal{C}$ that owns a Kriging data set (P, S) along with a choice of variogram γ, a server $\mathcal{S}$ that is willing to perform outsourced Kriging on behalf of the client, and additional users $\mathcal{U}$ that are authorized by $\mathcal{C}$ to issue Kriging queries to $\mathcal{S}$. Furthermore, there may be additional data generating nodes (e.g. other users or remote sensors) that may update the outsourced data set by producing additional measurement data or by removing prior or outdated measurements. The requirements of each entity are as follows:

- The data owner must choose the variogram to be used and upload a Kriging data set, and it should be able to update data and request Kriging predictions.
- Data users may request Kriging predictions and update data.
- Data generating nodes should only be able to update data.
- The server should only be able to perform Kriging predictions, and should do so without learning the sensitive data used in the computation. We assume that the server $\mathcal{S}$ is honest-but-curious, i.e. it follows the Kriging protocol since its business model may depend on doing so, but it may attempt to learn information about the outsourced data.

Informally, the protocol runs as follows. The data owner $\mathcal{C}$ chooses the variogram to be used and runs the Outsource algorithm to generate the protected data set to be sent to the server, as well as the keys that are issued to authorize entities to update the outsourced data set or to perform Kriging queries respectively. Upon receipt of the protected data, the server may run the Setup algorithm to process the data and to perform any necessary precomputation. After this step, the system is ready to accept queries. The data owner or an authorized data user holding the query key may request a Kriging prediction at a specified location by running the Query algorithm to generate a query token Q. This is sent to the server, who then runs the Interpolate algorithm

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

using the processed database to generate an encrypted prediction and an encoding of the Kriging variance (which estimates the error in the prediction). An entity authorized to perform queries may learn the prediction and variance by running the Decrypt algorithm.

To dynamically update the outsourced data set, an authorized entity in possession of the update key may run the AddRequest algorithm on a specified location r' and measurement z', or the DeleteRequest algorithm on a specified location r. These algorithms produce an addition token $\alpha_{r',z'}$ or a deletion token δ_r respectively that is sent to the server. Upon receipt of such a token, the server may run the Add or Delete algorithm respectively to update the database accordingly.

For the purposes of our work, we assume that any user authorized to generate a Kriging query is also permitted to update the data set. If this should not be the case, then the proposed construction can be easily modified to include a digital signature computed on any addition or deletion token, where the signing key is contained in the update key but not in the query key. The server should then be trusted to reject any tokens that do not have a valid signature. In this way, only users in possession of the private signature key would be able to update the data set.

Definition 4.3. A private outsourced Kriging interpolation scheme comprises the following algorithms:

- $(C, UK, QK) \xleftarrow{\$} \text{Outsource}(1^\lambda, P, S, \gamma)$: A probabilistic algorithm run by $\mathcal{C}$ which takes as input a security parameter λ, the Kriging data set comprising measurement locations P and measurement values S and the chosen variogram γ. It produces an encrypted data set C that may be outsourced to the server, an update key UK that may used to update the outsourced data set, and a query key QK which may be used to form Kriging queries.
- $\text{DB} \leftarrow \text{Setup}(C)$: A deterministic algorithm run by $\mathcal{S}$ which takes as input the encrypted data set C. This algorithm enables $\mathcal{S}$ to perform any necessary processing that will enable it to compute Kriging predictions, and produces a processed outsourced data set DB.
- $Q \xleftarrow{\$} \text{Query}(r_0, QK)$: A probabilistic algorithm run by $\mathcal{C}$ or by a data user in $\mathcal{U}$ which takes as input a location $r_0 = (x_0, y_0) \in R$ for which a Kriging prediction should be computed, and the query key QK. It produces a query token Q to be sent to $\mathcal{S}$.
- $(\tilde{Z}_0, \tilde{\sigma_0}^{*2}) \leftarrow \text{Interpolate}(Q, \text{DB})$: A deterministic algorithm run by $\mathcal{S}$ that, given a query token Q and the database DB, returns an encrypted Kriging interpolation $\tilde{Z}_0$ and the partially computed Kriging variance $\tilde{\sigma_0}^{*2}$.
- $(z_0^*, \sigma_0^{*2}) \leftarrow \text{Decrypt}(\tilde{Z}_0, \tilde{\sigma_0}^{*2}, QK)$: A deterministic algorithm run by $\mathcal{C}$ or a user in $\mathcal{U}$ that takes as input the Kriging results $\tilde{Z}_0$ and $\tilde{\sigma_0}^{*2}$ from the server and the query key QK, and outputs the Kriging prediction z_0^* and the Kriging variance σ_0^{*2} at the queried location.
- $\alpha_{r',z'} \leftarrow \text{AddRequest}(r', z', UK)$: A deterministic algorithm run by $\mathcal{C}$, a data user in $\mathcal{U}$ or a data generating node. It takes a location r', a measurement value z' and the update key UK, and outputs an addition token $\alpha_{r',z'}$.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

- $\mathrm{DB}' \leftarrow \mathrm{Add}(\mathrm{DB}, \alpha_{r',z'},)$: A deterministic algorithm run by $\mathcal{S}$ which takes as input the current outsourced database DB and an addition token $\alpha_{r',z'}$, and outputs an updated database DB' representing the Kriging data set $(P \cup \{r'\}, S \cup \{z'\})$.
- $\delta_r \leftarrow \mathrm{DeleteRequest}(r, UK)$: A deterministic algorithm run by $\mathcal{C}$, a data user in $\mathcal{U}$ or a data generating node. The algorithm takes as input a location $r \in P$ and the update key UK and outputs a deletion token δ_r.
- $\mathrm{DB}' \leftarrow \mathrm{Delete}(\mathrm{DB}, \delta_r)$: A deterministic algorithm by the server which takes as input the current database DB and a deletion token δ_r and outputs an updated database DB' representing the Kriging data set $(P \setminus \{r\}, S \setminus \{z_r\})$ where $z_r \in S$ is the measurement corresponding to location $r \in P$ in DB.

We now analyze the security requirements of each component within a Kriging system. Table 4.1 summarizes the analysis.

- The measurement values $z_i \in S$ are highly sensitive and business-critical and must be protected at all times.
- In the current work, we consider the coordinates $r_i \in P$ of previous measurements to not be sensitive. This is reasonable, since in some applications they may be externally observable, for instance if they are the locations of previous mining activity.
- The queried location r_0 at which a new prediction should be computed may reveal areas of particular interest to the user. The sensitivity of this relies on the setting and on individual user requirements. However, in practice, Kriging queries are often made at every location within a region to produce a heat map of a phenomenon, which may limit the sensitivity of individual query locations. Further, the basic assumption of Kriging is that the quality of prediction degrades with distance. Thus, the best Kriging results will be obtained when the queried location lies broadly within the region of prior observed measurements.
- The computed prediction z_0^* is highly sensitive as it may form the basis of future decisions and may be business-critical, and must be protected.
- The choice of variogram model, without the variogram parameters, may reveal some information about the overall trend of the spatial dependencies of the measurements. We assume that this is not particularly sensitive information.
- The range parameter ρ of the variogram is a constant scaling of the region R denoting the inter-measurement distance h at which the spatial dependency becomes negligible. For distances $h > \rho$, the variogram approaches the variance of the measurements [21], which is represented by the sill ν.

  The nugget effect η reveals the spatial dependency at very small distances.

  In this work, we assume that the range is not sensitive as it merely scales the region R, and we also assume that information revealed by the nugget and sill may be sensitive. Even in applications where this direct information on the variance and spatial dependency of measurements is deemed non-sensitive, it may be the case that the variogram parameters are commercially sensitive. These parameters must be chosen empirically to best fit the experimental data, a process which may be time-consuming, and the quality of predictions depends on how well the variogram matches the experimental variogram.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Data	r_i	z_i	(x_0, y_0)	z_0^*	γ model	ρ	ν	η
Protection	✗	✓	✗	✓	✗	✗	✓	✓

TABLE 4.1: Data protection offered by our private outsourced Kriging scheme.

4.4 Our Techniques

In this section we introduce the main concept used in our construction, namely the canonical variogram. We then show how to factor out the variogram parameters in the Normal equations, which ultimately allows us to remove these parameters from the outsourced data set and to use them only to recover the final prediction on the client side.

The crux of our solution for the private outsourcing of Kriging interpolation is to observe how the Kriging solution varies according to the variogram nugget effect η, the sill ν, and range ρ in the non-degenerate case. We define a *canonical* variogram for each variogram model by arbitrarily fixing the parameters $\eta = \rho = 1$ and $\nu = 0$, although our results clearly translate to other choices.

Since the Kriging process is inherently linear, we show how to factor out the sensitive parameters η and ν from the variogram to leave just the canonical variogram. Using this result and an additively homomorphic scheme, an untrusted server can compute a related Kriging prediction and variance without any knowledge of η, ν and the actual measurements. The variogram parameters can then be efficiently re-added by the client locally to compute the final prediction.

Definition 4.4 (Canonical Variogram)**.** Let $\gamma(h)$ be a non-degenerate variogram function with nugget effect η, sill ν and range ρ. We define its associated *canonical variogram* as the function $\tilde{\gamma} : \mathbb{R}^{\geq 0} \to \mathbb{R}$ satisfying $\tilde{\gamma}(0) = 0$ and

$$\tilde{\gamma}(h) = -\frac{1}{\nu - \eta}\gamma(\rho h) + \frac{\nu}{\nu - \eta} \quad \text{for } h > 0. \tag{4.1}$$

Note that for any non-degenerate variogram function coming from the parametric variogram models defined in Section 4.2, the canonical variogram depends only on the considered model itself and not on any parameters. This is the case for several other parametric variogram models used in practice.

Now, given a Kriging data set (P, S) of n measurements, a query position $r_0 \notin P$ and a variogram function γ with nugget effect η, sill ν and range ρ, let $K\lambda = v$ be the corresponding Normal equations as defined in Section 4.2. Our main result in this stage is that it suffices to consider a canonical version of the Normal equations that depends only on the chosen variogram model, as well as on P and on the range parameter ρ of γ.

Definition 4.5. We define the *canonical Normal equations* as the linear system obtained from the Normal equations $K\lambda = v$ by replacing

- every $r_i \in P$ by r_i/ρ,
- the query position r_0 by r_0/ρ,

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

- the variogram $\gamma(h)$ by the canonical variogram $\tilde{\gamma}(h)$,

and we denote the canonical Normal equations by $\tilde{K}\tilde{\lambda} = \tilde{v}$.

Note that, since the canonical variogram is parameterless, the canonical Normal equations involve only the variogram model and the locations in P scaled by the inverse of the range parameter ρ. We make extensive use of this observation in our construction. Indeed, this observation allows us to take advantage of the linearity of the Kriging predictor in order to protect the measurements and the interpolation value, while the sill and the nugget parameters ν, η are hidden from the server by storing them locally.

The solution to the canonical Normal equations can be described as follows:

Proposition 4.6. *Let $K, K' \in \mathbb{R}^{(n+1)\times(n+1)}$ be real matrices, and let $v, v' \in \mathbb{R}^{n+1}$ be real vectors such that:*

- *there exist $a, b \in \mathbb{R}$ such that $K'_{i,j} = aK_{i,j} + b$ and $v'_i = av_i + b$ for all $1 \leq i, j \leq n$,*
- *$K_{i,n+1} = K_{n+1,i} = K'_{i,n+1} = K'_{n+1,i} = v_{n+1} = v'_{n+1} = 1$ for all $1 \leq i \leq n$,*
- *$K_{n+1,n+1} = K'_{n+1,n+1} = 0$.*

Then, if $\lambda \in \mathbb{R}^{n+1}$ satisfies $K\lambda = v$, the vector $\lambda' \in \mathbb{R}^{n+1}$ defined by

$$\lambda'_i = \lambda_i \text{ for all } 1 \leq i \leq n,$$
$$\lambda'_{n+1} = a\lambda_{n+1}$$

satisfies $K'\lambda' = v'$.

Proof. Note that $(K'\lambda')_i = av_i + b\sum_{i=1}^{n} \lambda_i$ for $1 \leq i \leq n$, and $(K'\lambda')_{n+1} = 1$. Since $\sum_{i=1}^{n} \lambda_i = 1$ by the last equation of the system $K\lambda = v$, the result follows. □

This result extends an observation by [19], which states that summing a constant to the variogram does not alter the solutions of the Normal equations, and that such a transformation of the variogram may sometimes be necessary in order to obtain a numerically stable Kriging prediction.

We apply this proposition to the Normal equations with $a = -1/(\nu - \eta)$ and $b = \nu/(\nu - \eta)$, and consider the canonical Normal equations. By the definitions of the Kriging prediction and the Kriging variance in Section 4.2, we directly obtain the following Corollary.

Corollary 4.7. *Let z_0^* and $\tilde{z_0}^*$ be the Kriging predictions computed from the Normal and the canonical Normal equations described above, respectively. Denote by σ_0^{*2} and $\tilde{\sigma_0}^{*2}$ the Kriging variance associated to each of the predictors. Then*

$$\tilde{z_0}^* = z_0^* \quad \text{and} \quad \tilde{\sigma_0}^{*2} = -\frac{1}{\nu - \eta}\sigma_0^{*2} + \frac{\nu}{\nu - \eta}.$$

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Therefore, in case that the employed variogram is non-degenerate, the Kriging prediction is independent of the sill ν and nugget η parameters of the variogram, while the range parameter ρ scales positions. We also see that, when applying a linear transformation to the variogram, the Kriging variance of the obtained Kriging predictor varies according to the same transformation.

4.5 Our Construction

We now outline the operation of each of the algorithms in Definition 4.3. Let $\mathcal{H} = (\mathcal{H}.\text{Gen}, \mathcal{H}.\text{Enc}, \mathcal{H}.\text{Dec})$ be an IND-CPA-secure additive HE scheme, such as the Paillier encryption scheme [108]. Then:

- $(C, UK, QK) \xleftarrow{\$} \text{Outsource}(1^\lambda, P, S, \gamma)$: If γ is a degenerate variogram function, halt and return $\perp$. In this case, our protocol fails. However, if γ is degenerate, the variogram is constant (the so-called *nugget effect model*) and models a purely random variable with no spatial correlation. Hence it is particularly easy to compute predictions in this case: the prediction is $z_0^* = \sum z_i/n$ for $r_0 \notin P$ and the variance is $\sigma_0^{*2} = n + 1$.

  Otherwise, generate a key-pair for the homomorphic encryption scheme:

  $$(pk, sk) \xleftarrow{\$} \mathcal{H}.\text{Gen}(1^\lambda).$$

  Recall that $P \subseteq \mathbb{R}^2$ is the ordered set of locations $(r_i)_{i=1}^n$ and that $S \subseteq \mathbb{R}$ is the ordered set of measurements $(z_i)_{i=1}^n$. Recall also that the variogram γ comprises three parameters: the nugget η, the sill ν and the range ρ. Let $\tilde{\gamma}$ be the canonical variogram associated to γ, as defined in Section 4.4. Define

  $$UK = (pk, \rho) \text{ and } QK = (sk, \eta, \nu, \rho).$$

  To account for the factor of ρ in the input to γ in equation 4.1, compute

  $$\tilde{P} = ((x_i/\rho, y_i/\rho))_{i=1}^n.$$

  Finally, encrypt each measurement in S and define the ordered set

  $$\text{Z} = (\mathcal{H}.\text{Enc}_{pk}(z_i))_{i=1}^n.$$

  Output $C = (\tilde{P}, \text{Z}, \tilde{\gamma})$, along with UK and QK.

- $\text{DB} \leftarrow \text{Setup}(C)$: Instantiate the matrix $\tilde{K}$ from the canonical Normal equations using positions in $r'_i \in \tilde{P}$ and the canonical variogram function $\tilde{\gamma}$:
  - $\tilde{K}_{i,j} = \tilde{\gamma}(d(r'_i, r'_j))$ for $1 \leq i, j \leq n$,
  - $\tilde{K}_{n+1,i} = \tilde{K}_{i,n+1} = 1$ for $i \neq n + 1$, and
  - $\tilde{K}_{n+1,n+1} = 0$.

  Return $\text{DB} = (\tilde{K}, C)$.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

- $Q \xleftarrow{\$} \text{Query}(r_0, QK)$: Let $r_0 = (x_0, y_0)$ and, recalling that ρ is contained within QK, return $Q = (x_0/\rho, y_0/\rho)$.

- $(\tilde{Z_0}, \tilde{\sigma_0}^{*2}) \leftarrow \text{Interpolate}(Q, \text{DB})$: Recall that $C = (\tilde{P}, \text{Z}, \tilde{\gamma})$. If $Q \in \tilde{P}$, then the exact measurement is contained in the outsourced data set and no prediction is required. Let j be the index such that $Q = r_j$, and return $(Z_j, \perp)$, where $\perp$ is a distinguished symbol denoting that the prediction is exact.

  Otherwise, compute the vector $\tilde{v}$ from the canonical Normal equations using the locations $r'_i \in \tilde{P}$, the query position Q and the canonical variogram $\tilde{\gamma}$:

  - $v_i = \tilde{\gamma}(d(Q, r'_i))$ for $1 \leq i \leq n$, and
  - $v_{n+1} = 1$.

  Compute the solution $\tilde{\lambda}$ to the canonical Normal equation $\tilde{K}\tilde{\lambda} = \tilde{v}$. This step essentially computes the Kriging coefficients λ using the canonical variogram and the scaled locations without requiring the parameters of the variogram. Then, using the homomorphic property of the encryption, compute:

$$\tilde{Z_0} = \sum_{i=1}^{n} \tilde{\lambda}_i Z_i \text{ and } \tilde{\sigma_0}^{*2} = \tilde{\lambda}_{n+1} + \sum_{i=1}^{n} \tilde{\lambda}_i \tilde{\gamma}(Q, r'_i).$$

  Return the encrypted prediction $\tilde{Z_0}$ and the partially computed Kriging variance $\tilde{\sigma_0}^{*2}$.

- $(z_0^*, \sigma_0^{*2}) \leftarrow \text{Decrypt}(\tilde{Z_0}, \tilde{\sigma_0}^{*2}, QK)$: First decrypt the Kriging prediction:

$$\tilde{z_0}^* = \mathcal{H}.\text{Dec}_{sk}(\tilde{Z_0}),$$

  where sk is contained within QK. Then, if $\tilde{\sigma_0}^{*2} = \perp$, set $\sigma_0^{*2} = 0$. Else, compute the Kriging variance

$$\sigma_0^{*2} = \nu - (\nu - \eta)\tilde{\sigma_0}^{*2}.$$

  This final step essentially adds back in the parameters of the variogram, which were removed for outsourcing, using the result from Corollary 4.7.

- $\alpha_{r',z'} \leftarrow \text{AddRequest}(r', z', UK)$: Let $r_a = \frac{r'}{\rho}$ and compute the ciphertext

$$Z_a = \mathcal{H}.\text{Enc}_{pk}(z'),$$

  where ρ and pk are contained within UK. Output the addition token

$$\alpha_{r',z'} = (r_a, Z_a).$$

- $\text{DB}' \leftarrow \text{Add}(\text{DB}, \alpha_{r',z'})$: Recall that $\alpha_{r',z'} = (r_a, Z_a)$. Compute the updated data set: if $r_a \in \tilde{P}$ then let j be the index such that $r_j = r_a$ and modify $Z_j \in \text{Z}$ to be Z_a. Otherwise, set $C' = (\tilde{P} \cup \{r_a\}, \text{Z} \cup \{Z_a\}, \tilde{\gamma})$. Return the output of $\text{Setup}(C')$.

- $\delta_r \leftarrow \text{DeleteRequest}(r, UK)$: Return $\delta_r = r/\rho$.

- $\text{DB}' \leftarrow \text{Delete}(\text{DB}, \delta_r)$: If $\delta_r \notin \tilde{P}$, return DB as there is nothing to remove. Otherwise, let j be the index such that $r = r_j$ in $\tilde{P}$. Compute the updated data set $C' = (\tilde{P} \setminus \{r_j\}, \text{Z} \setminus \{Z_j\}, \tilde{\gamma})$ and return the output of $\text{Setup}(C')$.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

4.6 Discussion

The correctness of the scheme is immediate from Corollary 4.7 as well as the correctness and homomorphic properties of the encryption scheme $\mathcal{H}$. These homomorphic properties enable addition and scalar multiplication of ciphertexts, all while ensuring that the results decrypt correctly. Corollary 4.7 shows that the Kriging prediction, as well as the Kriging variance, can be computed by applying a linear transformation to the result computed using the canonical (parameterless) variogram. Correctness of the updates is apparent because the addition and deletion tokens format the data in the same way as the original data set. Since the server is trusted to act honestly but curiously, it shall modify the data set correctly. The remainder of the update algorithms then simulate a new setup procedure running Setup on a new Kriging data set from Outsource.

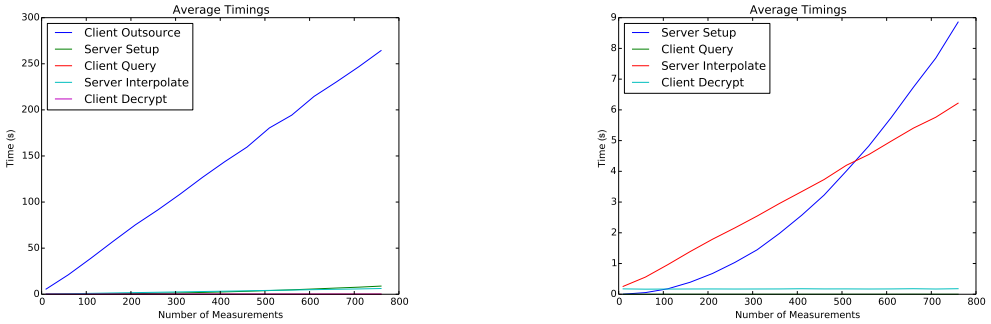
In terms of security, it is easy to see that the measurement values are always in encrypted form while outsourced, and that the leakage is bounded by the variogram model as well as both the queried and observed locations scaled by the inverse of range parameter ρ. Thus, assuming no collusion between the server and users, the data is protected from the server. Furthermore, the homomorphic and security properties of the encryption scheme permit the computation to be performed on the measurements while they are in encrypted form. At no point during the computation is the data revealed. The security of the encryption scheme requires each ciphertext to be indistinguishable from a random number, while the final prediction $\tilde{Z}_0$ computed by the server comprises a weighted sum of such pseudorandom numbers. Thus, $\tilde{Z}_0$ is a valid ciphertext and is indistinguishable from random, and hence the server cannot learn the prediction from this value.

It is also clear that neither the variogram parameters η and ν, nor any values computed from them, are ever revealed to the server. The final parameter of the variogram, the range ρ, is never explicitly given to the server. However, the server does learn the coordinates of measurements scaled by ρ. Hence, the range could be revealed if the server has existing knowledge of the measurement locations. Of the three variogram parameters, we believe that the range is the least sensitive, since it reveals how quickly the variogram approaches the sill (i.e. the distance at which the spatial correlation between measurements becomes negligible) but does not reveal anything relating to the measurement values themselves.

Even though the queried location is revealed in the plain to the server, we note that the mechanism of Tugrul and Polat [76] may easily be used to gain a weak form of secrecy: during the Query algorithm, the party carrying out the query may choose $q-1$ additional locations from the region, and scale each of them by ρ. The query token then comprises q randomly permuted scaled locations. The server must perform Interpolate for each location, and the client may discard all results except the one it is interested in. Unlike [76], we do not require an oblivious transfer protocol since the querier is authorized to learn as many queries on the data set as it wishes. However, as in [76], the server may guess the location of interest with probability $1/q$, but it can not learn the prediction at this location.

Data generating nodes cannot learn Kriging predictions as they do not have the decryption key and $\mathcal{H}$ is assumed to be IND-CPA-secure.

Regarding the performance evaluation of our scheme, we have implemented our scheme in Python 3.4.3 using the PHE library [139] to provide the Paillier encryption scheme.



(A) All algorithms (B) Excluding Outsource

FIGURE 4.1: Graphs showing the timing costs of each algorithm.

The implementation is intended as a proof of concept to evaluate the efficiency of the proposed solution. The encryption scheme has not been further optimized beyond that provided by default in the PHE library, and does not use the provided countermeasures to avoid leaking the exponent of floating point numbers. We remark that implementations of Paillier typically manage issues related to fixed-point arithmetic and overflows in a transparent manner. It is not the aim of this work to discuss such issues. All code is executed locally on a t2.micro Amazon EC2 instance with a 2.5GHz Intel Xeon processor and 1GB memory running Ubuntu 14.04.4. In practice, one would expect the server to have a better specification. All timings are averaged over 30 iterations, each on a new randomly generated data set.

Figures 4.1a and 4.1b give some simple timing results using our construction. Figure 4.1b shows the per-algorithm costs excluding the update algorithms. The cost of the Outsource algorithm dominates all others due to the cost of n encryption operations. Hence, for clarity, Figure 4.1a shows the same results with the exclusion of the Outsource algorithm. It can be seen that, with the exception of the high one-time cost of Outsource, which may be amortized over many queries, the remaining client-side processes are very efficient. The server must perform quadratic work to execute Setup, but this will be required relatively rarely, i.e. during initial setup and when the outsourced data set is updated. The online workload of the client is very low, while the online work of the server is linear in the size of the data set and greater than the workload of the client, making outsourcing worthwhile. We believe that these experiments are sufficient to demonstrate the performance and scalability of our solution. To our knowledge, the range of the number of measurements is reasonable compared to what may be used in practice. For example, the well-known Meuse data set [140], often used to illustrate the Kriging process, comprises 155 measurements.

4.7 Conclusion

In this work analyze the Kriging algorithm to identify the required interaction and the required data protection when the process is outsourced. We allow a client to outsource the storage of large data sets to a cloud service provider, and to request the computation of Kriging estimates while retaining the privacy of the sensitive measurements. We discuss a number of potential solutions and we present efficient solutions that strike a balance between efficiency and security. This exploration provides a suitable solution

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

to users that wish to outsource functionality but are worried about data privacy, and it may aid designers in privately outsourcing similar functionalities, particularly regression and interpolation computations.

Although we believe our solution to be a good first step, there is obviously much that can be improved going forwards. In particular, other techniques should be studied to enable matrix inversion on encrypted data, as this can be a bottleneck in outsourced algorithms such as Kriging.

Chapter 5

Local Bounds for the Optimal Information Ratio of Secret Sharing Schemes

In this chapter we present our contributions to secret sharing. We remind that secret sharing schemes are cryptographic schemes whose aim is to protect a secret piece of information by dividing it into shares. This is done in such a way that the secret can only be reconstructed from some combinations of shares. Given a secret sharing scheme Σ, its *access structure* Γ is the collection of combinations of shares that allow the reconstruction of the secret.

The secret sharing literature considers mainly two complexity measures: the *information ratio* and the *optimal information ratio*. The information ratio of a secret sharing scheme Σ is the ratio between the length of the largest share of the scheme and the length of the secret, and it is denoted by $\sigma(\Sigma)$. The optimal information ratio of an access structure Γ is the infimum of $\sigma(\Sigma)$ among all schemes Σ with access structure Γ, and it is denoted by $\sigma(\Gamma)$.

The main result of our work is that, for every two access structures Γ and Γ', $|\sigma(\Gamma) - \sigma(\Gamma')| \leq |\Gamma \cup \Gamma'| - |\Gamma \cap \Gamma'|$. We prove this constructively. Given any secret sharing scheme Σ for Γ, we present a method to construct a secret sharing scheme Σ' for Γ' that satisfies the inequality $\sigma(\Sigma') \leq \sigma(\Sigma) + |\Gamma \cup \Gamma'| - |\Gamma \cap \Gamma'|$. As a consequence of this result, we see that *close* access structures admit secret sharing schemes with similar information ratio. We extend this property to other families of secret sharing schemes and models of computation, such as the family of linear secret sharing schemes, span programs and Boolean formulas. In order to understand the studied property, we also analyze the limitations of the techniques for finding lower bounds on the information ratio and other complexity measures. We examine the behavior of these bounds when we add or delete subsets from an access structure.

5.1 Introduction

Secret sharing is cryptographic primitive that is used to protect a *secret value* by distributing it into *shares*. Secret sharing is used to prevent both the disclosure and the loss

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

of secrets. In the typical scenario, a user called the *dealer* holds the secret value, and it generates a set of shares. Then, it sends each share privately to a different *participant*. A subset of participants is *authorized* if their shares determine the secret value, and *forbidden* if their shares do not contain any information on the secret value. The family of authorized subsets is called the *access structure* of the scheme. If every subset of participants is either authorized or forbidden, we say that the scheme is *perfect*. In this work we just consider perfect secret sharing schemes that are *information-theoretically secure*, that is, schemes whose security does not rely on any computational assumption.

Secret sharing schemes were introduced by Shamir [50] and Blakley [51] in 1979, and are used in many cryptographic applications such as secure multiparty computation, attribute-based encryption and distributed cryptography (see [121] for more details). These applications require the use of efficient secret sharing schemes, namely schemes with short shares, efficient generation of the shares and efficient reconstruction of the secret. The *information ratio* of a secret sharing scheme Σ is the ratio of the maximum length in bits of the shares to the length of the secret value, and we denote it by $\sigma(\Sigma)$. The information ratio is widely used as a measure of the efficiency of secret sharing schemes. *Linear* secret sharing schemes are of particular interest because they have homomorphic properties, and because the shares are generated by using linear mappings, simplifying the generation of shares and the reconstruction of the secret.

Ito, Saito and Nishizeki [52] presented a method to construct a secret sharing scheme for any monotone increasing family of subsets. Viewing access structures as monotone Boolean functions, Benaloh and Leichter [53] presented a method to construct a secret sharing scheme from any monotone Boolean formula. However, for almost all access structures, the information ratios of the schemes constructed using these and other general methods [52, 53, 78] are exponential in the number of participants. In order to understand the length of shares required to realize an access structure Γ, one recurs to the optimal information ratio $\sigma(\Gamma)$ of Γ, which is defined as the infimum of the information ratios of all the secret sharing schemes for Γ.

The computation of the optimal information ratio of access structures is generally difficult, and concrete values are known only for certain families of access structures such as particular families of multipartite access structures (e.g. [79–81]), access structures with a small number of participants (e.g. [82]) or access structures with small minimal sets (e.g. [83]). A common method to obtain bounds for this parameter is to define random variables associated to the shares and to the secret, and then apply the information inequalities of the Shannon entropy of these random variables. Csirmaz [141] used a connection between the Shannon entropy and polymatroids to develop a technique for finding lower bounds. Using this technique, it is possible to find an access structure with n participants for which the optimal information ratio is $\Omega(n/\log(n))$. Currently, this is the best lower bound on the information ratio for an access structure.

Monotone span programs over a finite field $\mathbb{F}$ are equivalent to linear secret sharing schemes with secret in $\mathbb{F}$ [78, 121]. This connection is very useful to extend bounds on the complexity of monotone span programs to bounds on the information ratio of linear secret sharing schemes. Robere et al. [142] showed that there is an access structure that requires linear secret sharing schemes of information ratio $2^{\Omega(n^{1/14}\log(n))}$.

For every perfect secret sharing scheme, the information ratio must be at least 1. The schemes that attain this bound are called *ideal*, and their access structures are also called *ideal*. Brickell and Davenport [143] showed that the access structure of ideal

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

secret sharing schemes determines a *matroid*. Conversely, entropic matroids determine ideal access structures, but a little is known about the access structures associated to other families of matroids. The connection between ideal access structures and matroids is a powerful tool to characterize families of ideal access structures, e.g. [80], and it allows to transform secret sharing problems into combinatorial ones.

Beyond this connection, we lack of general criteria to determine if an access structure admits an efficient scheme. For instance, we lack of a criterion to determine if an access structure admits secret sharing scheme with information ratio at most r, for some $r > 1$. Moreover, we do not know general properties of the family of access structures admitting efficient schemes. For other models of secret sharing, recent works provide interesting results on the characterization of the structures accepting efficient schemes [144, 145], but it is not clear how to extend them to the perfect model.

The main objective of this work is to find properties of the access structures that admit efficient secret sharing schemes. The specific question we consider is whether or not access structures that are *close* admit secret sharing schemes with similar information ratios. More concretely, the objective is to bound the difference between the optimal information ratios of access structures that differ on a small number of subsets. Answers to this question may help understand the limitations of secret sharing, and the behavior of the optimal information ratio as a function from the set of all the access structures with a certain number of participants to the real numbers.

Our main result is that $|\sigma(\Gamma) - \sigma(\Gamma')| \leq |\Gamma \cup \Gamma'| - |\Gamma \cap \Gamma'|$ for every two access structures Γ and Γ'. The proof of this result is constructive. Given any secret sharing scheme Σ for Γ, we can construct a secret sharing scheme Σ' for Γ' that satisfies $\sigma(\Sigma') \leq \sigma(\Sigma) + |\Gamma \cup \Gamma'| - |\Gamma \cap \Gamma'|$. Moreover, if Σ is linear, then Σ' is linear as well. The construction relies on a combinatorial result that allows to describe Γ' through unions and intersections of Γ and of other access structures of a particular kind. Then, using an extension of the techniques of Benaloh and Leichter [53], we generate secret sharing schemes for the desired access structure.

An immediate consequence of this bound is that the access structures that are close to access structures with efficient secret sharing schemes also admit efficient schemes, and that the access structures that are close to access structures requiring large shares also require large shares. Furthermore, this bound has consequences on cryptographic schemes and protocols that use secret sharing. For instance, using the results in [49], we see that close Q_2 adversary structures admit secure multiparty computation protocols of similar complexity, in the passive adversary case. In the context of access control, for similar policies, we can build attribute-based encryption schemes of similar complexity [24].

Using the common terminology for functions between metric spaces, we can say that the optimal information ratio is a Lipschitz function with constant 1. Moreover, we prove that this constant is optimal, that is, that σ is not Lipschitz for any constant smaller than 1.

By taking advantage of the combinatorial nature of our main result, we extend this bound to other models of computation. We are able to bound the formula leafsize and the monotone span program size for monotone Boolean functions, obtaining analogous results. In order to understand this property, we also analyze the limitations of the techniques for finding lower bounds on the information ratio. We study the nature of the

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

bounds based on the Shannon inequalities [141, 146], the Razborov rank measure [147], the critical subfamilies method [148] and submodular formal complexity measures. We describe the behavior of these bounds when we add or delete subsets from an access structure.

The search for bounds on the information ratios of close access structures is motivated by a work by Beimel, Farràs and Mintz [149]. They presented a method that, given a secret sharing scheme Σ for an access structure Γ and an access structure Γ' with $\Gamma' \subseteq \Gamma$ and $\min \Gamma' \subseteq \min \Gamma$, provides a secret sharing scheme for Γ' (where $\min \Gamma$ stands for the family of minimal subsets of Γ). They showed that if Γ and Γ' are graph access structures, $\mathrm{dist}(\min \Gamma, \min \Gamma')$ is small and Σ is efficient, then the new scheme is also efficient. We also revise one of these techniques, and we provide an alternative general combinatorial formulation of a result in [149] that can also be extended to other models of computation.

In Section 5.2 we define secret sharing, and in Section 5.3 we show preliminary results about secret sharing and access structures. Section 5.4 is dedicated to our main bound on the information ratio of secret sharing schemes. In Section 5.5 we analyze the asymptotic behavior of the optimal information ratio, and Section 5.6 presents other secret sharing constructions used for bounding the optimal information ratio. Sections 5.7 and 5.8 analyze the existing techniques for finding lower bounds on the information ratio. In Section 5.9 we present results for Boolean formulas. Finally, we state some conclusions and open problems in Section 5.10.

5.2 Definition of Secret Sharing

This work is dedicated to unconditionally secure secret sharing schemes. In this section we define access structure and secret sharing scheme, and we present the complexity measures used in this work. The definition of secret sharing is taken from [121]. For an introduction to secret sharing, see for instance [121, 122].

Definition 5.1 (Access Structure)**.** Let P be a set. A collection $\Gamma \subseteq \mathcal{P}(P)$ is *monotone increasing* if $B \in \Gamma$ and $B \subseteq C \subseteq P$ implies $C \in \Gamma$. An *access structure* is a collection $\Gamma \subseteq \mathcal{P}(P)$ of non-empty subsets of P that is monotone increasing. The family of minimal subsets in Γ is denoted by $\min \Gamma$.

Definition 5.2 (Distribution Scheme)**.** Let $P = \{1, \ldots, n\}$ and let K be a finite set. A *distribution scheme* on P with *domain of secrets* K is a pair $\Sigma = (\Pi, \mu)$, where μ is a probability distribution on a finite set R, and Π is a mapping from $K \times R$ to a set of n-tuples $K_1 \times K_2 \times \cdots \times K_n$. The set R is called *the set of random strings* and K_j is called the *domain of shares* of j.

For a distribution scheme (Π, μ) and for any $A \subseteq P$, we denote by $\Pi_A(s, r)$ the entries of $\Pi(s, r)$ indexed by elements in A. If $A = \{i\}$, we set $\Pi_i(s, r) = \Pi_A(s, r)$.

Definition 5.3 (Secret Sharing)**.** Let K be a finite set of secrets with $|K| \geq 2$. A distribution scheme (Π, μ) on P with domain of secrets K is a *secret-sharing scheme* realizing an access structure Γ if the following two requirements hold for every $A = \{i_1, \ldots, i_r\} \subseteq P$:

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

- If $A \in \Gamma$, then there exists a *reconstruction function* $\mathrm{Recon}_A : K_{i_1} \times \ldots \times K_{i_r} \to K$ such that for every $k \in K$,

$$\Pr\left(\mathrm{Recon}_A(\Pi_A(k, r)) = k \right) = 1. \tag{5.1}$$

- If $A \notin \Gamma$, then for every $a, b \in K$, and for every possible vector of shares $v = (s_j)_{j \in A}$,

$$\Pr\left(\Pi_A(a, r) = v \right) = \Pr\left(\Pi_A(b, r) = v \right). \tag{5.2}$$

In a secret sharing scheme, we usually consider that there is an additional participant p_0 not in P called the *dealer*. The dealer distributes a secret $k \in K$ according to Σ by first sampling a random string $r \in R$ according to μ, computing a vector of *shares* $\Pi(k, r) = (s_1, \ldots, s_n)$, and privately communicating each share s_j to party j. The subsets of participants in P satisfying condition (5.1) are called *authorized*, and the ones satisfying condition (5.2) are called *forbidden*. In this work we just consider *perfect* secret sharing schemes, that is, schemes in which every subset of participants is either authorized or forbidden.

Definition 5.4 (Linear Secret Sharing Scheme)**.** Let $\mathbb{F}$ be a finite field. A secret sharing scheme $\Sigma = (\Pi, \mu)$ is $(\mathbb{F}, \ell)$-*linear* if $K = \mathbb{F}^\ell$, the sets R, K_1, …, K_n are vector spaces over $\mathbb{F}$, μ is the uniform distribution on R, and Π is $\mathbb{F}$-linear.

For a secret sharing scheme Σ on P, the *information ratio* of Σ is defined as

$$\sigma(\Sigma) = \frac{\max_{1 \le j \le n} \log |K_j|}{\log |K|},$$

and the *total information ratio* of Σ is

$$\sigma^{\mathrm{T}}(\Sigma) = \frac{\sum_{1 \le j \le n} \log |K_j|}{\log |K|}.$$

We say that Σ is *ideal* if $\sigma(\Sigma) = 1$. In this case, we say that its access structure is *ideal* as well.

For an access structure Γ, we define the *optimal information ratio* $\sigma(\Gamma)$ as the infimum of the information ratio of secret sharing schemes for Γ. Also, we define the *optimal total information ratio* $\sigma^{\mathrm{T}}(\Gamma)$ as the infimum of the total information ratio of the secret sharing schemes for Γ. Analogously, for every power of a prime q we define $\lambda_{q,\ell}(\Gamma)$ and $\lambda^{\mathrm{T}}_{q,\ell}(\Gamma)$ as the infimum of the information ratios and total information ratios of the $(\mathbb{F}_q, \ell)$-linear secret sharing schemes for Γ, respectively. In cases in which the finite field and the domain of secrets are not relevant, we use $\lambda(\Gamma)$ and $\lambda^{\mathrm{T}}(\Gamma)$, the infimum of the information ratios and total information ratios of the linear secret sharing schemes for Γ.

If a participant $i \in P$ does not receive any share from the dealer in a secret sharing scheme, we set $K_i = \{\perp\}$. In this case, we say that i is *not relevant* in its access structure because i is not in any subset of $\min \Gamma$.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

5.3 Preliminaries

First, we introduce some notation on access structures and we recall some of their properties. We use some definitions that are common in extremal combinatorics. See [150] for more details.

Let P be a set. We define the *distance* between $\mathcal{B}, \mathcal{B}' \subseteq \mathcal{P}(P)$ as

$$\operatorname{dist}(\mathcal{B}, \mathcal{B}') = |\mathcal{B} \cup \mathcal{B}'| - |\mathcal{B} \cap \mathcal{B}'|,$$

which is the size of the symmetric difference of the two sets. In this work, we measure the closeness between families of subsets by this distance. Observe that $\operatorname{dist}(\mathcal{B}, \mathcal{B}') = |\mathcal{B} \setminus \mathcal{B}'| + |\mathcal{B}' \setminus \mathcal{B}|$.

A family of subsets $\mathcal{B} \subseteq \mathcal{P}(P)$ is an *antichain* if $A \nsubseteq B$ for every $A, B \in \mathcal{B}$. For any $\mathcal{B} \subseteq \mathcal{P}(P)$ we define $\min \mathcal{B}$ and $\max \mathcal{B}$ as the families of minimal and maximal subsets in $\mathcal{B}$, respectively. Both $\min \mathcal{B}$ and $\max \mathcal{B}$ are antichains. We define the *complementary* of $\mathcal{B}$ as $\mathcal{B}^c = \mathcal{P}(P) \setminus \mathcal{B}$. The *degree* of $i \in P$ in $\mathcal{B}$, denoted by $\deg_i \mathcal{B}$, is defined as the number of subsets in $\mathcal{B}$ containing i. For every set $A \subseteq P$, we define the *closure* of a set A as $\operatorname{cl}(A) = \{B \subseteq P : A \subseteq B\}$. We also define the *closure* of $\mathcal{B}$ as $\operatorname{cl}(\mathcal{B}) = \bigcup_{A \in \mathcal{B}} \operatorname{cl}(A)$. The closure of any family of subsets is monotone increasing, and so it is an access structure. A family of subsets $\mathcal{B} \subseteq \mathcal{P}(P)$ is an access structure if and only if $\operatorname{cl}(\mathcal{B}) = \mathcal{B}$. If Γ is an access structure, then $\operatorname{cl}(\min \Gamma) = \Gamma$ and Γ^c is monotone decreasing.

5.3.1 Some Families of Ideal Access Structures

Now we define three parametrized families of access structures. As we show below, these access structures admit short formulas and ideal secret sharing schemes. For any nonempty set $A \subseteq P$, we define the access structures

$$F_A = \{B \subseteq P : B \nsubseteq A\}, \quad S_A = \{B \subseteq P : A \subsetneq B\}, \quad T_A = \operatorname{cl}(A).$$

The access structure T_A is the smallest access structure that contains A, and it is usually called the *trivial access structure* for A. The access structure S_A is T_A minus $\{A\}$, and $\min S_A = \{A \cup \{p\} : p \in P \setminus A\}$ is the *sunflower* of A [150]. The access structure F_A is the biggest access structure not containing A, and it has just one maximal forbidden subset, that is A. Its minimal access structure is $\min F_A = \{\{i\} : i \notin A\}$.

Now we present secret sharing schemes for the families of access structures F_A, S_A and T_A introduced above. These secret sharing schemes are ideal, and they are valid for any finite set of secrets K with $|K| \geq 2$. Moreover, if $K = \mathbb{F}^\ell$ for some finite field $\mathbb{F}$, then we show that these access structures also admit ideal (K, ℓ)-linear secret sharing schemes.

Let $K = \{a_0, \ldots, a_{m-1}\}$ be a set of size $m \geq 2$. For the constructions we present below, we assume that K is a group. In the case that K is not a group, our constructions will be defined over $\mathbb{Z}_m$ by using a bijection between K and $\mathbb{Z}_m$. Without loss of generality, let $P = \{1, \ldots, n\}$ and $A = \{1, \ldots, t\}$ for some $t \leq n$.

- F_A: Since $\min F_A = \{\{i\} : i \notin A\}$, the participants in A are not relevant, and so we just need to define the shares of the participants in $P \setminus A$. Let $K_j = \{\perp\}$ for

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

$j \in A$ and $K_j = K$ for $j \in P \setminus A$. In this case there is no need for randomness. A secret sharing scheme for F_A is defined by the mapping Π with $\Pi_j(k) = k$ for $j \in P \setminus A$.

- S_A: For $A \subsetneq P$, consider $K_j = K$ for $j = 1, \ldots, n$, and μ the uniform distribution on $R = K^t$. A secret sharing scheme for S_A is defined by the mapping Π with $\Pi_j(k, r) = r_j$ for $1 \leq j \leq t$ and $\Pi_j(k, r) = k - \sum_{i=1}^{t} r_i$ for $t + 1 \leq j \leq n$. Observe that adapting this scheme we can construct an ideal secret sharing for any access structure Γ with $\min \Gamma \subseteq \min S_A$. For $A = P$, we have $S_P = F_P$.

- T_A: Since $\min T_A = \{A\}$, we just need to define the shares of the participants in A. Consider $K_j = K$ for $j \in A$, $K_j = \{\perp\}$ for $j \in P \setminus A$, and μ the uniform distribution on $R = K^{t-1}$. A secret sharing scheme for T_A is defined by the mapping Π with $\Pi_j(k, r) = r_j$ for $1 \leq j < t$ and $\Pi_t(k, r) = k - \sum_{i=1}^{t-1} r_i$.

Given a secret sharing scheme Σ on P and $A \subseteq P$, we define $\Sigma|_A$ as the secret sharing scheme on P in which only the participants in A receive the shares from Σ. The access structure of $\Sigma|_A$ on P is $\Gamma|_A = \{B \subseteq P : B \cap A \in \Gamma\}$, and $\min(\Gamma|_A) = \min \Gamma \cap \mathcal{P}(A)$.

5.3.2 ANDs and ORs of Secret Sharing Schemes

For any access structure Γ on P, we can define the Boolean function $f : \mathcal{P}(P) \to \{0, 1\}$ satisfying $f(A) = 1$ if and only if $A \in \Gamma$. This function is monotone increasing because $f(A) \leq f(B)$ for every $A \subseteq B$.

Benaloh and Leichter [53] presented a recursive algorithm that, given a monotone Boolean formula computing the function f associated to Γ, creates a secret sharing scheme realizing Γ. The domain of secrets in this construction is $\mathbb{Z}_m$, and the scheme is obtained by translating the AND and OR logic operations into secret sharing operations [53]. Roughly speaking, the OR of two schemes Σ_1 and Σ_2 is a scheme in which the same secret is shared independently by using Σ_1 and Σ_2. In the case of the AND operation, the secret s is split into r and $s + r$, where r is a random value in $\mathbb{Z}_m$, and then r is shared by means of Σ_1 and $r + s$ is shared independently by means of Σ_2.

Here we consider an extension of the secret sharing operations defined by Benaloh and Leichter [53] that is valid for arbitrary secret sharing schemes. We define AND and OR operations between any secret sharing schemes with the same domain of secrets. Since we did not find a precise description of these extended operations in the literature, we prefer to define them rigorously. Notice that the properties of these operations are crucial for our results in σ and λ. The proof of Lemma 5.5 is similar to the one in [53], but we show it for the sake of completeness.

Let $\Sigma_1 = (\Pi^1, \mu^1)$ and $\Sigma_2 = (\Pi^2, \mu^2)$ be two secret sharing schemes on a set of participants P that have the same domain of secrets K, satisfying that μ^1 and μ^2 are independent probability distributions on some finite sets R^1 and R^2, and let $\Pi^i : K \times R^i \to K_1^i \times \ldots \times K_n^i$ for $i = 1, 2$.

We define the *OR* of Σ_1 and Σ_2 as the secret sharing scheme $\Sigma_1 \vee \Sigma_2 = (\Pi, \mu)$ where $\Pi : K \times R \to K_1 \times \ldots \times K_n$ is the mapping with $R = R^1 \times R^2$, $K_i = K_i^1 \times K_i^2$ for $i = 1, \ldots, n$, and

$$\Pi_i(k, r_1, r_2) = (\Pi_i^1(k, r_1), \Pi_i^2(k, r_2))$$

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

for $i = 1, \ldots, n$; and μ is the product of μ^1 and μ^2.

To define the *AND* of Σ_1 and Σ_2, we need to introduce an additional scheme. Let $\Sigma_3 = (\Pi^3, \mu^3)$ be the ideal secret sharing scheme on $P' = \{1, 2\}$ with access structure $\Gamma = T_{P'} = \{P'\}$ described above, with domain of secrets K, set of random strings $R^3 = K$, and uniform probability distribution μ^3 on K. The *AND* of Σ_1 and Σ_2 is the secret sharing scheme $\Sigma_1 \wedge \Sigma_2 = (\Pi, \mu)$ where $\Pi : K \times R \to K_1 \times \ldots \times K_n$ is the mapping with $R = R^1 \times R^2 \times R^3$, $K_i = K_i^1 \times K_i^2$ for $i = 1, \ldots, n$, and

$$\Pi_i(k, r_1, r_2, r_3) = (\Pi_i^1(\Pi_1^3(k, r_3), r_1), \Pi_i^2(\Pi_2^3(k, r_3), r_2))$$

for $i = 1, \ldots, n$; and μ is the product of μ^1, μ^2 and μ^3.

Lemma 5.5. *Let Σ_1 and Σ_2 be two secret sharing schemes on the same set of participants and with the same set of secrets. Let Γ_1 and Γ_2 be their access structures, respectively. Then the access structures of the schemes $\Sigma_1 \wedge \Sigma_2$ and $\Sigma_1 \vee \Sigma_2$ are $\Gamma_1 \cap \Gamma_2$ and $\Gamma_1 \cup \Gamma_2$, respectively.*

Proof. Let Recon_A^1, Recon_A^2 and Recon_A^3 be the reconstruction functions of the schemes Σ_1, Σ_2 and Σ_3, respectively. First we prove that the access structure of $\Sigma_1 \vee \Sigma_2$ is $\Gamma_1 \cup \Gamma_2$. For a subset $A \in \Gamma_1$, we define Recon_A as Recon_A^1 over the elements from Σ_1. If $A \notin \Gamma_1$ but $A \in \Gamma_2$, we define Recon_A as Recon_A^2 over the elements from Σ_2. Then subsets in $\Gamma_1 \cup \Gamma_2$ can recover the secret. If $A \notin \Gamma_1$ and $A \notin \Gamma_2$, then A is forbidden in $\Sigma_1 \vee \Sigma_2$, because for every $a, b \in K$ and for every possible vector of shares $(s_j)_{j \in A} = (s_j^1, s_j^2)_{j \in A}$,

$$\begin{aligned}
\Pr\left(\, \Pi_A(a, r_1, r_2) = (s_j)_{j \in A} \,\right) &= \Pr\left(\, \Pi_A^1(a, r_1) = (s_j^1)_{j \in A} \,\right) \cdot \Pr\left(\, \Pi_A^2(a, r_2) = (s_j^2)_{j \in A} \,\right) \\
&= \Pr\left(\, \Pi_A^1(b, r_1) = (s_j^1)_{j \in A} \,\right) \cdot \Pr\left(\, \Pi_A^2(b, r_2) = (s_j^2)_{j \in A} \,\right) \\
&= \Pr\left(\, \Pi_A(b, r) = (s_j)_{j \in A} \,\right).
\end{aligned}$$

Now we prove that the access structure of $\Sigma_1 \wedge \Sigma_2$ is $\Gamma_1 \cap \Gamma_2$. For a subset $A \in \Gamma_1 \cap \Gamma_2$, we can reconstruct the secret by applying Recon_A^3 to the outputs of Recon_A^1 and Recon_A^2, and so A is authorized. If A is neither in Γ_1 nor Γ_2, then A is forbidden in $\Sigma_1 \wedge \Sigma_2$. Now suppose that A is in Γ_1 but not in Γ_2. For every $a, b \in K$ and for every possible vector of shares $(s_j)_{j \in A} = (s_j^1, s_j^2)_{j \in A}$,

$$\begin{aligned}
&\Pr\left(\, \Pi_A(a, r_1, r_2, r_3) = (s_j^1, s_j^2)_{j \in A} \,\right) = \\
&= \Pr\left(\Pi_A^1(\Pi_1^3(a, r_3), r_1) = (s_j^1)_{j \in A} \,\right) \cdot \Pr\left(\, \Pi_A^2(\Pi_2^3(a, r_3), r_2) = (s_j^2)_{j \in A} \,\right) \\
&= \Pr\left(\Pi_A^1(r_3, r_1) = (s_j^1)_{j \in A} \,\right) \cdot \Pr\left(\, \Pi_A^2(a - r_3, r_2) = (s_j^2)_{j \in A} \,\right) \\
&= \Pr\left(\Pi_A^1(\Pi_1^3(b, r_3), r_1) = (s_j^1)_{j \in A} \,\right) \cdot \Pr\left(\, \Pi_A^2(\Pi_2^3(b, r_3), r_2) = (s_j^2)_{j \in A} \,\right) \\
&= \Pr\left(\, \Pi_A(b, r_1, r_2, r_3) = (s_j^1, s_j^2)_{j \in A} \,\right),
\end{aligned}$$

and so A is forbidden. For $A \in \Gamma_2 \setminus \Gamma_1$ the proof is analogous, and for $A \notin \Gamma_2 \cap \Gamma_1$ the proof is immediate. □

In both operations, each participant receives a share from Σ_1 and a share from Σ_2, so $\sigma(\Sigma_1 \wedge \Sigma_2) = \sigma(\Sigma_1 \vee \Sigma_2) \leq \sigma(\Sigma_1) + \sigma(\Sigma_2)$, and $\sigma^{\mathrm{T}}(\Sigma_1 \wedge \Sigma_2) = \sigma^{\mathrm{T}}(\Sigma_1 \vee \Sigma_2) = \sigma^{\mathrm{T}}(\Sigma_1) + \sigma^{\mathrm{T}}(\Sigma_2)$. Therefore, for every two access structures Γ_1 and Γ_2 we have that $\sigma(\Gamma_1 \cup \Gamma_2)$ and $\sigma(\Gamma_1 \cap \Gamma_2)$ are smaller than or equal to $\sigma(\Gamma_1) + \sigma(\Gamma_2)$. Both operations

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

preserve linearity. That is, if Σ_1 and Σ_2 are $(\mathbb{F}, \ell)$-linear secret sharing scheme for a finite field $\mathbb{F}$ and $\ell > 0$, then both $\Sigma_1 \vee \Sigma_2$ and $\Sigma_1 \wedge \Sigma_2$ are $(\mathbb{F}, \ell)$-linear.

Now we present two well-known constructions for every access structure Γ [52]. Since $\Gamma = \bigcup_{A \in \min \Gamma} T_A$ and each T_A admits an ideal secret sharing scheme on A, using the OR operation we can construct a scheme Σ for Γ with $\sigma(\Sigma) = \deg(\min \Gamma) \leq |\min \Gamma|$. Since $\Gamma = \bigcap_{A \in \max \Gamma^c} F_A$ and each F_A admits an ideal secret sharing scheme on $P \setminus A$, we can construct a secret sharing scheme Σ with $\sigma(\Sigma) \leq |\max \Gamma^c|$.

Remark 5.6. All the results in this section can be adapted to other kinds of secret sharing schemes: perfect secret sharing schemes defined by discrete random variables [121], statistical secret sharing schemes [121], or computational secret sharing schemes [151]. The AND and OR operations can also be defined in these models, but in some cases they require additional restrictions.

5.4 The Main Result

We dedicate this section to the proof and the analysis of the following theorem, which is the main result of this work.

Theorem 5.7. *Let Γ, Γ' be two access structures on a set P. Then*

$$|\sigma(\Gamma) - \sigma(\Gamma')| \leq \operatorname{dist}(\Gamma, \Gamma').$$

The approach we follow to give an upper bound for $|\sigma(\Gamma) - \sigma(\Gamma')|$ for any two access structures Γ and Γ' is the following. Given a secret sharing scheme Σ for Γ, we show a way to construct a secret sharing scheme Σ' for Γ' with $\sigma(\Sigma') \leq \sigma(\Sigma) + \operatorname{dist}(\Gamma, \Gamma')$. In order to do so, we find a description of Γ' in terms of Γ and some ideal access structures, which is presented in Lemma 5.8. Then, according to this description, we can construct Σ' by reusing Σ in a special form, according to the description of Γ'. This theorem is a direct consequence of Proposition 5.9.

The motivation for reusing Σ in the construction of Σ' is that, if Γ and Γ' are close, Σ already satisfies most of the reconstruction and privacy requirements we need for Σ'. Our construction is an elegant method to delete subsets from Γ, that is, to find a solution for the case $\Gamma' \subseteq \Gamma$. In this situation, we revoke the right of some subsets in Γ to learn the secret.

Lemma 5.8. *Let Γ, Γ' be two access structures on P. Then*

$$\Gamma' = \Big(\Gamma \cap \bigcap_{A \in I} F_A\Big) \cup \bigcup_{A \in J} T_A,$$

where $I = \max(\Gamma \setminus \Gamma')$ and $J = \min(\Gamma' \setminus \Gamma)$.

Proof. Recall that $\Gamma' = \cup_{A \in \Gamma'} T_A = \cap_{A \notin \Gamma'} F_A$. First, consider the following two cases:

1. If $\Gamma \subseteq \Gamma'$, then
$$\Gamma' = \bigcup_{A \in \Gamma} T_A \cup \bigcup_{A \in \Gamma' \setminus \Gamma} T_A = \Gamma \cup \bigcup_{A \in J} T_A.$$

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

2. If $\Gamma' \subseteq \Gamma$, then
$$\Gamma' = \bigcap_{A \notin \Gamma} F_A \cap \bigcap_{A \in \Gamma \setminus \Gamma'} F_A = \Gamma \cap \bigcap_{A \in I} F_A.$$

Suppose that Γ is not contained in Γ' and vice versa. Then consider their intersection and observe that $\Gamma \cap \Gamma' \subseteq \Gamma$. Following the arguments used above in case 2 we obtain that
$$\Gamma \cap \Gamma' = \Gamma \cap \bigcap_{A \in I'} F_A,$$
where $I' = \max(\Gamma \setminus (\Gamma \cap \Gamma')) = \max(\Gamma \setminus \Gamma') = I$. Since $\Gamma \cap \Gamma' \subseteq \Gamma'$, following the arguments used above in case 1 we obtain that
$$\Gamma' = (\Gamma \cap \Gamma') \cup \bigcup_{A \in J'} T_A,$$
where $J' = \min(\Gamma' \setminus (\Gamma \cap \Gamma')) = \min(\Gamma' \setminus \Gamma) = J$. This concludes the proof. □

Proposition 5.9. *Let Γ, Γ' be two access structures on P. Then*
$$\sigma(\Gamma') - \sigma(\Gamma) \leq |\max(\Gamma \setminus \Gamma')| + |\min(\Gamma' \setminus \Gamma)|.$$

Proof. Let Σ be a secret sharing scheme for Γ. By Lemma 5.8, the access structure Γ' is realized by the secret sharing scheme
$$\Sigma' = \Big(\Sigma \wedge \bigwedge_{A \in I} \Sigma_{F_A}\Big) \vee \bigvee_{A \in J} \Sigma_{T_A},$$
where $I = \max(\Gamma \setminus \Gamma')$, $J = \min(\Gamma' \setminus \Gamma)$, and Σ_{F_A} and Σ_{T_A} are ideal secret sharing schemes for F_A and T_A, respectively. Then $\sigma(\Sigma') \leq \sigma(\Sigma) + |I| + |J|$. □

Since $|\max(\Gamma \setminus \Gamma')| + |\min(\Gamma' \setminus \Gamma)| \leq |\Gamma \setminus \Gamma'| + |\Gamma' \setminus \Gamma| = \operatorname{dist}(\Gamma, \Gamma')$, Theorem 5.7 is a consequence of the previous proposition.

In the proof of Proposition 5.9, we construct a secret sharing scheme for Γ' in terms of ANDs and ORs of a scheme for Γ and of schemes for access structures of the form T_A and F_A. These access structures admit ideal schemes for any set of secrets. Therefore, this result is also valid if we restrict ourselves to secret sharing schemes for a particular secret size, for example to secret sharing schemes sharing one bit. In addition, these access structures also admit ideal $(\mathbb{F}, \ell)$-linear secret sharing schemes for any finite field $\mathbb{F}$, for any nonempty $A \subseteq P$ and for any $\ell > 0$. Hence, if we have an $(\mathbb{F}, \ell)$-linear secret sharing scheme realizing Γ, we obtain an $(\mathbb{F}, \ell)$-linear secret sharing scheme for Γ'.

Corollary 5.10. *Let Γ, Γ' be two access structures on P, and let $\mathbb{F}_q$ be a finite field. For every $\ell \geq 1$,*
$$|\lambda_{q,\ell}(\Gamma) - \lambda_{q,\ell}(\Gamma')| \leq \operatorname{dist}(\Gamma, \Gamma').$$

As a consequence of the previous results, the access structures that are close to access structures with efficient secret sharing schemes also admit efficient schemes, and the access structures that are close to access structures requiring large shares also require large shares.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Some applications of secret sharing schemes do not require a complete definition of the access structure. They require subsets in a family $\mathcal{A} \subseteq \mathcal{P}(P)$ to be forbidden, and subsets in a family $\mathcal{B} \subseteq \mathcal{P}(P)$ to be authorized. We say that an access structure Γ is *compatible* with $\mathcal{A}$ and $\mathcal{B}$ if $\mathcal{A} \subseteq \Gamma^c$ and $\mathcal{B} \subseteq \Gamma$. If $\mathcal{A} \cup \mathcal{B} \neq \mathcal{P}(P)$, then there is a certain degree of freedom when choosing the access structure. The number of subsets that are not required to be authorized or forbidden is $r = 2^n - (|\mathcal{A}| + |\mathcal{B}|)$. If we know $\sigma(\Gamma)$ for an access structure Γ that is compatible with $\mathcal{A}$ and $\mathcal{B}$, then we can deduce that the smallest optimal information ratio of the access structures compatible with $\mathcal{A}$ and $\mathcal{B}$ is at least $\sigma(\Gamma) - r$.

5.4.1 The Lipschitz Constant of the Optimal Information Ratio

Next, we present an example that shows that, for distance equal to one, it is not possible to improve the general bound in Theorem 5.7 and in Corollary 5.10. More concretely, we describe access structures Γ_n, Γ'_n and Γ''_n with $n \geq 3$ such that $\operatorname{dist}(\Gamma''_n, \Gamma_n) = \operatorname{dist}(\Gamma''_n, \Gamma'_n) = 1$ and $|\sigma(\Gamma''_n) - \sigma(\Gamma_n)| = |\sigma(\Gamma''_n) - \sigma(\Gamma'_n)| = 1 - 1/(n-2)$. For distance greater than one, we do not know whether the bounds in Theorem 5.7 and in Corollary 5.10 are tight.

Example 5.1. *Consider the access structures Γ_n and Γ'_n on $P = \{1, \ldots, n\}$ with $\min \Gamma_n = \{\{1, i\} : 2 \leq i \leq n\}$ and $\min \Gamma'_n = \{\{1\}, \{2, \ldots, n\}\}$. These access structures admit ideal secret sharing schemes for every set of secrets, and ideal linear secret sharing schemes for any finite field $\mathbb{F}$. Now consider the access structures Γ''_n with $\min \Gamma''_n = \{\{1, i\} : 2 \leq i \leq n\} \cup \{\{2, \ldots, n\}\}$. Observe that $\Gamma''_n = \Gamma_n \cup \{\{2, \ldots, n\}\} = \Gamma'_n \setminus \{\{1\}\}$, and so $\operatorname{dist}(\Gamma''_n, \Gamma_n) = \operatorname{dist}(\Gamma''_n, \Gamma'_n) = 1$. By Theorem 5.7 and Corollary 5.10, $\sigma(\Gamma''_n) \leq 2$ and $\lambda(\Gamma''_n) \leq 2$. It was proved in [81] that $\lambda(\Gamma''_n) = \sigma(\Gamma''_n) = 2 - 1/(n-2)$ for $n \geq 3$.*

Now we use the notion of Lipschitz continuity to describe the properties of the optimal information ratio. Let $f : X \to Y$ be a function mapping a metric space (X, d_X) to a metric space (Y, d_Y), where d_X and d_Y denote the distance functions in the domain X and in the range Y, respectively. We say that f has *Lipschitz constant* k if $d_Y(f(x), f(y)) \leq k \cdot d_X(x, y)$ for every $x, y \in X$. In this case we also say that f is k-Lipschitz.

In the context of this work, we view the information ratio σ as a function whose domain is M_n, the collection of access structures on $\{1, \ldots, n\}$, and whose range is $\mathbb{R}_{\geq 1}$. Observe that $(M_n, \operatorname{dist})$ and $\mathbb{R}_{\geq 1}$ with the Euclidean distance are metric spaces. Then, we can state the following result, which is in fact equivalent to Theorem 5.7.

Corollary 5.11. *The optimal information ratio is* 1*-Lipschitz.*

By the Example 5.1, it is not possible to give a better general Lipschitz constant for σ. The notion of Lipschitz is often used in continuous domains. However, it has also been used in discrete domains, for example in the study of differential privacy (e.g. [152]). The Lipschitz property provides valuable information about the sensitivity of the function when we vary the input. In this case, it illustrates that close access structures have similar optimal information ratio. Therefore, in M_n we have regions in which the access structures admit secret sharing schemes with low information ratio, for instance around ideal access structures. The distribution of these regions and their density in M_n is an open problem. Moreover, the characterization of the values of $\mathbb{R}_{\geq 1}$ that have a preimage by σ is also an open problem.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

5.5 Asymptotic Behavior of the Bound

Our work is focused on the local behavior of the optimal information ratio, and our results are motivated by the study of the optimal information ratio of access structures that are close. In this section we analyze the asymptotic behavior of the optimal information ratio, and the convenience of bounding the difference between the optimal information ratio of two access structures by the distance between them.

In Section 5.4.1, we presented pairs of access structures at distance one which satisfy that the difference between their optimal information ratios tends to one. We did not find an equivalent result for distance greater than one, but we can show some examples that suggest that our bounds are still useful for large distances, in general.

First, we analyze the bound in Proposition 5.9. Let $f : \mathbb{N} \to \mathbb{R}$ be a function satisfying that $\sigma(\Gamma') - \sigma(\Gamma) \leq f(r)$ for every two access structures Γ and Γ', where $r = |\max(\Gamma \setminus \Gamma')| + |\min(\Gamma' \setminus \Gamma)|$. Now we consider a well-known family of access structures defined by Csirmaz in [141], which we denote by $\mathcal{F}$. For every Γ in $\mathcal{F}$, $\sigma(\Gamma) = \Omega(n/\log n)$, where n is the number of participants, $n = N + \log N$, and N is the number of minimal authorized subsets. Observe that, since $x/\log x$ is an increasing function for $x > e$ we have that $n/\log n \geq N/\log N$ for $N \geq 3$, and so $\sigma(\Gamma) = \Omega(N/\log N)$.

If we take Γ to be the empty access structure and Γ' to be in $\mathcal{F}$, we then see that $\sigma(\Gamma') - \sigma(\Gamma) = \Omega(N/\log N)$ and that $|\max(\Gamma \setminus \Gamma')| + |\min(\Gamma' \setminus \Gamma)| = N$. Hence, we obtain the restriction that $f(r) = \Omega(r/\log r)$. Therefore, if it were possible to improve the bound in Proposition 5.9, it could be improved at most by a logarithmic factor.

Now we analyze the bound in Corollary 5.10. We consider two different families of access structures, and we analyze the bound using particular results for these families. Let $\mathbb{F}_q$ be a finite field, ℓ a positive integer, and let $g : \mathbb{N} \to \mathbb{R}$ be a function that satisfies $|\lambda_{q,\ell}(\Gamma) - \lambda_{q,\ell}(\Gamma')| \leq g(d)$ for every two access structures Γ and Γ', where $d = \mathrm{dist}(\Gamma, \Gamma')$.

Let $\mathcal{H}$ be the family of access structures on a set of n participants, n even, in which all subsets of size strictly greater than $n/2$ are authorized, and the ones of size strictly smaller than $n/2$ are forbidden. There are $2^{\binom{n}{n/2}}$ access structures in $\mathcal{H}$, including the n/2-threshold access structure. Observe that for every access structure in $\mathcal{H}$, half of the access structures in $\mathcal{H}$ are at a distance greater than or equal to $\binom{n}{n/2}/2$.

Linear secret sharing schemes can be represented by matrices (see for instance [121, 122]). In an $(\mathbb{F}_q, \ell)$-linear secret sharing scheme with information ratio at most s, the dealer is associated to ℓ rows, which can be considered to be fixed to any set of linearly independent vectors in $\mathbb{F}_q^\ell$. Each participant is associated to at most ℓs rows, and so we have at most $s\ell n + \ell$ rows. By linear algebra, since P is an authorized subset, we can always find an equivalent $(\mathbb{F}_q, \ell)$-linear secret sharing scheme in which the number of columns is smaller or equal than the number of rows minus ℓ. Hence, the number of matrices of this kind is at most $q^{s^2\ell^2n^2}$. The number of access structures Γ with $\lambda_{q,\ell}(\Gamma) \leq s$ is thus smaller than $q^{s^2\ell^2n^2}$. Now we take

$$s = \frac{2^{n/2-1}}{\ell^2 n^{5/4}\sqrt{\log q}}.$$

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Using the property that $\binom{n}{n/2} \sim \frac{2^n}{\sqrt{\pi n/2}}$, if we compare $q^{s^2\ell^2 n^2}$ with $2^{\binom{n}{n/2}}$, we see that almost all access structures Γ in $\mathcal{H}$ satisfy $\lambda_{q,\ell}(\Gamma) \geq s$. This counting argument is similar to the one in [153].

We take Γ to be the $n/2$-threshold access structure. Then there exists Γ' in $\mathcal{H}$ with $\lambda_{q,\ell}(\Gamma') \geq s$ at a distance $d = \text{dist}(\Gamma, \Gamma')$, where $\binom{n}{n/2}/2 \leq d \leq \binom{n}{n/2}$. These access structures satisfy

$$|\lambda_{q,\ell}(\Gamma) - \lambda_{q,\ell}(\Gamma')| \geq 1 + s = \Omega\left(\frac{\sqrt{d}}{\ell \log(d)\sqrt{\log q}}\right).$$

Hence, we obtain the restriction that $g(d) = \Omega(\sqrt{d}/\log d)$.

Now we consider the family of forbidden graph access structures. Given a graph $G = (V, E)$, the *forbidden graph* access structure determined by G is the access structure on V containing all the pairs in E and all subsets of size at least 3. In this case, the distance between the access structures determined by $G = (V, E)$ and $G' = (V, E')$ is $|E \cup E'| - |E \cap E'|$. As a consequence of the results in [154], for any two forbidden graph access structures Γ and Γ' and for every large enough finite field $\mathbb{F}_q$ we have $|\lambda_{q,1}(\Gamma) - \lambda_{q,1}(\Gamma')| = \tilde{O}(d^{1/4})$, where $d = \text{dist}(\Gamma, \Gamma')$. The results in [155] show that every forbidden graph access structure admits a non-linear secret sharing scheme of information ratio $n^{O(\sqrt{\log\log n/\log n})} = n^{o(1)}$. This suggests that there may exist a better bound for $|\sigma(\Gamma) - \sigma(\Gamma')|$ for forbidden graph access structures.

5.6 Other Constructions for Close Access Structures

In the previous section, we presented a way to describe an access structure Γ' in terms of another access structure Γ. This combinatorial result was used to construct a secret sharing scheme for Γ' by using a secret sharing scheme for Γ.

In this section we present a method to construct secret sharing schemes that follows the same strategy, but which uses different combinatorial results. As in the previous section, we are able to provide bounds on the optimal information ratio of access structures. These bounds are useful for access structures whose minimal sets are in a special disposition. We use combinatorial results that are different from the ones presented in the previous section. In particular, the results are based on a new combinatorial notion of $(\mathcal{B}_1, \mathcal{B}_2)$-*covering*, which will be used to construct secret sharing schemes. The interest in using $(\mathcal{B}_1, \mathcal{B}_2)$-coverings is that we can transform the problem of finding an efficient scheme into the search of small coverings, and so translate a secret sharing problem into a purely combinatorial one.

5.6.1 $(\mathcal{B}_1, \mathcal{B}_2)$-Coverings

We introduce here a notion of covering that will be used to find useful descriptions of minimal access structures that are close.

Definition 5.12. Let $\mathcal{B}_1, \mathcal{B}_2 \subseteq \mathcal{P}(P)$ be two families of subsets satisfying $\mathcal{B}_1 \cap \mathcal{B}_2 = \emptyset$. A family of subsets $\mathcal{C} \subseteq \mathcal{P}(P)$ is a $(\mathcal{B}_1, \mathcal{B}_2)$-*covering* if it satisfies the following properties:

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

1. for every $A \in \mathcal{B}_1$ and for every $B \in \mathcal{C}$, $A \not\subseteq B$, and
2. for every $A \in \mathcal{B}_2$ there exists $B \in \mathcal{C}$ such that $A \subseteq B$.

Example 5.2. *Let $\mathcal{B} \subseteq \mathcal{P}(P)$ be an antichain and let $A \in \mathcal{B}$. Then $\mathcal{C} = \{P \setminus \{i\} : i \in A\}$ is an $(\{A\}, \mathcal{B} \setminus \{A\})$ – covering.*

Next, we present in Lemma 5.13 a necessary and sufficient condition for the existence of coverings, and we present in Lemma 5.14 a technical result that is used in the proof of Theorem 5.15.

Lemma 5.13. *Let $\mathcal{B}_1, \mathcal{B}_2 \subseteq \mathcal{P}(P)$. There exists a $(\mathcal{B}_1, \mathcal{B}_2)$-covering if and only if*

$$A \not\subseteq B \text{ for every } A \in \mathcal{B}_1 \text{ and } B \in \mathcal{B}_2.$$

Proof. Let $\mathcal{C}$ be a $(\mathcal{B}_1, \mathcal{B}_2)$-covering. For every $A \in \mathcal{B}_1$ and $B \in \mathcal{B}_2$, $\mathrm{cl}(A) \cap \mathcal{C} = \emptyset$ and $\mathrm{cl}(B) \cap \mathcal{C} \neq \emptyset$, so $A \not\subseteq B$. Conversely, if $A \not\subseteq B$ for every $A \in \mathcal{B}_1$ and $B \in \mathcal{B}_2$, then $\mathcal{B}_2$ is a $(\mathcal{B}_1, \mathcal{B}_2)$-covering. □

Lemma 5.14. *Let $\mathcal{B}_1, \mathcal{B}_2 \subseteq \mathcal{P}(P)$. A $(\mathcal{B}_1, \mathcal{B}_2)$-covering is also a $(\mathcal{B}_1, \mathcal{B}_3)$-covering for every $\mathcal{B}_3 \subseteq \mathcal{B}_2$.*

Beimel, Farràs and Mintz constructed efficient secret sharing schemes for very dense graphs [149]. Some of these constructions have been recently improved in [156]. The next theorem abstracts some of the techniques used in [149, Lemma 5.2] and [149, Lemma 5.4]. The proof of the theorem uses colorings of hypergraphs. A *coloring* of $\mathcal{B} \subseteq \mathcal{P}(P)$ with c colors is a mapping $\mu : P \to \{1, \ldots, c\}$ such that for every $A \in \mathcal{B}$ there exists $u, v \in A$ with $\mu(u) \neq \mu(v)$.

Theorem 5.15. *Let $\mathcal{B}_1, \mathcal{B}_2 \subseteq \binom{P}{k}$ be two families of subsets with $\mathcal{B}_1 \cap \mathcal{B}_2 = \emptyset$ for some $k > 1$. If $\mathcal{B}_1$ has degree d, then there is a $(\mathcal{B}_1, \mathcal{B}_2)$-covering of degree at most $2^k k^k d^{k-1} \ln n$.*

Proof. Due to Lemma 5.13, if $\mathcal{B}_1 \subseteq \binom{P}{k}$, the biggest family of subsets $\mathcal{B}_2' \subseteq \binom{P}{k}$ admitting a $(\mathcal{B}_1, \mathcal{B}_2')$-covering is $\mathcal{B}_2' = \binom{P}{k} \setminus \mathcal{B}_1$. By Lemma 5.14, it is enough to restrict our proof to the case $\mathcal{B}_2 = \binom{P}{k} \setminus \mathcal{B}_1$.

In order to construct a $(\mathcal{B}_1, \mathcal{B}_2)$-covering, we use colorings of $\mathcal{B}_1$. Given a coloring μ of $\mathcal{B}_1$, we consider the family of subsets of elements in P of the same color. Observe that if all the elements in a subset $A \subseteq P$ have the same color by μ, then it implies that $B \not\subseteq A$ for every $B \in \mathcal{B}_1$.

The existence of the covering is proved by using the probabilistic method (see for instance [157]). We choose $r = 2^k k^k d^{k-1} \ln n$ random colorings $\mu_1, \ldots, \mu_r$ of $\mathcal{B}_1$ with $2kd$ colors. For every coloring μ_i, we define $\mathcal{C}_i = \{\mu_i^{-1}(c) : \text{c is a color of } \mu_i\}$, that is, $\mathcal{C}_i$ is the collection of maximal monochromatic subsets in μ_i. Now we show that $\mathcal{C} = \cup_{i=1}^{r} \mathcal{C}_i$ is a $(\mathcal{B}_1, \mathcal{B}_2)$-covering with probability at least $1 - 1/(k!)$.

Let $A = \{v_1, \ldots, v_k\} \in \mathcal{B}_2$. We fix i and compute the probability that $A \subseteq B$ for some $B \in \mathcal{C}_i$, which is equivalent to say that A is monochromatic in μ_i. Fix an arbitrary coloring of $\mathcal{B}_1 \cap \mathcal{P}(P \setminus A)$ with domain $P \setminus A$. We prove that conditioned on this coloring, the probability that A is monochromatic in an extended coloring of $\mathcal{B}_1$ is at

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

least $\frac{1}{2(2kd)^{k-1}}$. Let $B \in \mathcal{B}_1$ with $v_1 \in B$. If $B \setminus \{v_1\}$ is monochromatic, then the color of v_1 must be different from the color of $B \setminus \{v_1\}$. Thus, there are at most d colors that v_1 cannot take. By extending this argument, we see that there are at most kd colors that do not allow A to be monochromatic. Thus the probability that v_1 is colored by one of the remaining $2kd - kd$ colors is at least one half, and the probability that in this case $v_2, \ldots, v_k$ are colored in the same color as v_1 is at least $1/(2kd)^{k-1}$. Then $A \subseteq B$ for some $B \in \mathcal{C}_i$ with probability at least $1/(2(2kd)^{k-1})$.

The probability that $A \nsubseteq B$ for every $B \in \mathcal{C}$ is

$$\left(1 - \frac{1}{2(2kd)^{k-1}}\right)^r \leq e^{-\frac{r}{2(2kd)^{k-1}}} = \frac{1}{n^k}.$$

Thus, the probability that $\mathcal{C}$ is not a $(\mathcal{B}_1, \mathcal{B}_2)$-covering is less than $\binom{n}{k}/n^k \leq 1/k!$. In particular, such a covering exists. □

This result has also consequences in graph theory, which corresponds to the case $k = 2$. It implies that every graph $G = (V, E)$ with $E \subseteq \binom{V}{2}$ admits an equivalence cover of degree $16d \ln n$, where d is the degree of the complementary graph $(V, \binom{V}{2} \setminus E)$ (see [149] for more details).

As we show in the next subsection, the following proposition allows the construction of Boolean formulas and secret sharing schemes for access structures.

Proposition 5.16. *Let Γ, Γ' be two access structures with $\min\Gamma' \subseteq \min\Gamma$. If $\mathcal{C}$ is a $(\min\Gamma \setminus \min\Gamma', \min\Gamma')$-covering, then*

$$\min\Gamma' = \{A \in \min\Gamma \,:\, A \subseteq B \text{ for some } B \in \mathcal{C}\}.$$

Proof. For every subset $A \in \min\Gamma'$, there exists $B \in \mathcal{C}$ with $A \subseteq B$. For every $A \in \min\Gamma \setminus \min\Gamma'$, $A \nsubseteq B$ for every $B \in \mathcal{C}$, and so the equality holds. □

5.6.2 Secret Sharing Constructions Using Coverings

The main result of this subsection is Theorem 5.18. The quality of the bounds in this theorem depends on the degree of a covering. In Theorem 5.15, we provide a bound on the degree of coverings. In Example 5.4, we show an access structure for which this technique provides optimal secret sharing schemes.

Lemma 5.17. *Let Γ, Γ' be two access structures with $\min\Gamma' \subseteq \min\Gamma$. Let Σ be a secret sharing scheme for Γ. If there exists a $(\min\Gamma \setminus \min\Gamma', \min\Gamma')$-covering of degree d, then there exists a secret sharing scheme Σ' for Γ' with*

$$\sigma(\Sigma') \leq d\sigma(\Sigma) \quad \text{and} \quad \sigma^T(\Sigma') \leq d\sigma^T(\Sigma).$$

Proof. Let $\mathcal{C}$ be a $(\min\Gamma \setminus \min\Gamma', \min\Gamma')$-covering of degree d. We define a secret sharing scheme Σ' as the OR of all the secret sharing schemes $\Sigma|_A$ for $A \in \mathcal{C}$. By Proposition 5.16, Σ' realizes Γ'. In this scheme, each $i \in P$ receives $\deg_i(\mathcal{C})$ shares. Since $\deg_i(\mathcal{C}) \leq d$, $\sigma(\Sigma') \leq d\sigma(\Sigma)$, and $\sigma^{\mathrm{T}}(\Sigma') = \sum_{A \in \mathcal{C}} \sigma^{\mathrm{T}}(\Sigma|_A) \leq d\sigma^{\mathrm{T}}(\Sigma)$. □

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Example 5.3. *Let* Γ, Γ' *be two access structures with* $\operatorname{dist}(\min\Gamma, \min\Gamma') = 1$ *and* $\min\Gamma' \subseteq \min\Gamma$. *Observe that in this case* $\operatorname{dist}(\Gamma, \Gamma')$ *can be much bigger than 1. As we saw in Example 5.2, there exists a* $(\min\Gamma \setminus \min\Gamma', \min\Gamma')$*-covering* $\mathcal{C}$ *of degree at most* $n-1$. *Hence, given a secret sharing scheme* Σ *for* Γ *we can construct a secret sharing scheme for* Γ' *whose information ratio is less than* $(n-1)\sigma(\Sigma)$.

Theorem 5.18. *Let* Γ, Γ' *be two access structures on* P. *If there exists a* $(\min\Gamma \setminus \min\Gamma', \min\Gamma')$*-covering of degree* d, *then*

$$\sigma(\Gamma') \leq d\sigma(\Gamma) + t \quad \textit{and} \quad \sigma^T(\Gamma') \leq d\sigma^T(\Gamma) + nt,$$

where $t = \deg(\min\Gamma' \setminus \min\Gamma)$.

Proof. Let Γ'' be the access structure defined by $\min\Gamma'' = \min\Gamma' \cap \min\Gamma$. Observe that $\min\Gamma \setminus \min\Gamma' = \min\Gamma \setminus \min\Gamma''$, and that every $(\min\Gamma \setminus \min\Gamma', \min\Gamma')$-covering is also a $(\min\Gamma \setminus \min\Gamma'', \min\Gamma'')$-covering by Lemma 5.14. Given a secret sharing scheme Σ for Γ, there is a secret sharing scheme Σ'' for Γ'' with $\sigma(\Sigma'') \leq d\sigma(\Sigma)$ and $\sigma^{\mathrm{T}}(\Sigma'') \leq d\sigma^{\mathrm{T}}(\Sigma)$ by Lemma 5.17. Then, using the construction in the proof of Proposition 5.9, we can construct a secret sharing scheme Σ' with access structure Γ' as $\Sigma' = \Sigma'' \vee \Sigma'''$, where $\Sigma''' = \bigvee_{A \in I} \Sigma_{T_A}$ and $I = \min(\Gamma' \setminus \Gamma'') = \min\Gamma' \setminus \min\Gamma$. □

In Example 5.3, we studied the case of two access structures Γ and Γ' such that $\operatorname{dist}(\min\Gamma, \min\Gamma') = 1$, and the technique we described can be extended to distances greater than 1. By Theorems 5.15 and 5.18, if $\min\Gamma' \subseteq \min\Gamma$, $|A| \leq k$ for every $A \in \min\Gamma$, and if the degree of $\min\Gamma \setminus \min\Gamma'$ is d, then $\sigma(\Gamma') \leq (2^k k^k d^{k-1} \ln n)\sigma(\Gamma)$. This result was proved in [149], and it was improved for the case $k = 2$ [149, Theorem 6.1]. The covering technique has also been studied in [86], which describes other methods to find lower bounds and constructions.

In the following example, we use a technique involving coverings to construct an optimal secret sharing scheme.

Example 5.4. *Let* P *be a set of* $n = 2\ell + 1$ *participants for some* $\ell > 0$. *Consider a partition* $P = \{a\} \cup P_1 \cup P_2$, *where* $|P_1| = |P_2| = \ell$. *Let* Γ *be the 2-threshold access structure on* P *and let* Σ *be an ideal secret sharing scheme for* Γ. *Let* Γ' *be the access structure on* P *with* $\min\Gamma' = \binom{P}{2} \setminus \{\{a, b\} : b \in P_2\}$. *By [149, Theorem 7.1], it holds that* $\sigma^T(\Gamma') \geq n + \ell = 3\ell + 1$. *Now we prove that this bound is tight.*

Let $\mathcal{C} = \{C_1, C_2\}$ *be the* $(\min\Gamma \setminus \min\Gamma', \min\Gamma')$*-covering with* $C_1 = \{a\} \cup P_1$ *and* $C_2 = P_1 \cup P_2$. *Using the construction described in Lemma 5.17, we obtain that* $\Sigma' = \Sigma|_{C_1} \vee \Sigma|_{C_2}$ *is a secret sharing scheme for* Γ'. *It satisfies* $\sigma^T(\Sigma') = \sigma^T(\Sigma|_{C_1}) + \sigma^T(\Sigma|_{C_2}) = \ell + 1 + 2\ell = 3\ell + 1$. *Therefore we conclude that* $\sigma^T(\Gamma') = n + \ell$.

5.6.3 A Construction Using Sunflowers

In Proposition 5.20, we present another secret sharing construction that follows a procedure analogous to the one in Theorem 5.7, which uses a different description of the access structures.

Lemma 5.19. *Let Γ, Γ' be two access structures on P. Let $\tilde{\Gamma}$ be the access structure defined by $\min\tilde{\Gamma} = (\min\Gamma) \cap \Gamma'$. Then*

$$\Gamma' = \tilde{\Gamma} \cup \bigcup_{A \in \Gamma \setminus \Gamma'} G_A \cup \bigcup_{A \in J} T_A,$$

where $G_A = \mathrm{cl}((\min S_A) \cap \Gamma')$ and $J = \min(\Gamma' \setminus \Gamma)$.

Proof. Let $\Gamma'' = \Gamma \cap \Gamma'$. According to Lemma 5.8, we can describe Γ' as $\Gamma' = \Gamma'' \cup \bigcup_{A \in J} T_A$. We dedicate the rest of the proof to show that $\Gamma'' = \tilde{\Gamma} \cup \bigcup_{A \in \Gamma \setminus \Gamma'} G_A$. Since $\Gamma = \min\Gamma \cup \bigcup_{A \in \Gamma} \min S_A$, we have that

$$\begin{aligned}\Gamma'' &= \mathrm{cl}(\Gamma'') = \mathrm{cl}(\Gamma \cap \Gamma') = \mathrm{cl}((\min\Gamma \cup (\Gamma \setminus \min\Gamma)) \cap \Gamma') \\ &= \mathrm{cl}((\min\Gamma) \cap \Gamma') \cup \bigcup_{A \in \Gamma} G_A = \tilde{\Gamma} \cup \bigcup_{A \in \Gamma} G_A.\end{aligned}$$

Let $\mathcal{B}_1 = \Gamma \setminus \Gamma'$, $\mathcal{B}_2 = \min(\Gamma \cap \Gamma')$, and $\mathcal{B}_3 = (\Gamma \cap \Gamma') \setminus \min(\Gamma \cap \Gamma')$. Let $\mathcal{A}_i = \bigcup_{A \in \mathcal{B}_i} G_A$ for $i = 1, 2, 3$. Observe that $\mathcal{B}_1 \cup \mathcal{B}_2 \cup \mathcal{B}_3 = \Gamma$, that $\Gamma'' = \tilde{\Gamma} \cup \mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3$ and that $\mathcal{A}_3 \subseteq \mathcal{A}_2$.

We claim that $\mathcal{A}_2 \subseteq \tilde{\Gamma} \cup \mathcal{A}_1$. Let $A \in \mathcal{B}_2$. If $A \in \min\Gamma$, then $A \in \tilde{\Gamma}$ because $\mathcal{B}_2 \subseteq \Gamma'$, and so $\min S_A \subseteq \tilde{\Gamma}$. Suppose that $A \notin \min\Gamma$. Then there exists $B \in \Gamma$ satisfying $A \in \min S_B$, and in particular $A \in (\min S_B) \cap \Gamma'$. Since $A \in \min(\Gamma \cap \Gamma')$, we have that $B \in \Gamma \setminus (\Gamma \cap \Gamma') = \Gamma \setminus \Gamma' = \mathcal{B}_1$. Then $\mathrm{cl}(\min S_A) \subseteq \mathrm{cl}(A) \subseteq G_B$. Therefore $\mathcal{A}_2 \subseteq \tilde{\Gamma} \cup \mathcal{A}_1$ and so $\Gamma'' = \tilde{\Gamma} \cup \mathcal{A}_1$, which concludes the proof. □

Proposition 5.20. *Let Γ, Γ' be two access structures. Let $\tilde{\Gamma}$ be the access structure with $\min\tilde{\Gamma} = (\min\Gamma) \cap \Gamma'$. Then*

$$\sigma(\Gamma') \leq \sigma(\tilde{\Gamma}) + \mathrm{dist}(\Gamma', \Gamma).$$

Proof. Let Σ and $\tilde{\Sigma}$ be secret sharing schemes for Γ and $\tilde{\Gamma}$, respectively. We use Lemma 5.19 to construct a secret sharing scheme for Γ'. Observe that for every $A \in \Gamma$, $(\min S_A) \cap \Gamma' \subseteq \min S_A$. Hence, using the scheme described for S_A in Section 5.3 we can construct an ideal secret sharing scheme for G_A, which we call Σ''_A. Then the access structure Γ' is realized by the secret sharing scheme

$$\Sigma' = \Big(\tilde{\Sigma} \vee \bigvee_{A \in \Gamma \setminus \Gamma'} \Sigma''_A\Big) \vee \bigvee_{A \in \Gamma' \setminus \Gamma} \Sigma_{T_A},$$

where Σ_{T_A} is an ideal secret sharing scheme for T_A. It satisfies $\sigma(\Sigma') \leq \sigma(\tilde{\Sigma}) + |\Gamma \setminus \Gamma'| + |\Gamma' \setminus \Gamma| = \sigma(\tilde{\Sigma}) + \mathrm{dist}(\Gamma, \Gamma')$. □

Theorem 5.7 and Proposition 5.20 are based on similar constructions, but they cannot be compared, in general.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

5.7 Lower Bounds on the Information Ratio

In this section, and in the following one, we study techniques for finding lower bounds on the information ratio. For these bounds, we analyze the effect of adding and deleting subsets in the access structure

If we view the secret and the shares of a scheme as random variables, we can compute the entropy of the secret and of the shares. Then, we can obtain bounds on the information ratio using the Shannon information inequalities and other information inequalities (see for instance [121, 146, 158]). We study the lower bound on $\sigma(\Gamma)$ introduced by Martí-Farré and Padró [146], which is denoted by $\kappa(\Gamma)$. The bound κ exploits the connection between secret sharing schemes and polymatroids, and we present it in the following. The value of κ for an access structure can also be obtained by requiring the Shannon inequalities on the entropies of the shares and the secret (see [122, 141] for more details).

The main result in this section is Theorem 5.23, which shows a property of κ that is analogous to the one in Theorem 5.7. We dedicate Section 5.7.1 to the proof of this theorem.

Definition 5.21. A *polymatroid* is a pair $\mathcal{S} = (Q, f)$ formed by a finite set Q, the *ground set*, and a *rank function* $f\colon \mathcal{P}(Q) \to \mathbb{R}$ satisfying the following properties.

- $f(\emptyset) = 0$.
- f is *monotone increasing*: if $X \subseteq Y \subseteq Q$, then $f(X) \leq f(Y)$.
- f is *submodular*: $f(X \cup Y) + f(X \cap Y) \leq f(X) + f(Y)$ for every $X, Y \subseteq Q$.

Additionally, if $f(X) \leq |X|$ for every $X \subseteq Q$ and f is integer-valued, then we say that $\mathcal{S}$ is a *matroid*.

Definition 5.22. Let Γ be an access structure on P and let $\mathcal{S} = (Q, f)$ be a polymatroid with $Q = P \cup \{p_0\}$. Then $\mathcal{S}$ is a Γ-*polymatroid* if $f(\{p_0\}) = 1$ and if it satisfies the following properties for every $A \subseteq P$:

- If $A \in \Gamma$ then $f(A \cup \{p_0\}) = f(A)$.
- If $A \notin \Gamma$ then $f(A \cup \{p_0\}) = f(A) + 1$.

For every access structure Γ on P, we define $\kappa(\Gamma)$ as the infimum of $\max_{p \in P} f(p)$ over all Γ-polymatroids $\mathcal{S} = (Q, f)$. It satisfies $\sigma(\Gamma) \geq \kappa(\Gamma)$ [146]. Most of the known lower bounds on the optimal information ratio have been obtained by computing the exact value of κ, or by computing lower bounds on κ. The exact value of κ can be obtained by solving a linear programming problem. More details about this technique can be found in [158]. Next, we present the main result of this section.

Theorem 5.23. *Let Γ, Γ' be two access structures on P. Then*

$$|\kappa(\Gamma) - \kappa(\Gamma')| \leq \operatorname{dist}(\Gamma, \Gamma').$$

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

The proof of this theorem is constructive, and resembles that of Theorem 5.7. In this case we show that, given a Γ-polymatroid, we can construct a Γ'-polymatroid in which the rank of the singletons increases at most by $\mathrm{dist}(\Gamma, \Gamma')$. The proof requires new technical lemmas and constructions on polymatroids and we defer it to Section 5.7.1.

The access structures presented in Example 5.1 have the property that σ and κ coincide, and so we have the same asymptotic behavior for κ. That is, κ is 1-Lipschitz and it is not possible to give a better general Lipschitz constant for κ.

An access structure Γ is a *matroid port* if there exists a Γ-polymatroid $\mathcal{S}$ that is a matroid. If Γ is a matroid port, then $\kappa(\Gamma) = 1$ [143, 146]. As a consequence of Theorem 5.23, the κ value of access structures that are close to matroid ports is small. Martí-Farré and Padró [146] showed that, if an access structure Γ is not a matroid port, then $\kappa(\Gamma) \geq 3/2$. Hence, if Γ is at distance 1 of a matroid port it holds that $3/2 \leq \kappa(\Gamma) \leq 2$.

Csirmaz [141] proved that $\kappa(\Gamma) \leq n$ for every access structure Γ. Therefore, the previous theorem only provides useful bounds for access structures that are very close. However, it illustrates the nature of the optimization problems with restrictions derived from Shannon inequalities and the access structure, which may be interesting for other results of information theory.

Recently, this method has been extended to non-Shannon inequalities, for instance in [158–160]. For an access structure Γ on P and for a family of information inequalities or rank inequalities I, we can define $\kappa_I(\Gamma)$ as the infimum of $\max_{p\in P} f(p)$ over all Γ-polymatroids satisfying the restrictions of I. An interesting open problem is to study for which restrictions I Theorem 5.23 can be extended to κ_I.

5.7.1 Proof of Theorem 5.23

This subsection is dedicated to the proof of Theorem 5.23. Here we use notation introduced in [160, 161] to describe polymatroids and some technical results in [161]. For a function $F : \mathcal{P}(Q) \to \mathbb{R}$ and subsets $X, Y, Z \subseteq Q$, we denote

$$\Delta_F(Y{:}Z|X) = F(X \cup Y) + F(X \cup Z) - F(X \cup Y \cup Z) - F(X)$$

and $\Delta_F(Y{:}Z) = \Delta_F(Y{:}Z|\emptyset)$. Observe that $\Delta_F(Y{:}Z|X) = \Delta_F(Y{:}X \cup Z) - \Delta_F(Y{:}X)$. In order to simplify the notation, we write $F(x)$ instead of $F(\{x\})$ for any $x \in Q$.

Proposition 5.24 ([161])**.** *A map $f\colon \mathcal{P}(Q) \to \mathbb{R}$ is the rank function of a polymatroid with ground set Q if and only if $f(\emptyset) = 0$ and $\Delta_f(y{:}z|X) \geq 0$ for every $X \subseteq Q$ and $y, z \in Q \setminus X$.*

If $\mathcal{S} = (Q, f)$ is a Γ-polymatroid, then $\Delta_f(p_0{:}A) = 1$ if $A \in \Gamma$ and $\Delta_f(p_0{:}A) = 0$ if $A \notin \Gamma$. In this case, $f(A \cup \{p_0\}) = f(A) + 1 - \Delta_f(p_0{:}A)$ for every $A \subseteq P$. Next, we enumerate some properties of Γ-polymatroids that will be used in the proof of Proposition 5.26.

Lemma 5.25. *Let Γ be an access structure and let $\mathcal{S} = (Q, f)$ be a Γ-polymatroid. Then, for every $A \subseteq Q$ and $p, q \in Q \setminus A$ we have*

p1) $\Delta_f(p{:}p|A) = f(p \cup A) - f(A) \geq 0$.

p2) $\Delta_f(p{:}A \cup \{q\}) \geq \Delta_f(p{:}A)$

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

p3) $\Delta_f(p{:}q|A\cup\{p_0\}) - \Delta_f(p{:}q|A) =$
$= \Delta_f(p_0{:}A\cup\{p,q\}) + \Delta_f(p_0{:}A) - \Delta_f(p_0{:}A\cup\{p\}) - \Delta_f(p_0{:}A\cup\{q\}).$

Below we define the AND and OR operations on Γ-polymatroids. We show in Proposition 5.26 that these operations are well defined and that the resulting polymatroids are associated to the intersection and union of access structures, respectively.

Proposition 5.26. *Let Γ_1 and Γ_2 be two access structures on P. Let $\mathcal{S}_1 = (Q, f_1)$ be a Γ_1-polymatroid and $\mathcal{S}_2 = (Q, f_2)$ a Γ_2-polymatroid. Let $f_1 \vee f_2$ and $f_1 \wedge f_2$ be the real functions on Q satisfying that for every $A \subseteq P$*

- $(f_1 \vee f_2)(A) = f_1(A) + f_2(A) - \min\{\Delta_{f_1}(p_0{:}A), \Delta_{f_2}(p_0{:}A)\}$
- $(f_1 \vee f_2)(A\cup\{p_0\}) = f_1(A\cup\{p_0\}) + f_2(A\cup\{p_0\}) - 1$
- $(f_1 \wedge f_2)(A) = f_1(A) + f_2(A)$
- $(f_1 \wedge f_2)(A\cup\{p_0\}) = f_1(A\cup\{p_0\}) + f_2(A\cup\{p_0\}) + \max\{\Delta_{f_1}(p_0{:}A), \Delta_{f_2}(p_0{:}A)\} - 1$

Then the pair $(Q, f_1 \vee f_2)$ is a $\Gamma_1 \cup \Gamma_2$-polymatroid, and $(Q, f_1 \wedge f_2)$ is a $\Gamma_1 \cap \Gamma_2$-polymatroid. These polymatroids are denoted by $\mathcal{S}_1 \vee \mathcal{S}_2$ and $\mathcal{S}_1 \wedge \mathcal{S}_2$, respectively.

Proof. First we prove that $\mathcal{S}_1 \vee \mathcal{S}_2$ and $\mathcal{S}_1 \wedge \mathcal{S}_2$ are polymatroids using Proposition 5.24. We show that $\Delta_{f_1\vee f_2}(p{:}q|A) \geq 0$ and $\Delta_{f_1\vee f_2}(p{:}q|A\cup\{p_0\}) \geq 0$ for every $p, q \in Q$ and $A \subseteq P$ and then we show the same property for $f_1 \wedge f_2$. By p1) in Lemma 5.25, it is enough to check it for $p \neq q$, and so we can split the proof into the following 6 different cases.

Let $A \subseteq P$ and let $p, q \in P$. In order to simplify the notation, we define the set $Ap = A \cup \{p\}$ and, analogously, Ap_0, Aq and Apq.

1. $\Delta_{f_1\vee f_2}(p{:}q|A) = \Delta_{f_1}(p{:}q|A) + \Delta_{f_2}(p{:}q|A) + a - b$, where

   $$a = \min\{\Delta_{f_1}(p_0{:}Apq), \Delta_{f_2}(p_0{:}Apq)\} + \min\{\Delta_{f_1}(p_0{:}A), \Delta_{f_2}(p_0{:}A)\}, \text{ and}$$
   $$b = \min\{\Delta_{f_1}(p_0{:}Ap), \Delta_{f_2}(p_0{:}Ap)\} + \min\{\Delta_{f_1}(p_0{:}Aq), \Delta_{f_2}(p_0{:}Aq)\}.$$

   Suppose that $a < b$. If $a = 0$ then $\Delta_{f_1}(p_0{:}Apq) = 0$ or $\Delta_{f_2}(p_0{:}Apq) = 0$. By p2) of Lemma 5.25, this implies that $b = 0$. Hence, we can restrict ourselves to the case $a = 1$ and $b = 2$. In this case, there exists some $i \in \{1, 2\}$ for which $\Delta_{f_i}(p_0{:}Apq) = \Delta_{f_i}(p_0{:}Ap) = \Delta_{f_i}(p_0{:}Aq) = 1$ and $\Delta_{f_i}(p_0{:}A) = 0$. Using p3) of Lemma 5.25, we have that

   $$\begin{aligned} a - b =& \Delta_{f_i}(p_0{:}Apq) + \Delta_{f_i}(p_0{:}A) - \Delta_{f_i}(p_0{:}Ap) - \Delta_{f_i}(p_0{:}Aq) \\ =& \Delta_{f_i}(p{:}q|Ap_0) - \Delta_{f_i}(p{:}q|A) \end{aligned}$$

   Therefore $\Delta_{f_1}(p{:}q|A) + \Delta_{f_2}(p{:}q|A) + a - b \geq 0$.

2. $\Delta_{f_1\vee f_2}(p{:}p_0|A) = \Delta_{f_1\vee f_2}(p_0{:}Ap) - \Delta_{f_1\vee f_2}(p_0{:}A)$

   $$= \max\{\Delta_{f_1}(p_0{:}Ap), \Delta_{f_2}(p_0{:}Ap)\} - \max\{\Delta_{f_1}(p_0{:}A), \Delta_{f_2}(p_0{:}A)\}.$$

   This is non-negative by property p2) of Lemma 5.25.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

3. $\Delta_{f_1 \vee f_2}(p{:}q|Ap_0) = \Delta_{f_1}(p{:}q|Ap_0) + \Delta_{f_2}(p{:}q|Ap_0) \geq 0$.

4. $\Delta_{f_1 \wedge f_2}(p{:}q|A) = \Delta_{f_1}(p{:}q|A) + \Delta_{f_2}(p{:}q|A) \geq 0$.

5. $\Delta_{f_1 \wedge f_2}(p{:}p_0|A) = \Delta_{f_1 \wedge f_2}(p_0{:}Ap) - \Delta_{f_1 \wedge f_2}(p_0{:}A)$

$$= \min\{\Delta_{f_1}(p_0{:}Ap), \Delta_{f_2}(p_0{:}Ap)\} - \min\{\Delta_{f_1}(p_0{:}A), \Delta_{f_2}(p_0{:}A)\}.$$

   It is non-negative by property p2) of Lemma 5.25.

6. $\Delta_{f_1 \wedge f_2}(p{:}q|A \cup \{p_0\}) = \Delta_{f_1 \vee f_2}(p{:}q|A)$ because $f_1 \wedge f_2(Ap_0) = f_1 \vee f_2(A) + 1$. Hence, it is also non-negative.

Therefore, $\mathcal{S}_1 \vee \mathcal{S}_2$ and $\mathcal{S}_1 \wedge \mathcal{S}_2$ are polymatroids. Observe that $f_1 \vee f_2(p_0) = f_1 \wedge f_2(p_0) = 1$. Since $\Delta_{f_1 \vee f_2}(p_0{:}A) = \max\{\Delta_{f_1}(p_0{:}A), \Delta_{f_2}(p_0{:}A)\}$, we have that $\Delta_{f_1 \vee f_2}(p_0{:}A) = 1$ if and only if $A \in \Gamma_1$ or $A \in \Gamma_2$, and so $\mathcal{S}_1 \vee \mathcal{S}_2$ is a $\Gamma_1 \cup \Gamma_2$-polymatroid. Since $\Delta_{f_1 \wedge f_2}(p_0{:}A) = \min\{\Delta_{f_1}(p_0{:}A), \Delta_{f_2}(p_0{:}A)\}$, we have that $\Delta_{f_1 \wedge f_2}(p_0{:}A) = 1$ if and only if $A \in \Gamma_1$ and $A \in \Gamma_2$, and so $\mathcal{S}_1 \wedge \mathcal{S}_2$ is a $\Gamma_1 \cap \Gamma_2$-polymatroid. □

Proof of Theorem 5.23. The proof of this theorem is analogous to the proof of Theorem 5.7. Let $A \subseteq P$. We define the T_A-polymatroid $\mathcal{S}_{T_A} = (Q, h)$ as the one satisfying $h(B) = |B \cap A|$ for every $B \subseteq P$, and $\Delta_h(p_0 : B) = 1$ if and only if $A \subseteq B$. We define the F_A-polymatroid $\mathcal{S}_{F_A} = (Q, h)$ as the one satisfying $h(B) = 1$ if $|B \cap (P \setminus A)| \neq 0$ and $h(B) = 0$ otherwise, and $\Delta_h(p_0 : B) = 1$ if and only if $|B \cap (P \setminus A)| > 0$.

Let $\mathcal{S}$ be a Γ-polymatroid. By Lemma 5.8, the following construction is a Γ'-polymatroid:

$$\mathcal{S}' = \Big(\mathcal{S} \wedge \bigwedge_{A \in I} \mathcal{S}_{F_A}\Big) \vee \bigvee_{A \in J} \mathcal{S}_{T_A},$$

where $I = \max(\Gamma \setminus \Gamma')$ and $J = \min(\Gamma' \setminus \Gamma)$. Then $\kappa(\Gamma') \leq \kappa(\Gamma) + |\Gamma \setminus \Gamma'| + |\Gamma' \setminus \Gamma| = \kappa(\Gamma) + \mathrm{dist}(\Gamma, \Gamma')$. □

5.8 Bounds for Linear Secret Sharing Schemes

For any finite field $\mathbb{F}$, every $(\mathbb{F}, 1)$-linear secret sharing scheme Σ is equivalent to a monotone span program of size $\sigma^{\mathrm{T}}(\Sigma)$ (see [121] for more details). Since the bounds studied in this section are bounds on the total information ratio of $(\mathbb{F}, 1)$-linear secret sharing schemes, we have the same results for the size of monotone span programs. Next, we present a formulation of the Razborov rank measure [147] that is adapted to the context of secret sharing and access structures.

5.8.1 Razborov Rank Measure

Let Γ be an access structure on P, and let $U \subseteq \Gamma$ and $V \subseteq \Gamma^c$ be two families of subsets. For any $U_0 \subseteq U$ and $V_0 \subseteq V$, we say that the Cartesian product $U_0 \times V_0$ is a (U, V)-*rectangle*. For each $i \in P$, define the (U, V)-rectangle $R_i = (U \times V) \cap (T_{\{i\}} \times F_{\{i\}})$. Denote the set of all such rectangles by $\mathcal{R}_\Gamma(U, V) = \{R_1, \ldots, R_n\}$.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Let $\mathbb{F}$ be a field and let A be any $|U| \times |V|$ matrix over $\mathbb{F}$ with rows indexed by elements of U and columns indexed by elements of V. The *restriction* of A to the rectangle $R = U_0 \times V_0$ is the submatrix $A \restriction_R$ obtained by setting to 0 all entries not indexed by elements of R.

Definition 5.27 ([147])**.** Let $\Gamma \subseteq \mathcal{P}(P)$ be an access structure, $U \subseteq \Gamma$, $V \subseteq \Gamma^c$. Let $\mathbb{F}$ be a field and let A be a $|U| \times |V|$ matrix over $\mathbb{F}$. If $\operatorname{rank}(A) > 0$, the *rank measure of* Γ *with respect to* A is given by

$$\mu_A(\Gamma) = \frac{\operatorname{rank}(A)}{\max_{R \in \mathcal{R}_\Gamma(U,V)} \operatorname{rank}(A \restriction_R)}.$$

If $\operatorname{rank}(A) = 0$, we set $\mu_A(\Gamma) = 0$. We accordingly define the *rank measure of* Γ as

$$\mu(\Gamma) = \max_A \mu_A(\Gamma),$$

where the maximum is taken over all families of subsets $U \subseteq \Gamma$, $V \subseteq \Gamma^c$ and all matrices A of the form stated above.

Razborov [147] showed that the rank measure of a monotone Boolean function is a lower bound on the size of the shortest formula for this function (see Section 5.9). Later, Gál [162] proved that the rank measure is also a lower bound on the size of monotone span programs. Taking into account the connection between monotone span programs and linear secret sharing schemes mentioned above, we obtain that the rank measure is a lower bound on the optimal information ratio for linear secret sharing schemes. More concretely, we have the following result.

Theorem 5.28. *Let $\Gamma \subseteq \mathcal{P}(P)$ an access structure, $U \subseteq \Gamma$, $V \subseteq \Gamma^c$. Let $\mathbb{F}_q$ be a field and let A be a $|U| \times |V|$ matrix over $\mathbb{F}_q$. Then,*

$$\mu_A(\Gamma) \leq \lambda^T_{q,1}(\Gamma).$$

In the following theorem, we study the behavior of the rank measure when we add or delete subsets from an access structure.

Proposition 5.29. *Let $\Gamma, \Gamma' \subseteq \mathcal{P}(P)$ be access structures, $U \subseteq \Gamma$, $V \subseteq \Gamma^c$. Let $\mathbb{F}$ be a field and let A be a $|U| \times |V|$ matrix over $\mathbb{F}$. Then, there exist $U' \subseteq \Gamma'$, $V' \subseteq \Gamma'^c$ and a $|U'| \times |V'|$ matrix A' such that*

$$\mu_A(\Gamma) \leq \mu_{A'}(\Gamma') + \operatorname{dist}(\Gamma, \Gamma').$$

Proof. Set $U' = U \cap \Gamma'$ and $V' = V \cap \Gamma'^c$, and let A' be the restriction of A to $U' \times V'$. Then, observe that $|U \backslash U'| \leq |\Gamma \backslash \Gamma'|$, since $U \backslash U' = U \backslash \Gamma'$ and $U \subseteq \Gamma$. Similarly, we see that $|V \backslash V'| \leq |\Gamma' \backslash \Gamma|$ by using $\Gamma^c \backslash \Gamma'^c = \Gamma' \backslash \Gamma$. Since A' is the submatrix obtained by setting to 0 all rows of A indexed by $U \backslash U'$ and all columns indexed by $V \backslash V'$, we have

$$\operatorname{rank}(A) \leq \operatorname{rank}(A') + |U \backslash U'| + |V \backslash V'| \leq \operatorname{rank}(A') + \operatorname{dist}(\Gamma, \Gamma').$$

Let $\mathcal{R}_\Gamma(U, V) = \{R_1, \ldots, R_n\}$ and $\mathcal{R}_{\Gamma'}(U', V') = \{R'_1, \ldots, R'_n\}$. Since $R'_i = R_i \cap (U' \times V')$, we have that $A' \restriction_{R'_i}$ is a submatrix of $A \restriction_{R_i}$, and thus $\operatorname{rank}(A \restriction_{R_i}) \geq \operatorname{rank}(A' \restriction_{R'_i})$. Hence,

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

$$\max_{R\in\mathcal{R}_\Gamma(U,V)} \operatorname{rank}(A\restriction_R) \geq \max_{R'\in\mathcal{R}_{\Gamma'}(U',V')} \operatorname{rank}(A'\restriction_{R'}).$$

Given a rectangle $R \in \mathcal{R}_\Gamma(U,V)$, let $R' = R \cap (U' \times V')$. Note that $A'\restriction_{R'}$ is a submatrix of $A\restriction_R$, and so $\operatorname{rank}(A\restriction_R) \geq \operatorname{rank}(A'\restriction_{R'})$. Since the map $\mathcal{R}_\Gamma(U,V) \to \mathcal{R}_{\Gamma'}(U',V')$ given by $R \mapsto R \cap (U' \times V')$ is exhaustive, we get the inequality

$$\max_{R\in\mathcal{R}_\Gamma(U,V)} \operatorname{rank}(A\restriction_R) \geq \max_{R'\in\mathcal{R}_{\Gamma'}(U',V')} \operatorname{rank}(A'\restriction_{R'}).$$

By using the previous inequalities, we see that

$$\begin{aligned}\mu_A(\Gamma) &= \frac{\operatorname{rank}(A)}{\max_{R\in\mathcal{R}_\Gamma(U,V)} \operatorname{rank}(A\restriction_R)} \leq \frac{\operatorname{rank}(A') + \operatorname{dist}(\Gamma,\Gamma')}{\max_{R'\in\mathcal{R}'_{\Gamma'}(U',V')} \operatorname{rank}(A'\restriction_{R'})} \\ &\leq \mu_{A'}(\Gamma') + \operatorname{dist}(\Gamma,\Gamma').\end{aligned}$$

□

Theorem 5.30. *Let $\Gamma, \Gamma' \subseteq \mathcal{P}(P)$ be access structures. Then*

$$|\mu(\Gamma) - \mu(\Gamma')| \leq \operatorname{dist}(\Gamma,\Gamma').$$

Proof. Let A be the $|U|\times|V|$ matrix such that $\mu(\Gamma) = \mu_A(\Gamma)$, and let A' be the restriction of A to $U' \times V'$, where $U' = U \cap \Gamma'$ and $V' = V \cap \Gamma'^c$. By Proposition 5.29 we have $\mu(\Gamma) \leq \mu_{A'}(\Gamma') + \operatorname{dist}(\Gamma,\Gamma')$. Now, by definition $\mu_{A'}(\Gamma') \leq \mu(\Gamma')$, so $\mu(\Gamma) \leq \mu(\Gamma') + \operatorname{dist}(\Gamma,\Gamma')$. □

Note that the behavior of the rank measure bound is different from that of $\lambda^{\mathrm{T}}_{q,1}$. If we extend the bound on Corollary 5.10 to λ^{T} we have that $|\lambda^{\mathrm{T}}_{q,\ell}(\Gamma) - \lambda^{\mathrm{T}}_{q,\ell}(\Gamma')| \leq n \cdot \operatorname{dist}(\Gamma,\Gamma')$ for every two access structures Γ and Γ'.

Recently, in [163], the rank measure bound has been used to prove that for every prime p there exist access structures Γ^p for which $\lambda^{\mathrm{T}}_{q,1}(\Gamma^p) = 2^{\Omega(n)}$ for every finite field $\mathbb{F}_q$ of characteristic different from p. Let $P = P_2 \cup P_3$, where $P_2 = \{1,\ldots,n\}$ and $P_3 = \{n+1,\ldots,2n\}$. Let Γ be the access structure P with $\Gamma|_{P_2} = \Gamma^2$ and $\Gamma|_{P_3} = \Gamma^3$ satisfying that for every $A \in \min\Gamma$ either $A \subseteq P_2$ or $A \subseteq P_3$. This access structure satisfies $\lambda^{\mathrm{T}}_{q,1}(\Gamma) = 2^{\Omega(n)}$ for every finite field $\mathbb{F}_q$.

5.8.2 Critical Subfamilies

The next technique provides lower bounds on the size of the shares for linear secret sharing schemes. It was first introduced in [148].

Definition 5.31. Let Γ be an access structure and let $\mathcal{H} \subseteq \min\Gamma$. We say that $\mathcal{H}$ is a *critical subfamily* for Γ, if every $H \in \mathcal{H}$ contains a set $T_H \subseteq H$, $|T_H| \geq 2$, such that the following two conditions are satisfied

- The set T_H uniquely determines H in the subfamily $\mathcal{H}$: No other set in $\mathcal{H}$ contains T_H.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

- For any subset $Y \subseteq T_H$, the set $S_Y = \cup_{A\in\mathcal{H},\, A\cap Y\neq\emptyset} A \setminus Y$ does not contain any member of $\min\Gamma$.

Theorem 5.32 ([148])**.** *Let $\mathbb{F}_q$ be a finite field. Let Γ be an access structure and let $\mathcal{H}$ be a critical subfamily for Γ. Then $\lambda^T_{q,1}(\Gamma) \geq |\mathcal{H}|$.*

Given a critical subfamily for an access structure Γ, it is easy to construct a critical subfamily for an access structure Γ' obtained by deleting some subsets from Γ or from $\min\Gamma$. However, it is not easy to find a critical subfamily for access structures that are obtained by adding subsets to Γ or to $\min\Gamma$.

Lemma 5.33. *Let $\mathcal{H}$ be the critical subfamily for an access structure Γ. Let Γ' be an access structure with $\min\Gamma' \subseteq \min\Gamma$ and $|\min\Gamma \setminus \min\Gamma'| = \ell$, and let Γ'' be an access structure with $\Gamma'' \subseteq \Gamma$ and $|\Gamma \setminus \Gamma''| = \ell$. Then there exist two critical subfamilies $\mathcal{H}'$ and $\mathcal{H}''$ for Γ' and Γ'', respectively, with $|\mathcal{H}'| \geq |\mathcal{H}| - \ell$ and $|\mathcal{H}''| \geq |\mathcal{H}| - \ell$.*

Proof. The families of subsets $\mathcal{H}' = \mathcal{H} \cap \min\Gamma'$ and $\mathcal{H}'' = \mathcal{H} \cap \Gamma''$ are critical subfamilies for Γ' and Γ'', respectively. □

5.9 Formulas for Monotone Boolean Functions

In this section, we apply the approach of Section 5.4 to study the behavior of the complexity measures associated to monotone Boolean functions. Informally, our results show that similar monotone Boolean functions have close complexity measures. In the first subsection, we aim at giving similar bounds as those in Theorems 5.7 and 5.18 and to Proposition 5.20 for the leafsize of monotone Boolean functions. In the second subsection, we study the family of submodular complexity measures. For an introduction to Boolean functions, see for instance [164, 165].

5.9.1 Definitions

A *Boolean function* is a function of the form $f : \{0,1\}^n \to \{0,1\}$ for some $n \geq 1$. We also see the domain of a Boolean function as the power set of $P = \{1,\dots,n\}$ via the bijection $\{0,1\}^n \to \mathcal{P}(P)$ given by $(x_i)_{i\in P} \mapsto \{i \in P : x_i = 1\}$. Then we denote by Γ_f the collection of elements $A \in \mathcal{P}(P)$ such that $f(A) = 1$. A Boolean function f is called *monotone* if Γ_f is an access structure. In this case, we denote $\min f = \min\Gamma_f$. If the domain of a Boolean function f is $\{0,1\}^n$, we say f is *fanin-n*. For two monotone fanin-n Boolean functions f, f', we define the *distance* between f and f' as $\mathrm{dist}(f,f') = \mathrm{dist}(\Gamma_f, \Gamma_{f'})$.

Given a Boolean function $f : \mathcal{P}(P) \to \{0,1\}$ and a set $B \subseteq P$, we define the *restriction of f to B* as the Boolean function $f|_B : \mathcal{P}(P) \to \{0,1\}$ given by $f|_B(A) = f(A \cap B)$. We have that $\Gamma_{f|_B} = \Gamma_f|_B$.

If $\Phi, g_1, \dots, g_m$ are Boolean functions and Φ is fanin-m, we can define a Boolean function $\Phi(g_1,\dots,g_m)$ by applying all the outputs of $g_1,\dots,g_m$ to Φ in an orderly manner. For $i \in P$, we denote the *i-th input variable* by x_i. Note that x_i can be seen as the monotone Boolean function satisfying $\Gamma_{x_i} = T_{\{i\}}$.

We now define formulas and some related concepts.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Definition 5.34. Let Ω be a set of Boolean functions. A *formula S over Ω* is a sequence $(g_1, \ldots, g_m)$ of Boolean functions such that

- the first k Boolean functions $g_1, \ldots, g_k$ are input variables,
- for every g_j that is not an input variable, there exists $\Phi \in \Omega$ and $\ell_1, \ldots, \ell_{d_j} < j$ such that $g_j = \Phi(g_{\ell_1}, \ldots, g_{\ell_{d_j}})$, and
- for every g_j other than g_m, there exists a single function in S that is computed using g_j (i.e., g_j is *fanout*-1).

We say a formula $S = (g_1, \ldots, g_m)$ *computes* a Boolean function f if $f = g_m$. We say that a formula over Ω is *monotone* if $\Omega = \{\wedge, \vee\}$. Similarly, we say it is *deMorgan* if $\Omega = \{\wedge, \vee, \neg\}$ and the gate $\neg$ can only be applied to input variables.

Let F_f and F_g be formulas computing monotone Boolean functions f and g, respectively. Then, $F_f \wedge F_g$ denotes the formula computing the Boolean function $h = f \wedge g = \max\{f, g\}$ built by appending the AND of the outputs of F_f and F_g. We then have $\Gamma_h = \Gamma_f \cap \Gamma_g$. Similarly, $F_f \vee F_g$ denotes the formula computing the Boolean function $h' = f \vee g = \min\{f, g\}$ built by appending the OR of the outputs of F_f and F_g, and we have $\Gamma_{h'} = \Gamma_f \cup \Gamma_g$. For every formula F and $B \subseteq P$, we define $F|_B$ as the formula that is obtained by replacing x_i by 0 for every $i \notin B$. If F computes a function f, then $F|_B$ computes $f|_B$.

5.9.2 Bounds on the Size of Formulas

We now analyze the minimal leafsize L, which is a complexity measure attached to monotone Boolean functions. The *leafsize* of a formula is defined as the number of input variables in it. We define the *deMorgan* (resp. *monotone*) *minimal leafsize* $L(f)$ (resp. $L_+(f)$) of a Boolean function f as the smallest leafsize over all deMorgan (resp. monotone) formulas computing f. We state our results here in terms of L, but they all hold verbatim for L_+. Moreover, our results can be adapted to other complexity measures, such as the size of Boolean formulas and of circuits.

Before stating our results, we give formulas and complexity measures for particular families of Boolean functions. We start with the Boolean functions associated to the access structures T_A, F_A, S_A defined in Section 5.3, and we proceed with the restriction $f|_B$ of a Boolean function f to $B \subseteq P$.

The functions f_{T_A} and f_{F_A} admit the formulas $\bigwedge_{i \in A} x_i$ and $\bigvee_{i \in P \setminus A} x_i$, which have leafsizes $|A|$ and $n - |A|$, respectively. Since $S_A = T_A \cap F_A$, we have that $\left(\bigwedge_{i \in A} x_i\right) \wedge \left(\bigvee_{i \in P \setminus A} x_i\right)$ is a formula for f_{S_A} with leafsize n.

We now consider the restriction $f|_B : \{0,1\}^n \to \{0,1\}$ of a Boolean function f. By applying the restriction $x_i = 0$ for all $i \notin B$ to a minimal monotone or deMorgan formula for f, and by removing redundant input variables and Boolean functions, we get a formula for $f|_B$. Therefore $L(f|_B) \leq L(f)$.

Next, we present analogous results to Theorems 5.7 and 5.18 and Proposition 5.20 for the minimal leafsize of monotone Boolean functions. The following proposition shows that close monotone Boolean functions have similar minimal leafsizes.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Proposition 5.35. *For every two monotone Boolean functions f and f',*

$$|L(f) - L(f')| \leq n \cdot \mathrm{dist}(f, f').$$

Proof. Let F be a formula computing f. Let $I = \max(\Gamma \setminus \Gamma')$ and let $J = \min(\Gamma' \setminus \Gamma)$. Using Lemma 5.8 with $\Gamma = \Gamma_f$ and $\Gamma' = \Gamma_{f'}$ we see that

$$F' = (F \wedge \bigwedge_{A \in I} G_A) \vee \bigvee_{A \in J} H_A$$

is a formula computing f', where G_A and H_A are the formulas for F_A and T_A described above, respectively. Hence,

$$L(f') \leq L(f) + \sum_{A \in I} |P \backslash A| + \sum_{A \in J} |A| \leq L(f) + n \cdot \mathrm{dist}(\Gamma, \Gamma').$$

□

The proofs of the next results follow a similar strategy than the proofs of Theorem 5.18 and Proposition 5.20.

Proposition 5.36. *Let $f, f' : \{0,1\}^n \to \{0,1\}$ be two monotone Boolean functions. If there exists a $(\min f \setminus \min f', \min f' \cap \min f')$-covering of degree d, then*

$$L(f') \leq dL(f) + nt,$$

where $t = \deg(\min f' \setminus \min f)$.

Proof. Let $\mathcal{C}$ be a $(\min f \setminus \min f', \min f \cap \min f')$-covering, and take $A \in \min f$. In this case, $A \in \min f'$ if and only if there exists $B \in \mathcal{C}$ such that $A \subseteq B$. Hence $\min f \cap \min f' = \bigcup_{B \in \mathcal{C}} (\min f \cap \mathcal{P}(B))$. Now, since $\min f' = (\min f \cap \min f') \cup (\min f' \setminus \min f)$,

$$\begin{aligned}
\Gamma_{f'} &= \mathrm{cl}(\min f') \\
&= \mathrm{cl}(\min f \cap \min f') \cup \mathrm{cl}(\min f' \setminus \min f) \\
&= \left(\bigcup_{B \in \mathcal{C}} \mathrm{cl}(\min f \cap \mathcal{P}(B)) \right) \cup \bigcup_{A \in \min f' \setminus \min f} T_A \\
&= \left(\bigcup_{B \in \mathcal{C}} \Gamma_{f|_B} \right) \cup \bigcup_{A \in \min f' \setminus \min f} T_A.
\end{aligned}$$

Hence, if H_A is the formula for T_A described above, the formula

$$F' = \left(\bigvee_{B \in \mathcal{C}} F|_B \right) \vee \bigvee_{A \in \min f' \setminus \min f} H_A$$

computes f'. □

Proposition 5.37. *Let f, f' be two monotone Boolean functions, and let $\tilde{f}$ be the monotone Boolean function with $\min \tilde{f} = \min f \cap \Gamma_{f'}$. Then*

$$L(f') \leq L(\tilde{f}) + n \cdot \mathrm{dist}(f, f').$$

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Proof. Using Lemma 5.19 with $\Gamma = \Gamma_f$, $\Gamma' = \Gamma_{f'}$ and $\tilde{\Gamma} = \Gamma_{\tilde{f}}$ we have

$$\Gamma' = \left(\tilde{\Gamma} \cup \bigcup_{A \in \Gamma' \setminus \Gamma} \mathrm{cl}\left(\min S_A \cap \Gamma'\right) \right) \cup \bigcup_{A \in \min(\Gamma \setminus \Gamma')} T_A.$$

Now note that $\mathrm{cl}\left(\min S_A \cap \Gamma'\right) = T_A \cap \bigcup_{i \notin A : A \cup \{i\} \in \Gamma'} T_{\{i\}}$, hence this access structure admits the formula $\left(\bigwedge_{i \in A} x_i\right) \wedge \bigvee_{i \notin A : A \cup \{i\} \in \Gamma'} x_i$, which has leafsize at most n. The rest of the proof is analogous to the proof of Proposition 5.35. □

5.9.3 Submodular Formal Complexity Measures

A non-negative real-valued function μ defined on the set of monotone Boolean functions in n variables is a *submodular formal complexity measure* if

- $\mu(x_i) \leq 1$ for $i = 1, \ldots, n$,
- $\mu(f \wedge g) + \mu(f \vee g) \leq \mu(f) + \mu(g)$ for all monotone Boolean functions f, g.

For every submodular formal complexity measure μ and for every monotone Boolean function f, it holds that $\mu(f) \leq L(f)$ [166]. See [164, 166] for more details about submodular formal complexity measures.

Proposition 5.38. *Let μ be a submodular formal complexity measure. Then for every two monotone Boolean functions f and f',*

$$|\mu(f) - \mu(f')| \leq n \cdot \mathrm{dist}(f, f').$$

Proof. Let $\Gamma = \Gamma_f$ and $\Gamma' = \Gamma_{f'}$. Let $I = \max(\Gamma \setminus \Gamma')$, let $J = \min(\Gamma' \setminus \Gamma)$, and let g and h be the monotone Boolean functions associated to the access structures $\cap_{A \in I} F_A$ and $\cup_{A \in J} T_A$, respectively. Since $f' = (f \wedge g) \vee h$ and μ is submodular,

$$\begin{aligned} \mu(f') &= \mu((f \wedge g) \vee h) \\ &\leq \mu(f \wedge g) + \mu(h) - \mu((f \wedge g) \wedge h) \\ &\leq \mu(f) + \mu(g) - \mu(f \vee g) + \mu(h) - \mu((f \wedge g) \wedge h) \\ &\leq \mu(f) + \mu(g) + \mu(h). \end{aligned}$$

Since μ is submodular, the size of the monotone formulas described above for T_A and F_A are upper bounds on $\mu(f_{T_A})$ and $\mu(f_{F_A})$ (see [166]). Then

$$\begin{aligned} \mu(g) + \mu(h) = \mu(\bigcap_{A \in I} F_A) + \mu(\bigcup_{A \in J} T_A) &\leq \sum_{A \in I} (n - |A|) + \sum_{A \in J} |A| \\ &\leq n \cdot |I| + n \cdot |J| \leq n \cdot \mathrm{dist}(f, f'). \end{aligned}$$

□

The Razborov rank measure introduced in Section 5.8.1 was originally defined over Boolean functions, and it is submodular (see [166]). Note that the bound we obtained

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

for the Razborov rank measure in Theorem 5.30 is much better than the one in the previous proposition.

The behavior of μ_A and L for close monotone Boolean functions is different. Let f and f' be two monotone Boolean functions at distance ℓ. Let A and A' be matrices over a finite field $\mathbb{F}$ that maximize $\mu_A(f)$ and $\mu_{A'}(f')$, respectively. By Theorem 5.30, the difference $\mu_A(f) - \mu_{A'}(f')$ is at most ℓ, but the difference $L(f) - L(f')$ can be much bigger than ℓ.

The following examples show that the functions κ, σ, σ^{T}, λ, and λ^{T} are neither submodular nor supermodular. Let Γ and Γ' be the access structures on $P = \{1,2,3,4\}$ with $\min\Gamma = \{\{1,2\},\{1,3\},\{2,3\},\{2,4\},\{3,4\}\}$ and $\min\Gamma' = \{\{1,2\},\{1,4\},\{2,3\},\{3,4\}\}$. Both Γ and Γ' admit ideal linear secret sharing schemes, but $\kappa(\Gamma \cap \Gamma') = 3/2$. Hence none of the functions is submodular. Now let Γ and Γ' be the access structures on P with $\min\Gamma = \{\{1,2\},\{2,3\},\{3,4\}\}$ and $\min\Gamma' = \{\{1,2\},\{1,4\},\{3,4\}\}$. Both $\Gamma \cap \Gamma'$ and $\Gamma \cup \Gamma'$ admit ideal linear secret sharing schemes, but $\kappa(\Gamma) = \kappa(\Gamma') = 3/2$. Hence none of the parameters is supermodular.

5.10 Conclusion

The main objective of this work is to discover properties to characterize access structures that admit efficient secret sharing schemes. We show that access structures that are close admit secret sharing schemes with similar information ratios. We also bound the difference between information ratios by the distance between the access structures. Our results are constructive, and we present a formula which, given a secret sharing scheme for a particular access structure, provides schemes for nearby access structures. This formula is simple, but it apparently provides good bounds for both short and large distances (see Sections 5.4.1 and 5.5).

Since access structures that are close admit secret sharing schemes with similar information ratios, in the domain of access structures we have regions in which the access structures admit secret sharing schemes with low information ratio, for instance around ideal access structures. An interesting line of research is to study these regions, and to analyze their distribution and their density in the domain of access structures.

We also provide a combinatorial result that leads to general bounds for the optimal information ratio for access structures whose minimal access structures are close. We translate the search of efficient secret sharing schemes to a combinatorial problem. For graph access structures there exist better constructions [149, 156], but for general access structures our approach is interesting.

The presented techniques are very general, and we extend them to other models of computation, bounding the formula leafsize and the monotone span program size for monotone Boolean functions. Moreover, we believe that our results can also be useful in information theory and coding theory, in particular in network coding and index coding. Our problem can be set as an information-theoretic problem as follows. Suppose that we have a family of random variables, satisfying certain dependence conditions. Then we can modify those conditions and we can consider the problem of constructing new random variables which satisfy the new conditions, while minimizing the their entropy.

We extend our results in order to analyze the techniques for finding lower bounds on the optimal information ratio, and we study their behavior when we add or delete subsets from an access structure. We study bounds based on the Shannon inequalities, the Razborov rank measure, the critical subfamilies method, and submodular formal complexity measures. These bounds are used in other models of computation and information theoretic schemes, and so the obtained results can be useful in other areas.

In the information theoretic setting, another interesting problem is to know the effect of small changes in the dependence conditions. For instance, given an access structure, it would be interesting to study the change in the optimal information ratio if we allow some forbidden subsets to have a certain amount of information about the secret. Since the family of perfect secret sharing schemes cannot capture these situations, we should consider non-perfect secret sharing schemes. In this work we see that the optimal information ratio is 1-Lipschitz, but it would be interesting to study the continuity of the optimal information ratio in this setting. In order to answer this question, we should find a more general description of the access structure of a scheme. For instance, with the access function [161]. For now, the continuity of the optimal information ratio remains an open problem.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Chapter 6

Data Splitting

In this chapter we show our contributions to *privacy-preserving data splitting*. Privacy-preserving data splitting is a technique that aims at protecting data privacy by storing different fragments of data at different locations.

In our work we give a new combinatorial formulation of the data splitting problem. We see the data splitting problem as a purely combinatorial problem in which we have to split data attributes into different fragments in a way that satisfies certain combinatorial properties derived from *processing* and *privacy constraints*. Using this formulation, we develop new combinatorial and algebraic techniques to obtain solutions to the data splitting problem. We present an algebraic method which builds an optimal data splitting solution by using Gröbner bases. Since this method is not efficient in general, we also develop a greedy algorithm for finding solutions that are not necessarily minimal sized.

6.1 Introduction

The size of the data sets collected by companies and organizations and the amount of features demanded when handling data have increased over time, and it is nowadays unfeasible for some data owners to locally store and process data because of the associated costs (such as hardware, energy and maintenance costs). The cloud offers a suitable alternative for data storage, by providing large and highly scalable storage and computational resources at a low cost and with ubiquitous access. However, many data owners are reluctant to embrace the cloud computing technology because of security and privacy concerns, which are mainly centered around the cloud service provider (CSP). The problem is not only that the CSPs may read, use or even sell the data outsourced by their customers, but also that they may suffer attacks or data breaches that can compromise data confidentiality.

Privacy-preserving data splitting is a technique that aims at protecting data privacy by leveraging multi-cloud architectures. Data splitting minimizes the leakage of information by distributing the data among several CSPs, assuming that they do not communicate with each other. Similar problems have been studied in other areas such as data mining, data sanitization, file splitting and data merging.

In general, in data splitting data sets are structured in a tabular format, according to a set of *attributes* (or *features*) identifiable by attribute names. Data is then

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

composed by *records*, where each record holds up to one value per attribute. For instance, we can consider the attributes `'Name','Age','Occupation'`, and a record `{'John','21','Student'}`, where the record holds values for all attributes.

Data splitting comes in three flavours: *horizontal*, *vertical* and *mixed*. In this work, we deal with vertical data splitting, where fragments consist of data on all records, but only contain information on a subset of the attributes. In horizontal data splitting, fragments contain part of the records, and information on all attributes is specified. In mixed data splitting, fragments hold partial information on some records.

In most practical cases, horizontal data splitting is not privacy-preserving by itself, because all the information of an individual register is stored together. Hence, it does not preserve privacy by decomposition [54]. Horizontal data splitting has been used to analyze data collected by different entities on a set of patients [167], or in conjunction with homomorphic encryption [168], to mine horizontally-partitioned data without violating privacy.

Vertical data splitting can be used for privacy-preserving purposes [54, 55]. In particular, in a setting where some combinations of attributes constitute the sensitive information, the data set can be vertically split and distributed among cloud servers so that no CSP holds any sensitive attribute combination. Assuming that CSPs do not communicate with each other, this measure enforces privacy. An example of a sensitive pair of attributes in a medical data setting is passport number and disease, whereas blood pressure and disease constitute a generally safe pair.

The results we present in this work focus on data splitting, but they can be applied to other related areas such as *file splitting*, *data sanitization*, and *data merging*.

In file splitting, pieces of files owned by the same entity are stored in different sites. This is done in such a way that pieces from each site, when considered in isolation, are not sensitive. In [169], the authors spread the data across multiple CSPs and introduce redundancy in order to tolerate possible failures or outages. Their solution follows what is done at the level of disks and file systems in the RAID (Redundant Array of Independent Disks) technology, which strips data across an array of disks and maintains parity data that can be used to reconstruct the contents of any individual failed disk. In [170], user files are categorized and split into chunks, and these chunks are provided to the proper storage servers. The categorization of data is done according to mining sensitivity. To ensure a greater amount of privacy, the possibility of adding misleading data in the chunks depending on the demand of clients is given. Wei et al. [171] proposed a new privacy method that involves bit splitting and bit combination. In their approach, the original files are broken up through bit splitting and each fragment is uploaded to a different storage server.

Data sanitization is the process of removing sensitive information from a document so as to ensure that only the intended information can be accessed. Typically, the result is a document that is suitable for dissemination to the intended audience. Data sanitization has been applied along with data splitting in [1], where the terms in the input document that cause disclosure risk according to the privacy requirements are first detected, and then those terms are distributed in multiple servers in order to prevent disclosure.

Data merging consists on securely splitting and merging data from potentially many sources in a single repository. An approach for data merging is to split and compress

the data into multiple fragments, and to require certain privacy constraints on the fragments [172].

In data splitting, once the data is split, the main issue is how to securely compute over distributed data (see [173] for more details). For some computations, the servers may need to exchange data, but none of them ought to reveal its own private stored information. Computing over distributed data is also studied in the context of parallel processing for statistical computations. In this case, the main difficulty lies in combining partial results obtained from independent processors. Existing related literature reduces statistical analysis to performing either secure distributed scalar products or secure distributed matrix products, e.g. see [174–176]. On a similar note, the field of privacy-preserving data mining deals with the problem of computing over distributed data. It has as its main objective to mine data owned by different parties, who are willing to collaborate in order to get better results, but that do not want or cannot share the raw original data. For instance, see [177–179].

6.1.1 Our Results

In this work we give a new combinatorial formulation to the data splitting problem. In the considered data splitting problem we force some subsets of attributes to be stored separately, because the combination of these attributes may reveal sensitive information to the CSPs. Moreover, we impose some subsets of attributes to be jointly stored in some cloud server, for instance because we want to query on them efficiently or to compute statistics on them (such as data mining or selective correlations). Regarding privacy and security, the CSPs are not trusted and hence they are not given access to the entire original data set. We thus assume that the access of the CSPs to the original data set is restricted to a single chosen fragment.

Our treatment of privacy-preserving data splitting assumes the *honest-but-curious* security model, where the CSPs honestly fill their role in the protocols and do not share information with each other, but they may try to infer information on the data made available to them. In particular, each CSP may be curious to analyze the data it stores and the message flows received during the protocol in order to acquire additional information. Therefore, in our model the information leakage is the sensitive information that can be extracted from single stored data fragments. This model is common in the cloud computing literature, e.g. see [180].

In this setting, our main objective is to minimize the number of different CSPs that are needed to store a data set using data splitting, without applying any other privacy-preserving techniques. To study this problem, we regard the data splitting constraints as consisting of two families of subsets of attributes: the family of subsets of attributes that must be stored together by some CSP, and the family of subsets of attributes that must not be jointly stored by any CSP. These two families respectively define *processing constraints* and *privacy constraints*. We define a data splitting solution as a family of subsets of attributes which satisfies the processing and privacy constraints. Each set in this family should be outsourced to a single CSP in order to enforce privacy. Therefore, we see the data splitting problem as a purely combinatorial problem, in which attributes must be split into different fragments in a way that satisfies the combinatorial properties derived from processing and privacy constraints.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Using this formulation, we develop new combinatorial and algebraic techniques to obtain solutions to the data splitting problem. We first present an algebraic method which uses Gröbner bases to build a data splitting solution with the minimal number of fragments. Since this method has performance issues, we also develop an efficient greedy algorithm for finding solutions that are not generally minimal sized. We compare the efficiency and the accuracy of the two approaches by showing experimental results. Using results of graph theory, we are able to provide necessary and sufficient conditions for the existence of a solution to the data splitting problem, and we give upper and lower bounds on the number of needed fragments.

6.1.2 Related Work

Recently, the data splitting research has focused on finding the minimal sized decomposition of a given data set into privacy-preserving fragments. Related works suggest outsourcing a sensitive data set by vertically splitting it according to some privacy constraints [54–58]. In all previously proposed methods, privacy constraints are described by sensitive pairs of attributes.

In [54], Aggarwal et al. study the problem of finding a decomposition of a given data set into two privacy-preserving vertical fragments, so as to store them in two CSPs which have to be completely unaware of each other. Query execution is also optimized, i.e. they minimize the execution cost of a given query workload while obeying the constraints imposed by the needs of data privacy. Graph-coloring techniques are used to identify a decomposition with small query costs. In particular, their data splitting problem can be reformulated as a hypergraph-coloring instance of a graph G. In case that some sensitive attribute pairs can not be stored separately without increasing the number of fragments to more than two, encryption is used to ensure privacy. To improve the query workload, the storage of the same attribute by both CSPs is allowed.

The optimal decomposition problem described in [54] is hard to solve even if vertex deletion is not allowed. In fact, Guruswami et al. [181] proved that it is NP-hard to color a 2-colorable, 4-uniform hypergraph using only c colors for any constant c. This means that, in the case that all 4-tuples of attributes are sensitive, it is NP-hard to find a partition of attributes into two sets that satisfies all privacy constraints, even knowing that it exists. Because of the hardness of this problem, in [54] they present three different heuristics to solve it.

A later article [55] by Ganapathy et al. studies the same scenario as [54]. Here as well, they consider vertically splitting data into exactly two fragments, though their results are easily extendable to more fragments. They also allow encrypting sensitive attributes and storing the same attribute in both fragments. Three heuristics are introduced to find a decomposition with small query costs. These heuristic search techniques are based on the greedy hill climbing approach, and give a nearly optimal solution.

In [55], the authors study the time complexity of the proposed optimal decomposition problem in terms of the number of attributes. The general problem can theoretically be solved in polynomial time if the collection contains only few sets of constraints, by solving the minimum cut problem. It can also be solved in $O(\log(n))$ time when the problem is equivalent to the hitting set problem. And it can be solved in an approximation factor of $O(\sqrt{n})$ if each constraint set has size 2, by using directed multicat (i.e., solving

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

the minimum edge deletion bipartition problem). The problem becomes intractable if the sets of constraints have size 3. In fact, in this case the problem is reduced to the not-all-equal 3-satisfiability problem, which is an NP-complete problem.

Also, Ciriani et al. [56] present a solution for vertically splitting data into two fragments without requiring the use of encryption, but rather by letting the data owner store a portion of the data set and perform part of the computations.

Another solution presented by Ciriani et al. in [57] uses both encryption and data splitting, but it allows the CSPs to communicate between each other. Because of this assumption, in order to ensure unlinkability between attributes, no attribute must appear in the clear in more that one fragment. In their solution, data is split into possibly more than two different fragments. This lowers the complexity of the problem with respect to [54] and [55]. The optimization problem is then to find a partition that minimizes the number of fragments and maximizes the number of attributes stored in the clear. Also, in this case, the problem of finding a partition of the attribute set is NP-complete. Hence, they present two heuristic methods with time complexity $O(n^2m)$ and $O(n^3m)$, where m is the number of privacy constraints and n is the number of attributes. The first one is based on the definition of vector minimality, and the second one works with an affinity matrix that expresses the advantage of having pairs of attributes in the same fragment.

A similar approach to [57] is also illustrated in [58], where they split a data set into an arbitrary number of non-linkable data fragments and distribute them among an arbitrary number of non-communicating servers.

The data splitting problem studied in this work is also related to other well known combinatorial optimization problems. We want to emphasize the connection with the *job shop scheduling problem*. The job shop scheduling problem consists on assigning jobs to resources at particular times. Welsh and Powell [182] described a basic scheduling problem as follows: let $J = \{J_i\}_{i=1}^N$ be a set of N jobs. Suppose that it takes an entire day to complete each job, and that resources are unbounded. Let $M = \{m_{ij}\}_{i,j=1}^N$ be an incompatibility matrix, where m_{ij} is zero or one depending on whether or not J_i and J_j can be carried out on the same day. The problem consists in scheduling the n jobs using the minimum needed number of days according to the restrictions imposed by M. An efficient algorithm to solve this problem is presented in [182], and subsequent works [183, 184] improve on this solution. See [185] for a survey on this and similar scheduling problems.

By interpreting jobs as attributes, days as data locations and the incompatibility matrix as a set of privacy constraints, we observe the equivalence between the problem posed in [182] and the data splitting problem. Through this same analogy, our setting extends to the following job scheduling problem: let $J = \{J_i\}_{i=1}^N$ be a set of N jobs, and suppose that it takes an entire day to complete each job, and that resources are unbounded. Let $\mathcal{A} \subseteq \mathcal{P}(J)$ be a family of sets of jobs that can not be carried out all on the same day. Similarly, let $\mathcal{B} \subseteq \mathcal{P}(J)$ be a family of sets of jobs that must be carried out all on the same day. The problem consists in scheduling the n jobs using the minimum needed number of days according to the restrictions imposed by $\mathcal{A}$ and $\mathcal{B}$.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

6.1.3 Outline of the Work

Section 6.2 states the problem of privacy-preserving data splitting as a purely combinatorial problem, which consists in splitting sensitive data into several fragments. Section 6.3 presents an algebraic formulation of the combinatorial problem stated in the previous section. Here, Gröbner bases are used to find the optimal (i.e., minimal-sized) solutions. Section 6.4 proposes a polynomial-time method which solves the combinatorial problem. This method sacrifices the solution optimality in favor of efficiency. In addition, an heuristic improvement is also proposed. Section 6.5 presents the experimental results obtained by implementing the methods presented in Sections 6.3 and 6.4. First, we depict a comparison between the methods in a practical situation. Then, a performance analysis of the polynomial-time methods is carried out over random graphs. Finally, we list some conclusions in Section 6.6.

6.2 A Combinatorial Approach

In this section we state the problem of privacy-preserving data splitting as a purely combinatorial problem. This problem consists in splitting a given data set in which some attributes are sensitive. As discussed above, this situation also covers problems of file splitting, data sanitization and data merging. First we introduce some notation.

Let P be a set and let $\mathcal{C} \subseteq \mathcal{P}(P)$. For any $i \in P$, we define $\deg_i(\mathcal{C})$ as the number of subsets of $\mathcal{C}$ containing i, and we define the degree $\deg(\mathcal{C})$ of $\mathcal{C}$ as the maximum of $\deg_i(\mathcal{C})$ for every $i \in P$. For any $B \subseteq P$, we also denote by $\deg_B(\mathcal{C})$ the number of subsets $A \in \mathcal{C}$ such that $A \cap B \neq \emptyset$. For any $i \in P$ we define $\deg_i(\mathcal{C}) = \deg_{\{i\}}(\mathcal{C})$. Given a set $A \subseteq P$, we define its closure $\mathrm{cl}(A) = \{B \subseteq P : A \subseteq B\}$. We define $\min \mathcal{C}$ and $\max \mathcal{C}$ as follows. A subset $A \subseteq P$ is in $\min \mathcal{C}$ if and only if $A \in \mathcal{C}$ and there does not exist $B \in \mathcal{C}$ with $B \subsetneq A$. Analogously, a subset $A \subseteq P$ is in $\max \mathcal{C}$ if and only if $A \in \mathcal{C}$ and there does not exist $B \in \mathcal{C}$ with $A \subsetneq B$. That is, a $\min \mathcal{C}$ is the family of *minimal subsets* in $\mathcal{C}$, and $\max \mathcal{C}$ is the family of *maximal subsets* in $\mathcal{C}$. We say that $\mathcal{C}$ is an *antichain* if $A \nsubseteq B$ for every $A, B \in \mathcal{C}$. In this case, $\mathcal{C} = \min \mathcal{C} = \max \mathcal{C}$.

In the considered data splitting setting we have a set of attributes P, and some combinations of the attributes are not to be stored by any individual server because they would leak sensitive information. We assume that individual attributes, when considered in isolation, are not sensitive (otherwise, encryption can be used). Moreover, we want some other attributes to be stored in the same location, for example to perform statistical analysis computations such as contingency tables, correlations or principal component analysis of the attributes. We thus describe a *data splitting problem* using two families of attributes: $\mathcal{A} \subseteq \mathcal{P}(P)$ is the family of subsets of attributes that cannot be stored jointly in any single server, and $\mathcal{B} \subseteq \mathcal{P}(P)$ is the family of subsets of attributes that must be stored together in some server. We state the data splitting problem in terms of $(\mathcal{A}, \mathcal{B})$-*coverings*, a notion first introduced in [84].

Definition 6.1. Let $\mathcal{A}, \mathcal{B} \subseteq \mathcal{P}(P)$. An $(\mathcal{A}, \mathcal{B})$-covering $\mathcal{C}$ is a family of subsets of P satisfying that

1. for every $A \in \mathcal{A}$ and for every $C \in \mathcal{C}$, $A \nsubseteq C$, and
2. for every $B \in \mathcal{B}$ there exists $C \in \mathcal{C}$ with $B \subseteq C$.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Let $\mathcal{A}, \mathcal{B}$ be the families of subsets defined by the data splitting restrictions described above, and let $\mathcal{C}$ be an $(\mathcal{A}, \mathcal{B})$-covering. Then $\mathcal{C}$ defines a solution for data splitting by associating each fragment i with a set $C_i \in \mathcal{C}$. That is, we solve the data splitting problem by storing the data corresponding to attributes in C_i at the i-th location. Observe that, according to this definition, for each $B \in \mathcal{B}$ there is at least one fragment containing all attributes in B, and none of the fragments contain all attributes in A for any $A \in \mathcal{A}$. These are exactly the restrictions we have for data splitting. Note that we distribute data in as many fragments as $|\mathcal{C}|$. Since $\mathcal{B} \subseteq \mathcal{P}(P)$ is the family of subsets of attributes that we want to be stored together, we will always assume that $\cup_{B \in \mathcal{B}} B = P$.

Our work is focused on minimizing the size of the coverings, which corresponds to the number of fragments in data splitting. Therefore, we say that $\mathcal{C}$ is an *optimal* $(\mathcal{A}, \mathcal{B})$-covering if $|\mathcal{C}|$ is minimal among all $(\mathcal{A}, \mathcal{B})$-coverings. Also, it could be desirable to minimize $\sum_{X \in \mathcal{C}} |X|$, which corresponds to the total amount of information that will be stored, and $\max_{i \in P} \deg_i(\mathcal{C})$, which corresponds to the maximum redundancy in the storage.

Example 6.1. *Let $\mathcal{B} \subseteq \mathcal{P}(P)$ be an antichain, and let $A \in \mathcal{B}$. Then $\mathcal{C} = \{P \setminus \{i\} : i \in A\}$ is an $(\{A\}, \mathcal{B} \setminus \{A\}) - covering$.*

Next we present some technical results about coverings. The main results of this section are Proposition 6.4, which characterizes the existence of coverings, and Proposition 6.6, which justifies the search for $(\mathcal{A}, \mathcal{B})$-coverings in the case that $\mathcal{A}$ and $\mathcal{B}$ are antichains. In addition, we present a theoretical lower bound on the size of $(\mathcal{A}, \mathcal{B})$-coverings in Proposition 6.7.

Lemma 6.2. *Let $\mathcal{A}, \mathcal{B}, \mathcal{C} \subseteq \mathcal{P}(P)$. Then $\mathcal{C}$ is an $(\mathcal{A}, \mathcal{B})$-covering if and only if*

1. $\mathrm{cl}(A) \cap \mathcal{C} = \emptyset$ *for every* $A \in \mathcal{A}$, *and*
2. $\mathrm{cl}(B) \cap \mathcal{C} \neq \emptyset$ *for every* $B \in \mathcal{B}$.

Proof. Let $\mathcal{C}$ be an $(\mathcal{A}, \mathcal{B})$-covering. For every $A \in \mathcal{A}$ and for every $C \in \mathcal{C}$ it holds that $A \nsubseteq C$, and so for any $A' \in \mathrm{cl}(A)$ we have $A' \nsubseteq C$. Hence $\mathrm{cl}(A) \cap \mathcal{C} = \emptyset$. For every $B \in \mathcal{B}$, there exists $C \in \mathcal{C}$ with $B \subseteq C$, i.e. $C \in \mathrm{cl}(B)$. Hence $\mathrm{cl}(B) \cap \mathcal{C} \neq \emptyset$. This concludes the proof of one implication.

For any $A \subseteq P$, if $\mathrm{cl}(A) \cap \mathcal{C} = \emptyset$ then $A \nsubseteq C$ for every $C \in \mathcal{C}$. For any $B \subseteq P$, if $\mathrm{cl}(B) \cap \mathcal{C} \neq \emptyset$ then there exists $C \in \mathcal{C}$ with $B \subseteq C$. Hence the converse implication holds. □

As a direct consequence of this lemma, we have the following result.

Lemma 6.3. *Let $\mathcal{A}, \mathcal{A}', \mathcal{B}, \mathcal{B}' \subseteq \mathcal{P}(P)$ with $\mathcal{A}' \subseteq \mathcal{A}$ and $\mathcal{B}' \subseteq \mathcal{B}$. Every $(\mathcal{A}, \mathcal{B})$-covering is also an $(\mathcal{A}', \mathcal{B}')$-covering.*

The next proposition characterizes the pairs of subsets $(\mathcal{A}, \mathcal{B})$ admitting $(\mathcal{A}, \mathcal{B})$-coverings.

Proposition 6.4 (Condition 6.1)**.** *Let $\mathcal{A}, \mathcal{B} \subseteq \mathcal{P}(P)$. There exists an $(\mathcal{A}, \mathcal{B})$-covering if and only if*

$$A \nsubseteq B \text{ for every } A \in \mathcal{A} \text{ and } B \in \mathcal{B}. \tag{6.1}$$

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Proof. Let $\mathcal{C}$ be an $(\mathcal{A},\mathcal{B})$-covering. By Lemma 6.2, for every $A \in \mathcal{A}$ and $B \in \mathcal{B}$, $\mathrm{cl}(A) \cap \mathcal{C} = \emptyset$ and $\mathrm{cl}(B) \cap \mathcal{C} \neq \emptyset$, so $A \nsubseteq B$. Conversely, if $A \nsubseteq B$ for every $A \in \mathcal{A}$ and $B \in \mathcal{B}$, then $\mathcal{B}$ is an $(\mathcal{A},\mathcal{B})$-covering. □

Lemma 6.5. *Let $\mathcal{A},\mathcal{A}',\mathcal{B},\mathcal{B}' \subseteq \mathcal{P}(P)$. If*

- *for every $A' \in \mathcal{A}'$ there exists $A \in \mathcal{A}$ with $A \subseteq A'$, and*
- *for every $B' \in \mathcal{B}'$ there exists $B \in \mathcal{B}$ with $B' \subseteq B$,*

then any $(\mathcal{A},\mathcal{B})$-covering is also an $(\mathcal{A}',\mathcal{B}')$-covering.

Proof. Let $\mathcal{C}$ be an $(\mathcal{A},\mathcal{B})$-covering. Let $A' \in \mathcal{A}'$ and let $A \in \mathcal{A}$ with $A \subseteq A'$. For every $C \in \mathcal{C}$ we have $A \nsubseteq C$, and so $A' \nsubseteq C$. Now let $B' \in \mathcal{B}'$ and let $B \in \mathcal{B}$ with $B' \subseteq B$. Then there exists a subset $C \in \mathcal{C}$ satisfying $B \subseteq C$, which also satisfies $B' \subseteq C$. Hence $\mathcal{C}$ is an $(\mathcal{A}',\mathcal{B}')$-covering. □

Proposition 6.6. *Let $\mathcal{A},\mathcal{B},\mathcal{C} \subseteq \mathcal{P}(P)$. Then $\mathcal{C}$ is an $(\mathcal{A},\mathcal{B})$-covering if and only if it is a $(\min\mathcal{A},\max\mathcal{B})$-covering.*

Proof. By Lemma 6.3, every $(\mathcal{A},\mathcal{B})$-covering is an $(\min\mathcal{A},\max\mathcal{B})$-covering. The converse implication is a direct consequence of Lemma 6.5. □

According to the previous proposition, we can always restrict the search for $(\mathcal{A},\mathcal{B})$-coverings to the case where $\mathcal{A}$ and $\mathcal{B}$ are antichains. Further, as a consequence of Lemma 6.5 we can define a partial hierarchy among the pairs of antichains $(\mathcal{A},\mathcal{B})$. For example, every $(\{\{1,2\}\},\{\{3,4,5\}\})$-covering is also an $(\{\{1,2,3\}\},\{\{3,4\}\})$-covering.

To conclude this section, we describe a theoretical lower bound on the size of $(\mathcal{A},\mathcal{B})$-coverings. Note that, in the case $\mathcal{B} = \binom{P}{1} = \{\{i\} : i \in P\}$ and $\mathcal{A} \subseteq \binom{P}{2}$, the problem of finding an $(\mathcal{A},\mathcal{B})$-covering is equivalent to the graph coloring problem on the graph $G = (P,\mathcal{A})$. In this case, the size of an optimal $(\mathcal{A},\mathcal{B})$-covering is just the chromatic number $\chi(G)$.

Existing general lower bounds on the chromatic number include the clique number, the minimum degree bound, Hoffman's bound, the vector chromatic number, Lovász number and the fractional chromatic number. Our proposed bound generalizes the minimum degree bound $\chi(G) \geq \frac{n}{n-\delta(G)}$, where n is the number of vertices and $\delta(G)$ is the minimum degree of G, to the case of $(\mathcal{A},\mathcal{B})$-coverings.

Proposition 6.7. *Let $\mathcal{A},\mathcal{B} \subseteq \mathcal{P}(P)$ be families of subsets satisfying condition 6.1, and let $\mathcal{C}$ be an $(\mathcal{A},\mathcal{B})$-covering. Then*

$$|\mathcal{C}| \geq \frac{|\mathcal{B}|}{|\mathcal{B}| - \max_{A\in\mathcal{A}} \min_{a\in A} \deg_a(\mathcal{B})}.$$

Proof. Let $\mathcal{C}$ be an $(\mathcal{A},\mathcal{B})$-covering. Given $C \in \mathcal{C}$, denote $\mathcal{B}_C = \mathcal{B} \cap \mathcal{P}(C)$.

By the properties of $(\mathcal{A},\mathcal{B})$-coverings we have that $\mathcal{B} \subseteq \cup_{C\in\mathcal{C}}\mathcal{P}(C)$, and this implies that $\mathcal{B} = \cup_{C\in\mathcal{C}}\mathcal{B}_C$. Hence $|\mathcal{B}| \leq \sum_{C\in\mathcal{C}} |\mathcal{B}_C| \leq |\mathcal{C}| \cdot \max_{C\in\mathcal{C}} |\mathcal{B}_C|$, so $|\mathcal{C}| \geq |\mathcal{B}| / \max_{C\in\mathcal{C}} |\mathcal{B}_C|$. We now proceed to upper bound $\max_{C\in\mathcal{C}} |\mathcal{B}_C|$.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Since for every $B \in \mathcal{B}_C$ we have $B \subseteq C$, we see that $\cup_{B\in\mathcal{B}_C} B \subseteq C$. Therefore, by the definition of $(\mathcal{A},\mathcal{B})$-coverings we have that $A \nsubseteq \cup_{B\in\mathcal{B}_C} B$ for every $A \in \mathcal{A}$. Denote by $\alpha(\mathcal{A},\mathcal{B})$ the size of the largest subfamily of $\mathcal{B}$ with this property, i.e.

$$\alpha(\mathcal{A},\mathcal{B}) = \max\{|\mathcal{B}'| : \mathcal{B}' \subseteq \mathcal{B} \text{ and } A \nsubseteq \cup_{B\in\mathcal{B}'} B \text{ for every } A \in \mathcal{A}\}.$$

By the preceding observation, we get that $\max_{C\in\mathcal{C}} |\mathcal{B}_C| \leq \alpha(\mathcal{A},\mathcal{B})$. By definition of $\alpha(\mathcal{A},\mathcal{B})$, given any set $A \in \mathcal{A}$ we have $\alpha(\mathcal{A},\mathcal{B}) \leq \alpha(\{A\},\mathcal{B})$, and so $\alpha(\mathcal{A},\mathcal{B}) \leq \min_{A\in\mathcal{A}} \alpha(\{A\},\mathcal{B})$. Now, given a set $A \in \mathcal{A}$, a family $\mathcal{B}' \subseteq \mathcal{B}$ satisfies $A \nsubseteq \cup_{B\in\mathcal{B}'} B$ if and only if there exists an element $a \in A$ such that $a \notin \cup_{B\in\mathcal{B}'} B$. Therefore $\alpha(\{A\},\mathcal{B}) = \max_{a\in A} \alpha(\{a\},\mathcal{B})$. Finally, by definition we see that $\alpha(\{a\},\mathcal{B}) = |\mathcal{B}| - \deg_a(\mathcal{B})$. By composing the obtained results, we see that $\max_{C\in\mathcal{C}} |\mathcal{B}_C| \leq \min_{A\in\mathcal{A}} \max_{a\in A}(|\mathcal{B}| - \deg_a(\mathcal{B})) = |\mathcal{B}| - \max_{A\in\mathcal{A}} \min_{a\in A} \deg_a(\mathcal{B})$. The proposition follows by applying the first obtained inequality. □

6.2.1 Multi-Colorings of Hypergraphs

In order to construct $(\mathcal{A},\mathcal{B})$-coverings, we will use colorings of hypergraphs. Let $\mathcal{H} = (P,E)$ be a hypergraph. A *coloring* of $\mathcal{H}$ with k colors is a mapping $\mu : P \to \{1,\ldots,k\}$ such that, for every $A \in E$, there exists $u, v \in A$ with $\mu(u) \neq \mu(v)$.

Next, we describe the connection between colorings and coverings. Let μ be a coloring of the hypergraph $\mathcal{H} = (P,\mathcal{A})$ with k colors. Consider the family of subsets of elements in P that have the same color under μ. That is, consider a family of subsets $\mathcal{C} = \{C_1,\ldots,C_k\}$ that is a partition of P satisfying that $\mu(j) = i$ for every $j \in C_i$ and for every color i.

Now consider the pair $(\mathcal{A},\mathcal{B})$ with $\mathcal{B} = \binom{P}{1}$. Observe that $\mathcal{C}$ satisfies condition 1 in Definition 6.1 because, if a subset A is in $\mathcal{A}$, then it cannot be monochromatic. Since each element in P has a color, condition 2 is also satisfied. In order to construct $(\mathcal{A},\mathcal{B})$-coverings for other families of subsets $\mathcal{B} \subseteq \mathcal{P}(P)$, we can use sequences of colorings. To appropriately formalize these constructions, we use hypergraph multi-colorings.

For any integer $k > 0$, we define a *multi-coloring* of the hypergraph $\mathcal{H} = (P,E)$ of k colors as a mapping $\mu : P \to \{0,1\}^k$ with the following property: for every $A \in E$ and for every $1 \leq j \leq k$, there exists $i \in A$ for which the j-th coordinate of $\mu(i)$ is 0, namely $\mu(i)_j = 0$. If we associate each $1 \leq j \leq k$ with a different color, a multi-coloring of $\mathcal{H}$ is a mapping that maps each element in P to a set of at most k colors. This mapping must satisfy that for every subset in $A \in E$ and for each color, at least one element in A does not have this color. A sequence of colorings of a hypergraph defines a multi-coloring. A multi-coloring defines a family of subsets in a natural way, and vice-versa. More concretely, a multi-coloring μ induces the family of subsets $\mathcal{C} = \{C_i\}_{1\leq i\leq k} \subseteq \mathcal{P}(P)$, where C_i are the subsets of elements of P mapped to the same color i by μ.

Lemma 6.8. *Let $\mathcal{A},\mathcal{B},\mathcal{C} \subseteq \mathcal{P}(P)$, with $|\mathcal{C}| = k$. Then $\mathcal{C}$ is an $(\mathcal{A},\mathcal{B})$-covering if and only if $\mathcal{C}$ defines a multi-coloring μ of $\mathcal{H} = (P,\mathcal{A})$ of k colors with the property that, for every $B \in \mathcal{B}$, there exists $1 \leq j \leq k$ for which the j-coordinate of $\mu(i)$ is 1 for every $i \in B$.*

Proof. Let $\mathcal{C} = \{C_1,\ldots,C_k\}$ be an $(\mathcal{A},\mathcal{B})$-covering . We define a multi-coloring μ of k colors as follows. For every $i \in P$ and $1 \leq j \leq k$, set $\mu(i)_j = 1$ if and only if i is in $\mathcal{C}_j$. Let $B \in \mathcal{B}$, and let C_j be a subset in $\mathcal{C}$ with $B \subseteq C_j$. Then $\mu(i)_j = 1$ for every $i \in B$.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Taking into account the comments detailed above, it is straightforward to prove that the converse implication also holds. □

We use the connection between coverings and multi-colorings to find general constructions of coverings and upper bounds on their size. Beimel, Farràs, and Mintz constructed efficient secret sharing schemes for very dense graphs [149]. One of the techniques developed in that work is connected to our work. In [84], this result was described in terms of $(\mathcal{A}, \mathcal{B})$-coverings for $\mathcal{B} = \binom{P}{k} \setminus \mathcal{A}$. Due to Lemma 6.3, if $\mathcal{A} \subseteq \binom{P}{k}$, the biggest family of subsets $\mathcal{B} \subseteq \binom{P}{k}$ admitting an $(\mathcal{A}, \mathcal{B})$-covering is $\mathcal{B} = \binom{P}{k} \setminus \mathcal{A}$. The next lemma states the results described above in a more general way. A partial proof can be found in [149] and in Theorem 5.15 of the previous chapter.

Lemma 6.9. *Let $\mathcal{A}, \mathcal{B} \subseteq \mathcal{P}(P)$ be families of subsets satisfying condition 6.1. Let d denote the degree of $\mathcal{A}$, and suppose that sets in $\mathcal{A}$ and $\mathcal{B}$ have size at most k. Then there exists an $(\mathcal{A}, \mathcal{B})$-covering of degree $2(2kd)^{k-1} \ln n$ and size $2(2kd)^k \ln n$.*

6.2.2 Optimal Covers

Both the optimization problem of determining the size of an optimal $(\mathcal{A}, \mathcal{B})$-covering and the search problem of finding an optimal $(\mathcal{A}, \mathcal{B})$-covering are NP-hard. This is so because taking $\mathcal{B} = \binom{P}{1}$ and $\mathcal{A} \subseteq \binom{P}{2}$ transforms these problems to the corresponding graph coloring problems, and so there is a trivial reduction from the known NP-hard graph coloring problems to the $(\mathcal{A}, \mathcal{B})$-covering problems. Next, we see NP-completeness of the decisional problem.

Proposition 6.10. *The problem of deciding whether an $(\mathcal{A}, \mathcal{B})$-covering of size t exists is NP-complete.*

Proof. Let $\mathcal{A}, \mathcal{B}, t$ define an instance of the problem where the answer is affirmative. Given an $(\mathcal{A}, \mathcal{B})$-covering $\mathcal{C}$ of size t, a checking algorithm first verifies that $\mathcal{C}$ has size t, that every $B \in \mathcal{B}$ is contained in some $X \in \mathcal{C}$, and that no $A \in \mathcal{A}$ is contained in any $X \in \mathcal{C}$. The running time of this checking algorithm is at most quadratic in the size of the problem input, and thus the given problem is in NP.

Now, note that the case $\mathcal{B} = \binom{P}{1}$ and $\mathcal{A} \subseteq \binom{P}{2}$ is equivalent to the graph coloring problem. Therefore the given problem is NP-complete. □

6.3 Algebraic Formulation of the Problem

In this section we present an algebraic formulation of the combinatorial problem presented in the previous section. The purpose of this formulation is to exploit algebraic techniques to find solutions to the data splitting problem for a fixed number of fragments.

It is not unusual that graph-coloring problems are encoded in polynomial ideals [186–189]. In this case, the existence of a coloring is reduced to the solvability of a related system of polynomial equations over the algebraic closure of the field. Furthermore, the weak Hilbert's Nullstellensatz theorem allows to obtain a *certificate* that a system of polynomial has no solutions [190], and, consequently, that a graph is not colorable.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

The focus of this section is the use of polynomial ideals and Gröbner bases to provide an optimal multi-coloring μ with the property described in Lemma 6.8. Recall that obtaining a multi-coloring μ is equivalent to finding an $(\mathcal{A}, \mathcal{B})$-covering.

Let $\mathcal{H} = (P, \mathcal{A})$ be a hypergraph and let μ be a multi-coloring of $\mathcal{H} = (P, \mathcal{A})$ of k colors with the property that, for every $B \in \mathcal{B}$, there exists $1 \leq j \leq k$ for which the j-coordinate of $\mu(i)$ is 1 for every $i \in B$. The multi-coloring μ can be seen as assignment of $\{0, 1\}$ values to a set of kn variables $x_{i,j}$, where $n = |P|$, $1 \leq j \leq k$, $1 \leq i \leq n$ and $x_{i,j} = 1$ if and only if $\mu(i)_j = 1$. In other words, we assign k variables $x_{i,j}$ to each vertex i in P in such a way that,

$$x_{i,j} = \begin{cases} 1 & \text{if vertex } i \text{ takes color } j \text{ by applying } \mu \\ 0 & \text{otherwise.} \end{cases}$$

Encoding μ to a polynomial ring allows an algebraic formulation of the multi-coloring problem. Since we focus on optimal multi-colorings, the number of colors is fixed to a designated minimal k. Furthermore, each variable $x_{i,j}$ takes values in $\{0, 1\}$, which allows working over $\mathbb{F}_2$.

Therefore, given $X = (x_{i,j})_{1 \leq j \leq k, 1 \leq i \leq n}$, we define the *$k$-coloring ideal* $I_k(\mathcal{H}, \mathcal{B}) \subset \mathbb{F}_2[X]$ as the ideal generated by:

- $G_1 = \{\prod_{i \in A} x_{i,j} : 1 \leq j \leq k, A \in \mathcal{A}\}$
  - all vertices belonging to an edge set $A \in \mathcal{A}$ cannot have the same color,
- $G_2 = \{\prod_{j=1}^{k} (\prod_{i \in B} x_{i,j} - 1) : B \in \mathcal{B}\}$
  - there exists a color j such that all the vertices in B are colored with j.

Theorem 6.11 proves that finding a solution of $I_k(\mathcal{H}, \mathcal{B})$ is equivalent to obtaining a suitable multi-coloring μ.

Theorem 6.11. *Let $\mu : P \to \{0, 1\}^k$ be a multi-coloring of $\mathcal{H} = (P, \mathcal{A})$, and assume that $\cup_{B \in \mathcal{B}} B = P$. Then μ defines an $(\mathcal{A}, \mathcal{B})$-covering (in the sense of Lemma 6.8) if and only if $I_k(\mathcal{H}, \mathcal{B})$ has a common root in $\mathbb{F}_2[X]$. In other words, the multi-coloring μ of $\mathcal{H}$ does not define an $(\mathcal{A}, \mathcal{B})$-covering if and only if $I_k(\mathcal{H}, \mathcal{B}) = (1)$.*

Proof. According to the definition above, the map μ is a multi-coloring if:

- for every $A \in \mathcal{A}$ and for every $1 \leq j \leq k$, there exists $i \in A$ for which $\mu(i)_j$ is 0,
- for every $B \in \mathcal{B}$, there exists $1 \leq j \leq k$ for which the j-coordinate of $\mu(i)$ is 1 for every $i \in B$.

It is known that if a polynomial e_1 encodes a property and e_2 encodes another property, then the ideal generated by e_1 and e_2 encodes the conjunction (i.e., the *and*) of the properties. Therefore, if G_1 and G_2 encode the properties above, respectively, then $I_k(\mathcal{H}, \mathcal{B})$ encodes μ. We see that this is indeed the case:

- G_1: for all $A \in \mathcal{A}$ and for every color j, we have that $\prod_{i \in A} x_{i,j} \in I_k(\mathcal{H}, \mathcal{B})$. This happens if and only if $\prod_{i \in A} x_{i,j} = 0$, which means that there exists $i \in A$ such that $x_{i,j} = 0$. This is equivalent to say that there exists $i \in A$ such that $\mu(i)_j = 0$.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

- G_2: for all $B \in \mathcal{B}$, we have $\prod_{j=1}^{k}(\prod_{i\in B} x_{i,j} - 1) \in I_k(\mathcal{H}, \mathcal{B})$. This happens if and only if $\prod_{j=1}^{k}(\prod_{i\in B} x_{i,j} - 1) = 0$, which means that there exists j such that $\prod_{i\in B} x_{i,j} = 1$. This amounts to say that there exists j such that $x_{i,j} = 1$ for all i, which is equivalent to saying that there exists j such that $\mu(i)_j = 1$ for all $i \in B$.

□

Observe that imposing $\cup_{B\in\mathcal{B}} B = P$ is not restrictive. In fact, it is always possible to add the singletons of any vertices to $\mathcal{B}$ in order to guarantee that every vertex is assigned a color, without changing the request of the problem (see Example 6.2 for more details). In particular, if $\cup_{B\in\mathcal{B}} B = P$, then for all $i \in P$ there exists $B \in \mathcal{B}$ such that $i \in B$, and therefore there exists a color j such that $x_{i,j} = 1$, which amounts to saying that there exists a color j such that $\mu(i)_j = 1$. The hypothesis $\cup_{B\in\mathcal{B}} B = P$ has also been stated in Section 6.2.

Now that the data splitting problem is stated as an algebraic problem, a technique based on Gröbner bases can be used to solve it. A Gröbner basis is a generating set of an ideal I in a polynomial ring, which allows to determine if any polynomial belongs to I or not [190]. In other words, it allows to determine the variety associated to I, i.e. the solutions of I. It is proven that it is possible to associate a Gröbner basis to every polynomial ideal [190]. Informally, Gröbner basis computation can be viewed as a generalization of Gaussian elimination for non-linear equations.

In our case, Gröbner bases can be used to find the solutions of $I_k(\mathcal{H}, \mathcal{B})$. Once the Gröbner basis of the k-coloring ideal is obtained, the associated variety can be easily computed. The complexity of computing the Gröbner basis of a system of polynomial equations of degree d in n variables has been proven to be $d^{O(n^2)}$ when the number of solutions is finite [191]. In general, its complexity is $2^{2^{O(n)}}$. Since $I_k(\mathcal{H}, \mathcal{B})$ belongs to $\mathbb{F}_2[X]$, then it has a finite number of solutions, and so a bound for the worst-case complexity of computing Gröbner basis in this case is $(kn)^{O(n^2)}$. The complexity of computing Gröbner bases is at least that of polynomial-system solving.

As stated before, it is possible to derive a certificate for the unsolvability of a system of polynomials from the weak Hilbert's Nullstellensatz. In our case, this allows to prove that it is not possible to find a multi-coloring μ with a designated number of colors k.

Theorem 6.12 (Weak Hilbert's Nullstellensatz [190])**.** *Let $f_1, \ldots, f_m \in \mathbb{K}[x_1, \ldots, x_n]$. Then there are no solutions to the system $\{f_i = 0\}_{i=1,\ldots,m}$ in the algebraic closure of $\mathbb{K}$ if and only if there exist $\alpha_1, \ldots, \alpha_m \in \mathbb{K}[x_1, \ldots, x_n]$ such that*

$$\alpha_1 f_1 + \cdots + \alpha_m f_m = 1 \tag{6.2}$$

The set $\{\alpha_i\}$ is called a *Nullstellensatz certificate*. The complexity of computing such a certificate depends on the degree of $\{\alpha_i\}_{i=1}^{m}$, which is defined as the maximum degree of any α_i as polynomials. Fast results have been achieved when the Nullstellensatz certificate has small constant degree [192].

According to Theorem 6.12, by using methods to compute a Nullstellensatz certificate it is possible to find out whether or not $I_k(\mathcal{H}, \mathcal{B})$ has any solutions. If we start by fixing a tentative number of colors k, the data splitting problem can be solved by applying a Nullstellensatz certificate method. When there exists a Nullstellensatz certificate,

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

$I_k(\mathcal{H},\mathcal{B})$ does not have common root. The complexity of the Nullstellensatz certificate and the Gröbner basis methods grow with the number of variables which, in our case, grows with the number of colors. Therefore, it is convenient to start with as few colors as possible and increase them sequentially until a certificate of feasibility is found or until a Gröbner basis is computed. Notice that finding the optimal k is a NP-complete problem, because it has complexity equivalent to solving the system of equations.

Example 6.2. *Given* $\mathcal{A}=\{\{1,2,3\}\}$ *and* $\mathcal{B}=\{\{1,4\},\{2,4\},\{3\}\}$*, we want to compute an* $(\mathcal{A},\mathcal{B})$*-covering. As explained above, the problem can be encoded to polynomial ideals. We assign* k *variables to each attribute, where* k *is the number of colors needed to obtain an optimal covering. For example, we can encode vertex* 1 *to* $x_{1,1}$ *and* $x_{1,2}$ *when* 2 *colors are considered. The variable* $x_{i,j}$ *equals* 1 *if and only if vertex* i *takes color* j*, and it equals* 0 *otherwise.*

The ideal $I_2(\mathcal{H},\mathcal{B})$ *is generated by the polynomials in* G_1 *and* G_2*, where*

- $G_1=\{x_{1,1}x_{2,1}x_{3,1},\ x_{1,2}x_{2,2}x_{3,2}\}$.
- $G_2=\{x_{2,1}x_{4,1}x_{2,2}x_{4,2}+x_{2,1}x_{4,2}+x_{2,2}x_{4,1}+1,\ x_{1,1}x_{4,1}x_{1,2}x_{4,2}+x_{1,1}x_{4,1}+x_{1,2}x_{4,2}+1,\ x_{3,1}x_{3,2}+x_{3,1}+x_{3,2}+1\}$.

Note that there does not exist an $(\mathcal{A},\mathcal{B})$*-covering of size one, because* $\{1,2,3\}\subseteq\cup_{B\in\mathcal{B}}B$*. Since we can compute the Gröbner basis of* $I_2(\mathcal{H},\mathcal{B})$*, there exist* $(\mathcal{A},\mathcal{B})$*-coverings of size two, and we obtain the optimal* $(\mathcal{A},\mathcal{B})$*-coverings:*

$$\{x_{1,1},\ x_{2,1}+1,\ x_{3,1},\ x_{4,1}+1,\ x_{1,2}+1,\ x_{2,2},\ x_{3,2}+1,\ x_{4,2}+1\},$$
$$\{x_{1,1},\ x_{2,1}+1,\ x_{3,1}+1,\ x_{4,1}+1,\ x_{1,2}+1,\ x_{2,2},\ x_{3,2},\ x_{4,2}+1\},$$
$$\{x_{1,1},\ x_{3,1}+1,\ x_{1,2}+1,\ x_{2,2}+1,\ x_{3,2},\ x_{4,2}+1\}.$$

In the last solution, the variables $x_{2,1}$ *and* $x_{4,1}$ *are missing, which means that they can take both* 0 *and* 1 *values. Therefore, the solutions can be re-written as the following coverings:*

$$\{\{2,4\},\{1,3,4\}\},$$
$$\{\{2,3,4\},\{1,4\}\},$$
$$\{\{3\},\{1,2,4\}\},\ \{\{2,3\},\{1,2,4\}\},\ \{\{3,4\},\{1,2,4\}\},\ \{\{2,3,4\},\{1,2,4\}\}.$$

To obtain the number of colors k *which allows to compute an optimal covering, a small number of colors* $k=2$ *is fixed. If the Gröbner basis method applied to* $I_2(\mathcal{H},\mathcal{B})$ *outputs the ideal* (1)*, then* k *is incremented. This process is repeated until an ideal different from* (1) *is obtained. The resulting number of colors* k *is the smallest one for which we have an* $(\mathcal{A},\mathcal{B})$*-covering, and thus it is optimal.*

6.4 A Greedy Algorithm

In this section we aim for an efficient method to build $(\mathcal{A},\mathcal{B})$-coverings and for upper bounds on the size of an optimal $(\mathcal{A},\mathcal{B})$-covering.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

As seen above, the problem of finding an optimal $(\mathcal{A}, \mathcal{B})$-covering is NP-hard. Hence, as expected, the labour involved in finding an optimal $(\mathcal{A}, \mathcal{B})$-covering can render methods inefficient when solving practical data splitting instances. Our strategy to circumvent this consists in sacrificing optimality to achieve a polynomial-time algorithm.

The problem of finding upper bounds on the size of an optimal $(\mathcal{A}, \mathcal{B})$-covering $\mathcal{C}$ has been studied in the literature for the following particular cases:

- In the case $\mathcal{B} = \binom{P}{1}$ and $\mathcal{A} \subseteq \binom{P}{2}$, the problem of finding an $(\mathcal{A}, \mathcal{B})$-covering is easily seen to be equivalent to the graph coloring problem. Then $|\mathcal{C}|$ is the chromatic number of the graph $G = (\mathcal{B}, \mathcal{A})$. For instance, the greedy coloring bound gives $|\mathcal{C}| \leq \deg(\mathcal{A}) + 1$.
- The case $\mathcal{B} \subseteq \binom{P}{2} \cup \binom{P}{1}$ and $\mathcal{A} = \binom{P}{2} \backslash \mathcal{B}$ has been studied as the *clique covering* and the *clique partition numbers*. Hall [193] and Erdős et al. [194] showed that $|\mathcal{C}| \leq \lfloor |P|^2/4 \rfloor$.
- In the case $\mathcal{B} = \binom{P}{l}$ and $\mathcal{A} \subseteq \cup_{i>k} \binom{P}{i}$ for $1 \leq l \leq k < n$, the problem of finding a k-uniform $(\mathcal{A}, \mathcal{B})$-covering is equivalent to finding an (n, k, l)-*covering design*. In this case, Spencer [195] showed that $|\mathcal{C}| \leq \binom{n}{l} \Big/ \binom{k}{l} \left(1 + \ln \binom{k}{l}\right)$.

In the following, we first describe a general upper bound on the size of an optimal $(\mathcal{A}, \mathcal{B})$-covering. We then deduce from this bound an algorithm to build $(\mathcal{A}, \mathcal{B})$-coverings, and we analyze its worst-time complexity. Finally, we introduce an heuristic improvement and a theoretical bound that improve the prior results for sparse enough $\mathcal{B}$.

6.4.1 Our Construction

The next result generalizes the greedy coloring bound to $(\mathcal{A}, \mathcal{B})$-coverings. It gives a general bound of the size of an optimal $(\mathcal{A}, \mathcal{B})$-covering in terms of the degrees of $\mathcal{A}, \mathcal{B}$.

Theorem 6.13. *Let $\mathcal{A}, \mathcal{B} \subseteq \mathcal{P}(P)$ be families of subsets satisfying condition 6.1, and suppose that sets in $\mathcal{B}$ have size at most k. Then there exists an $(\mathcal{A}, \mathcal{B})$-covering $\mathcal{C}$ of size*

$$|\mathcal{C}| \leq k \deg(\mathcal{A}) \deg(\mathcal{B}) + 1$$

such that $\deg_v(\mathcal{C}) \leq \deg_v(\mathcal{B})$ for every $v \in P$.

Proof. We prove this by induction on $|\mathcal{B}|$. If $|\mathcal{B}| = 1$, then $\mathcal{C} = \mathcal{B}$ satisfies the lemma. Now let $s > 1$ be an integer and assume that the proposition holds for every pair $\mathcal{A}, \mathcal{B} \subseteq \mathcal{P}(P)$ of families of subsets satisfying $|\mathcal{B}| < s$ and the proposition hypotheses. Let $\mathcal{A}, \mathcal{B}' \subseteq \mathcal{P}(P)$ be a pair of families of subsets satisfying $|\mathcal{B}'| = s$ and the proposition hypotheses, and express $\mathcal{B}' = \mathcal{B} \cup \{B\}$ for some fixed $B \in \mathcal{B}'$ and $|\mathcal{B}| = s$. Then, by induction hypothesis, there exists an $(\mathcal{A}, \mathcal{B})$-covering $\mathcal{C}$ with $|\mathcal{C}| \leq k \deg(\mathcal{A}) \deg(\mathcal{B}) + 1$ and such that every $v \in P$ is contained in at most $\deg_v(\mathcal{B})$ elements of $\mathcal{C}$. We now build an $(\mathcal{A}, \mathcal{B}')$-covering $\mathcal{C}'$ from $\mathcal{C}$, in such a way that $|\mathcal{C}'| \leq k \deg(\mathcal{A}) \deg(\mathcal{B}') + 1$ and that every $v \in P$ is contained in at most $\deg_v(\mathcal{B}')$ elements of $\mathcal{C}$.

If B is contained in some $X \in \mathcal{C}$, then $\mathcal{C}' = \mathcal{C}$ satisfies the lemma. Otherwise, let

$$\mathcal{F}_B = \{X \in \mathcal{C} : \text{there exists } A \in \mathcal{A} \text{ with } A \subseteq X \cup B\}.$$

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Note that the condition $A \subseteq X \cup B$ is equivalent to $A \cap B \neq \emptyset$ and $A \setminus B \subseteq X$. Since there are at most $k \deg(\mathcal{A})$ elements $A \in \mathcal{A}$ with $A \cap B \neq \emptyset$, and since every set of the form $A \setminus B$ can be contained in at most $\deg(\mathcal{B})$ elements of $\mathcal{C}$ (because $\deg_v(\mathcal{C}) \leq \deg_v(\mathcal{B})$ for every $v \in P$ by hypothesis), we have that $|\mathcal{F}_B| \leq k \deg(\mathcal{A}) \deg(\mathcal{B})$.

Therefore, either there exists an element $X \in \mathcal{C} \setminus \mathcal{F}_B$, in which case we take

$$\mathcal{C}' = \{X \cup B\} \cup (\mathcal{C} \setminus \{X\})$$

or $\mathcal{C} = \mathcal{F}_B$, in which case $|\mathcal{C}| \leq k \deg(\mathcal{A}) \deg(\mathcal{B})$ and we let $\mathcal{C}' = \mathcal{C} \cup \{B\}$. □

Algorithm 6 is a greedy algorithm to compute an $(\mathcal{A}, \mathcal{B})$-covering that follows directly from the constructive proof of the previous lemma. This algorithm simply builds a ordered $(\mathcal{A}, \mathcal{B})$-covering $\mathcal{C}$ by iterating through $\mathcal{B}$. Every set B in $\mathcal{B}$ is merged with the first available element of $\mathcal{C}$, i.e., with the first element $X \in \mathcal{C}$ such that no $A \in \mathcal{A}$ is contained in $X \cup B$. If no such X exists, then B is added as a singleton in $\mathcal{C}$. Note that this algorithm is a generalization of the usual greedy coloring algorithm.

Algorithm 6: Construction

Input: $\mathcal{A} = \{A_1, \ldots, A_r\}$, $\mathcal{B} = \{B_1, \ldots, B_s\}$

```
1  Initialize C ← ∅
2  for i = 1, ..., s do
3  |  if B_i is not contained in any X ∈ C then
4  |  |  if there exists X ∈ C such that A ⊈ X ∪ B_i for every A ∈ A then
5  |  |  |  C ← {X ∪ B_i} ∪ (C \ {X})
6  |  |  else
7  |  |  |  C ← C ∪ {B_i}
8  |  |  end
9  |  end
10 end
```

Output: The $(\mathcal{A}, \mathcal{B})$-covering $\mathcal{C}$

To see the worst-time complexity of Algorithm 6, note that the first loop (line 2) is repeated $|\mathcal{B}|$ times. At step i, the first `if` statement (line 3) requires checking at most $i-1$ inclusions, and the second `if` statement (line 4) requires checking at most $(i-1)|\mathcal{A}|$ inclusions. Therefore, Algorithm 6 runs in time $O(|\mathcal{A}| \cdot |\mathcal{B}|^2)$.

6.4.2 An Heuristic Improvement

In order to motivate the heuristic procedure proposed later, we must first note that the output of Algorithm 6 depends strongly on the particular order in which elements of $\mathcal{B}$ are taken in the first loop. In particular, we see in the following proposition that there always exists an optimal ordering of the elements of $\mathcal{B}$. Of course, since the problem of finding an optimal $(\mathcal{A}, \mathcal{B})$-covering is NP-hard and an optimal ordering can be verified in polynomial time, finding an optimal ordering in our case is NP-complete.

Proposition 6.14. *Let $\mathcal{A}, \mathcal{B} \subseteq \mathcal{P}(P)$ be families of subsets satisfying condition 6.1. Then there exists an ordering of $\mathcal{B}$ such that Algorithm 6 outputs an optimal $(\mathcal{A}, \mathcal{B})$-covering.*

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Proof. Let $\mathcal{C} = \{X_1, \ldots, X_t\}$ be an optimal $(\mathcal{A}, \mathcal{B})$-covering. For every $j \in \{1, \ldots, t\}$, define S_j as the family of elements of $\mathcal{B}$ that are contained in X_j and that are not contained in any X_l for $l < j$,

$$S_j = \{B \in \mathcal{B} : B \subseteq X_j \text{ and } B \not\subseteq X_l \text{ for every } l < j\}.$$

We first prove that $\{S_j\}_{j=1}^t$ defines a partition of $\mathcal{B}$.

Indeed, $S_j \neq \emptyset$, because otherwise $\mathcal{C} \setminus \{X_j\}$ would be an $(\mathcal{A}, \mathcal{B})$-covering smaller than $\mathcal{C}$.

Also, $S_i \cap S_j = \emptyset$ for every i, j. Otherwise, if $B \in S_i \cap S_j$ with $i < j$, then $B \in S_i$ implies $B \subseteq X_i$, and $B \in S_j$ implies $B \not\subseteq X_i$, a contradiction.

Finally, since every $B \in \mathcal{B}$ is contained in some element of $\mathcal{C}$ by the definition of $(\mathcal{A}, \mathcal{B})$-covering, we can take $X_j \in \mathcal{C}$ with minimal index j among those that contain B. Then $B \in S_j$ by definition, and therefore $\mathcal{B} = \cup_{j=1}^t S_i$.

Now, define a new ordering of $\mathcal{B}$ by taking the sets in S_j sequentially. That is, if $S_j = \{B_{j,1}, \ldots, B_{j,k_j}\}$, define

$$\mathcal{B}' = \{B_{1,1}, \ldots, B_{1,k_1}, \ldots, B_{t,1}, \ldots, B_{t,k_t}\}.$$

Consider the behavior of Algorithm 6 on input $\mathcal{A}, \mathcal{B}'$. It is easy to see that, when the algorithm finishes processing the sets in S_j, the local variable $\mathcal{C}$ holds at most j elements. Therefore, since the covering $\mathcal{C}$ is optimal, Algorithm 6 outputs an optimal $(\mathcal{A}, \mathcal{B})$-covering. □

Following this result, we propose an heuristic procedure to build an ordering of $\mathcal{B}$, inspired in the Welsh-Powell algorithm [182]. This procedure can be deduced from the proof of the following proposition, which effectively reduces the upper bound given in Theorem 6.13 for sparse enough $\mathcal{B}$.

Proposition 6.15. *Assume the hypotheses of Theorem 6.13. Then there exists an $(\mathcal{A}, \mathcal{B})$-covering $\mathcal{C}$ of size*

$$|\mathcal{C}| \leq \max_i \min\{\deg_{B_i}(\mathcal{A}) \deg(\mathcal{B}) + 1, i\}$$

such that $\deg_v(\mathcal{C}) \leq \deg_v(\mathcal{B})$ for every $v \in P$.

Proof. First reorder $\mathcal{B}$ so that $\mathcal{B} = \{B_1, \ldots, B_s\}$ satisfies

$$\deg_{B_1}(\mathcal{A}) \geq \deg_{B_2}(\mathcal{A}) \geq \cdots \geq \deg_{B_s}(\mathcal{A}).$$

Now, consider the behavior of Algorithm 6 on input $\mathcal{A}$ and the reordered $\mathcal{B}$. At step i, algorithm 6 processes B_i. In this step, there can be at most $\deg_{B_i}(\mathcal{A}) \deg(\mathcal{B})$ sets $X \in \mathcal{C}$ such that B_i does not satisfy the condition in line 4 (that is, such that there exists $A \in \mathcal{A}$ with $A \subseteq X \cup B_i$). To see this, note that by definition at most $\deg_{B_i}(\mathcal{A})$ elements $A \in \mathcal{A}$ intersect B_i, and that each set of the form $A \backslash B_i$ can be contained in at most $\min\{\deg(\mathcal{B}), |\mathcal{C}|\} \leq \deg(\mathcal{B})$ elements of $\mathcal{C}$.

Now, at step i the number of elements $X \in \mathcal{C}$ checked in the condition of line 4 is at most $\min\{\deg_{B_i}(\mathcal{A}) \deg(\mathcal{B}), |\mathcal{C}|\}$. Since at step i the family $\mathcal{C}$ has at most $i - 1$ sets, at most

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

$\min\{\deg_{B_i}(\mathcal{A})\deg(\mathcal{B}), i-1\}$ elements of $\mathcal{C}$ are checked until either line 5 or 7 is executed, and line 7 can add an additional element to $\mathcal{C}$. Hence, by iterating through all elements of $\mathcal{B}$, the size of the final output can be at most $\max_i \min\{\deg_{B_i}(\mathcal{A})\deg(\mathcal{B}), i-1\}+1$. □

We now state our heuristic improvement of Algorithm 6, which follows directly from the previous proof.

Algorithm 7: Heuristic Improvement of Algorithm 6

Input: $\mathcal{A} = \{A_1, \ldots, A_r\}$, $\mathcal{B} = \{B_1, \ldots, B_s\}$
1 **for** $B \in \mathcal{B}$ **do**
2 | Compute $\deg_B(\mathcal{A}) = |\{A \in \mathcal{A} : A \cap B \neq \emptyset\}|$
3 **end**
4 Sort $\mathcal{B}$ so that $\mathcal{B} = \{B'_1, \ldots, B'_s\}$ satisfies $\deg_{B'_1}(\mathcal{A}) \geq \deg_{B'_2}(\mathcal{A}) \geq \cdots \geq \deg_{B'_s}(\mathcal{A})$
Output: The output of Algorithm 6 on input $\mathcal{A}, \mathcal{B}$

To see the worst-time complexity of Algorithm 7, note that the computation of each quantity $\deg_B(\mathcal{A})$ requires $O(|\mathcal{A}|)$ time. Adding in the sorting time, we see that our heuristic takes $O(|\mathcal{B}|\cdot(|\mathcal{A}|+\log(|\mathcal{B}|)))$ time. In turn, adding this to the cost of Algorithm 6 does not alter the total $O(|\mathcal{A}|\cdot|\mathcal{B}|^2)$ worst-time complexity.

Remark 6.16. In fact, the previous proof indicates a slightly better bound. For an integer α define the function f_α by

$$f_\alpha(\beta) = \begin{cases} \beta & \text{if } \alpha < \beta \\ \beta + 1 & \text{otherwise.} \end{cases}$$

Then the bound on the previous proposition can be replaced by

$$|\mathcal{C}| \leq f_{\deg_{B_s}(\mathcal{A})\deg(\mathcal{B})}(f_{\deg_{B_{s-1}}(\mathcal{A})\deg(\mathcal{B})}(\cdots(f_{\deg_{B_2}(\mathcal{A})\deg(\mathcal{B})}(1))\cdots)) + 1.$$

6.5 Experimental Results

This section details the experimental results obtained by implementing the proposed methods in the Sage Mathematical Software System [196], version 7.4. First, a comparison between Algorithm 7 and the Gröbner basis method on a practical setting is shown. Then, a performance analysis of Algorithms 6 and 7 is carried out over random graphs. The reported experiments have been conducted on an AMD Ryzen 7 1700X Eight-core 3.4 GHz processor, with 32 GB of RAM, in Sage [196] and under Ubuntu 4.10.0-37. All experiments have been carried out without parallelization. All CPU running times have been collected using the function `cputime(subprocesses=True)` in Sage.

6.5.1 Medical Data Example

Medical data applications tend to be extremely storage and functionality-demanding, and thus it is often unfeasible for the data holder to locally store and manage the data. Therefore, medical data provides a good candidate for a privacy-preserving data splitting use case.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Table 6.1 depicts several possible features that can be found in medical data. Since the features *patient ID* and *address* completely identify the patient, they need to be stored in an encrypted form, and therefore they are not taken into account in the associated data splitting problem. A different numerical identifier is assigned to every other feature for presentation purposes.

#	Hospital Folder features
-	patient ID
-	address
0	ZIP code
1	birth date
2	gender
3	ethnicity

#	Hospital Folder features
4	weight
5	diagnosis
6	procedure
7	medication
8	charges
9	hospital ID

TABLE 6.1: Example of patient Healthcare features.

Observe that other combinations of attributes can also be sensitive. An example of such combination can be $\{0, 2, 3\}$ as it is shown in [197], where a 1990 federal census reports that in Dekalb, Illinois there were only two black women who resided in that town. We can also consider $\{0, 1, 3\}$, $\{0, 1, 4\}$ and $\{1, 2, 3\}$ sensitive for obvious reasons. Moreover, some attributes may need to be stored in the same fragment, for instance to perform statistical analysis computations. Possible combinations are: $\{1, 2, 5\}$, $\{1, 3, 5\}$ and $\{0, 2, 5\}$.

The Gröbner basis method (implemented as `buchberger2()` in Sage) and Algorithm 7 can be used to find an $(\mathcal{A}, \mathcal{B})$-covering that solves the data splitting problem, where $\mathcal{A} = \{\{0, 2, 3\}, \{0, 1, 2\}, \{0, 1, 4\}, \{1, 2, 3\}\}$ and $\mathcal{B} = \{\{1, 2, 5\}, \{1, 3, 5\}, \{0, 2, 5\}, \{4\}\}$.

			Gröbner basis method			Algorithm 7		
# v.	$\mathcal{A}$	$\mathcal{B}$	# sol.	$\lvert\mathcal{C}\rvert$	time	opt.	$\lvert\mathcal{C}\rvert$	time
6	$\{023, 012, 014, 123\}$	$\{125, 135, 025, 4\}$	15	3	6.68s	Yes	3	3.84ms
6	$\{023, 012, 014\}$	$\{125, 135, 025, 4\}$	3	2	1.12ms	Yes	2	1.19ms
8	$\{045, 123, 89\}$	$\{124, 458, 09, 238\}$	1	2	316ms	No	3	1.41ms
9	$\{13, 168, 34, 79, 036\}$	$\{023, 012, 36, 46, 78, 07, 9\}$	204	3	16h 45min	Yes	3	4.88ms
10	$\{02, 168, 34, 79, 03\}$	$\{01, 128, 35, 46, 78, 04, 23, 9\}$	2	2	1.87s	Yes	2	2.19ms

"# v.": number of vertices of the selected hypergraph,
"# sol.": number of optimal coverings,
$|\mathcal{C}|$: size of the $(\mathcal{A}, \mathcal{B})$-coverings computed by the respective method,
"time": the time needed by the related method to compute the solution,
"opt.": whether or not the solution of Algorithm 7 is optimal,
We use a compact notation for sets, i.e. $023 = \{0, 2, 3\}$.

TABLE 6.2: Comparison between Gröbner basis method and Algorithm 7 on several hypergraphs.

The first column of Table 6.2 shows the results of applying the Gröbner basis method and Algorithm 7 to the medical data set. Algorithm 7 has been chosen for the tests above instead of Algorithm 6 due to efficiency reasons. Both the Gröbner basis method and Algorithm 7 provide optimal solutions in the considered case but with a considerable time difference. Two of the optimal solutions computed by the Gröbner basis method are depicted in Figure 6.1.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Table 6.2 also depicts the results of the Gröbner basis method and Algorithm 7 to several other splitting problems, all referred to the medical data set of Table 6.1. The time needed to obtain a solution is strictly related to the degree of $\mathcal{A}$ and $\mathcal{B}$. Other parameters which affect the running time are the number of vertices and the size of the optimal covering (see Section 6.3 for more details). However, while having the same number of vertices, the needed time may vary greatly. Observe that Algorithm 7 does not always compute an optimal solution.

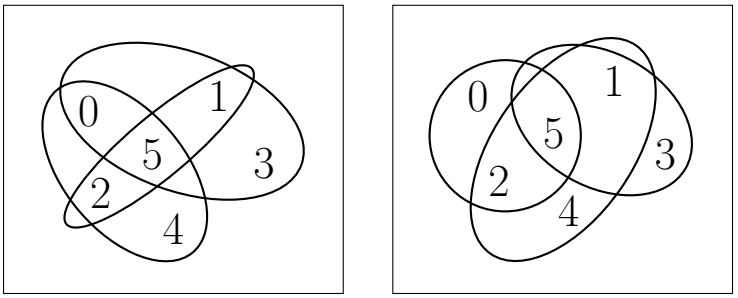


FIGURE 6.1: Two of the optimal solutions computed by Gröbner basis method of the medical data problem. The solution found by Algorithm 7 is depicted in the right chart. Vertices in the same set belong to the same fragment.

6.5.2 Performance Analysis over Random Graphs

We now give some performance measures to analyze the results presented in Section 6.4. In this performance analysis we restrict to the graph case, and thus we take $\mathcal{A} \subseteq \binom{P}{2}$ and $\mathcal{B} \subseteq \binom{P}{2} \cup \binom{P}{1}$ to be disjoint families of subsets. We classify the test cases according to two parameters: the number n of vertices and the sum of densities $\rho = \rho_{\mathcal{A}} + \rho_{\mathcal{B}}$ of $\mathcal{A}$ and $\mathcal{B}$. For each test case, we randomly generate graphs by choosing every single edge of the complete graph K_n with probability ρ, and we then throw a uniform random coin for each chosen edge to determine if it belongs either to $\mathcal{A}$ or to $\mathcal{B}$. Next, we add the necessary singletons to $\mathcal{B}$ so that $\cup_{B\in\mathcal{B}} B = P$. Finally, we randomly shuffle $\mathcal{A}$ and $\mathcal{B}$ and we apply the algorithm to test. Hence, in the considered experiments both $\mathcal{A}$ and $\mathcal{B}$ are generated with density $\rho_{\mathcal{A}} = \rho_{\mathcal{B}} = \rho/2$.

In Table 6.3 we analyze the time performance of Algorithm 6 and Algorithm 7. For each of the considered n and ρ, the reported CPU running times have been averaged over 10^3 independent random experiments.

	Algorithm 6				Algorithm 7			
n	$\rho = 0.1$	$\rho = 0.4$	$\rho = 0.7$	$\rho = 1.0$	$\rho = 0.1$	$\rho = 0.4$	$\rho = 0.7$	$\rho = 1.0$
10	0.003	0.010	0.029	0.066	0.005	0.020	0.057	0.125
20	0.018	0.181	0.740	1.989	0.036	0.347	1.255	3.021
30	0.065	1.150	5.134	14.80	0.136	2.023	7.769	19.88
40	0.186	4.415	21.10	61.85	0.384	7.116	29.03	78.39
50	0.442	12.69	63.28	190.6	0.891	19.29	81.76	227.8
60	0.941	30.45	155.8	478.7	1.864	43.95	193.2	552.9
70	1.803	63.69	338.6	1054	3.452	88.57	403.8	1177

TABLE 6.3: Time performance analysis (in seconds).

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Observe that average running times increase both in the number of attributes and the density, and range between milliseconds and 20 minutes for the considered parameters.

Next, in Table 6.4 we compile evidence on the size of the result output by Algorithm 7 over the size of the result output by Algorithm 6. For every considered n and ρ, we randomly instantiate 10^5 different $\mathcal{A}$ and $\mathcal{B}$ as stated above, and for each of them we compute the sizes s_{alg} and s_{heur} of the $(\mathcal{A}, \mathcal{B})$-coverings given by Algorithm 6 and by Algorithm 7, respectively. Then, we compute the decrease as the percentage $(100(s_{\text{alg}} - s_{\text{heur}})/s_{\text{alg}})\%$ in size offered by the heuristic. The reported percentage decreases have been averaged over at least 10^5 independent random experiments.

n	$\rho = 0.1$	$\rho = 0.3$	$\rho = 0.5$	$\rho = 0.7$	$\rho = 0.9$	$\rho = 1.0$
5	0.01233	0.2569	0.5101	0.3551	0.09567	0
6	0.07317	1.006	1.234	0.7517	0.3193	0.08730
7	0.2227	1.991	1.876	1.193	0.6438	0.3433
8	0.5131	3.000	2.486	1.709	1.075	0.7280
9	0.9214	3.675	3.007	2.269	1.600	1.184
10	1.609	4.275	3.563	2.799	2.087	1.634

TABLE 6.4: Average percent size reduction given by Algorithm 7 from the size given by Algorithm 6.

Following the same procedure, in Table 6.5 we compile evidence on the size increase of the covering given by Algorithm 7 in relation to the size of an optimal covering. The reported percentage decreases have been averaged over at least 10^4 independent random experiments.

n	$\rho = 0.1$	$\rho = 0.3$	$\rho = 0.5$	$\rho = 0.7$	$\rho = 0.9$	$\rho = 1.0$
5	0.00003333	0.0002666	0.003666	0.001333	0.00001333	0
6	0.001333	0.0250	0.1045	0.07500	0.03203	0.02680
7	0.009333	0.3333	0.3250	0.1683	0.1602	0.1548
8	0.06667	0.5167	0.4083	0.4844	0.3111	0.1940
9	0.1333	0.8417	1.041	0.8906	0.8411	0.3667
10	0.3333	1.493	1.601	1.396	1.210	0.9015

TABLE 6.5: Average percent size increase given by Algorithm 7 with respect to the optimal size.

In Table 6.4, we observe that our heuristic algorithm 7 generally improves the greedy algorithm 6 for a small number of attributes, and that this improvement grows in the number of attributes and is larger for medium densities. In addition, in Table 6.5 we confirm that our heuristic algorithm generally provides near-to-optimal sized decompositions for a small number of attributes, and that much better results are achieved for small densities. In the case $n = 5$ and $\rho = 1$, our algorithms always provide optimal coverings.

6.6 Conclusion

Recent data splitting research focuses in preserving the privacy of a sensitive data set by decomposing it into a small number of fragments. In this context, data is split into a

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

small number of fragments, frequently two or three. Since this does not usually suffice to ensure privacy, existing solutions build cryptographic techniques on top of data splitting. However, up to this point no research has engaged with the data splitting problem in a setting where no other privacy-preserving techniques are to be used.

In our work, we tackle the problem of addressing privacy concerns by finding a decomposition into fragments of a given data set. We also take into account processing constraints, which may impose some sets of data attributes to be stored together in the same fragment. We first consider the problem of finding the optimal number of fragments needed to satisfy privacy and processing constraints, and we further remove the optimality condition to provide better efficiency.

Firstly, we present a formulation of the stated problem and a concrete approach to solve it. The data splitting problem is presented as a purely combinatorial problem, by specifying two families of subsets $\mathcal{A}$ and $\mathcal{B}$. The first family $\mathcal{A}$ represents privacy constraints, and specifies sets of attributes that must not be stored together for privacy reasons. The second family $\mathcal{B}$ represents processing constraints, and specifies sets of attributes that must be stored together in the same fragment in order to speed up processing. In this setting, we introduce the notion of $(\mathcal{A}, \mathcal{B})$-covering, and we show that $(\mathcal{A}, \mathcal{B})$-coverings directly translate to solutions of the privacy-preserving data splitting problem.

Once the combinatorial problem of finding $(\mathcal{A}, \mathcal{B})$-covering is stated, we show that it can be solved by using purely algebraic techniques through its equivalence to a hypergraph-coloring problem. We exhibit an algebraic formulation of the data splitting problem, which translates privacy and processing constraints to a system of simultaneous equations. Through the use of Gröbner bases, this formulation allows the computation of optimally-sized data decompositions.

Since finding an optimal covering is an NP-hard problem, obtaining optimal solutions is often unfeasible in practice. We hence present a greedy algorithm that sacrifices optimality for efficiency, achieving a polynomial running time in the size of the considered problem. We further present an heuristic improvement of this greedy algorithm, that provides smaller decompositions when the family of constraints is sparse enough.

A performance analysis is carried out to evaluate all of the presented solutions. First, we analyze the execution time of our first algebraic approach in the context of a medical data set. Next, we report the execution times of our greedy and heuristic algorithms over random graphs, and we estimate the size overhead incurred by both algorithms with respect to the optimal size for a small number of attributes. The experimental results show that our greedy algorithm requires milliseconds to find a solution, whereas computing an optimal solution may require hours depending on the problem at hand.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

Bibliography

[1] CLARUS Consortium. CLARUS: a framework for user centered privacy and security in the cloud. Horizon 2020 project H2020-ICT-2014-1-644024, 2016. URL http://www.clarussecure.eu/.

[2] Vigilante.pw: the breached database directory. https://vigilante.pw/, 2018. Accessed: 04/01/2018.

[3] List of data breaches, Wikipedia. https://en.wikipedia.org/wiki/List_of_data_breaches/, 2018. Accessed: 04/01/2018.

[4] The 2016 IDG cloud computing survey. https://www.idg.com/tools-for-marketers/2016-idg-enterprise-cloud-computing-survey/, 2016. Accessed: 04/01/2018.

[5] RightScale state of the cloud report. https://www.rightscale.com/lp/state-of-the-cloud, 2017. Accessed: 04/01/2018.

[6] R.A. Popa, C.M.S. Redfield, N. Zeldovich, and H. Balakrishnan. CryptDB: protecting confidentiality with encrypted query processing. In *SOSP*, pages 85–100, 2011.

[7] S. Tu, M.F. Kaashoek, S. Madden, and N. Zeldovich. Processing analytical queries over encrypted data. In *Proceedings of the 39th international conference on Very Large Data Bases*, PVLDB'13, pages 289–300. VLDB Endowment, 2013.

[8] CipherCloud. http://www.ciphercloud.com, 2018. Accessed: 04/01/2018.

[9] V.S.K. Reddy and C. Rajendra. EKST: Eficient keyword searching technique for encrypting data in CipherCloud. *International Journal of Advanced Research in Computer Engineering and Technology*, 1(4):–552, 2012.

[10] TrendMicro. SecureCloud. http://www.trendmicro.com/us/enterprise/cloud-solutions/secure-cloud/, 2018. Accessed: 04/01/2018.

[11] PerspecSys. http://perspecsys.com, 2018. Accessed: 04/01/2018.

[12] BoxCryptor. https://www.boxcryptor.com/en, 2018. Accessed: 04/01/2018.

[13] CloudFogger. https://www.cloudfogger.com/en/, 2018. Accessed: 04/01/2018.

[14] CloudSpaces. http://cloudspaces.eu/, 2018. Accessed: 04/01/2018.

[15] TRESSCA. http://www.trescca.eu/, 2018. Accessed: 04/01/2018.

[16] ESCUDO-CLOUD. http://www.escudocloud.eu/, 2018. Accessed: 04/01/2018.

[17] MUSA. `http://www.musa-project.eu/`, 2018. Accessed: 04/01/2018.

[18] J.-P. Chilès and P. Delfiner. Multivariate methods. *Geostatistics: Modeling Spatial Uncertainty, Second Edition*, pages 299–385, 1999.

[19] N. Cressie. Statistics for spatial data. *Terra Nova*, 4(5):613–617, 1992.

[20] D.G. Krige. A statistical approach to some basic mine valuation problems on the Witwatersrand. *Journal of the Southern African Institute of Mining and Metallurgy*, 52(6):119–139, 1951.

[21] H. Wackernagel. *Multivariate geostatistics: an introduction with applications*. Springer Science & Business Media, 2013.

[22] BRGM: geoscience for a sustainable earth. `http://www.brgm.eu/`, 2018. Accessed: 04/01/2018.

[23] GEUS: geological survey of denmark and greenland. `http://www.geus.dk/geuspage-uk.htm`, 2018. Accessed: 04/01/2018.

[24] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM Conference on Computer and Communications Security*, CCS'06, pages 89–98. ACM, 2006.

[25] 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*, SP'00, pages 44–. IEEE Computer Society, 2000.

[26] D. Cash, P. Grubbs, J. Perry, and T. Ristenpart. Leakage-abuse attacks against searchable encryption. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, pages 668–679. ACM, 2015. ISBN 978-1-4503-3832-5.

[27] M.-S. Lacharité and K. G. Paterson. A note on the optimality of frequency analysis vs. ℓ_p-optimization. *IACR Cryptology ePrint Archive*, 2015:1158, 2015. URL `http://eprint.iacr.org/2015/1158`.

[28] M. S. Islam, M. Kuzu, and M. Kantarcioglu. Access pattern disclosure on searchable encryption: Ramification, attack and mitigation. In *NDSS*. The Internet Society, 2012.

[29] G. Kellaris, G. Kollios, K. Nissim, and A. O'Neill. Generic attacks on secure outsourced databases. In *ACM Conference on Computer and Communications Security*, pages 1329–1340. ACM, 2016.

[30] M.-S. Lacharité, B. Minaud, and K. G. Paterson. Improved reconstruction attacks on encrypted data using range query leakage. In *IEEE Symposium on Security and Privacy (SP)*, pages 297–314, 2018.

[31] D. Pouliot and C. V. Wright. The shadow nemesis: Inference attacks on efficiently deployable, efficiently searchable encryption. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1341–1352, 2016.

[32] Y. Zhang, J. Katz, and C. Papamanthou. All your queries are belong to us: The power of file-injection attacks on searchable encryption. In *USENIX Security Symposium*, pages 707–720. USENIX Association, 2016.

[33] O. Goldreich. Towards a theory of software protection and simulation by oblivious RAMs. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, STOC '87, pages 182–194. ACM, 1987.

[34] O. Goldreich and R. Ostrovsky. Software protection and simulation on oblivious RAMs. *Journal of the ACM*, 43(3):431–473, 1987.

[35] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. Private information retrieval. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, FOCS '95, pages 41–. IEEE Computer Society, 1995.

[36] H. Lipmaa. First CPIR protocol with data-dependent computation. In *Information, Security and Cryptology - ICISC, 12th International Conference*, pages 193–210, 2009.

[37] C.R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky. Searchable symmetric encryption: improved definitions and efficient constructions. In *Proceedings of the 13th ACM conference on Computer and communications security*, CCS 06, ACM, pages 79–88, 2006.

[38] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search. In *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, volume 3027 of *EUROCRYPT'04*, pages 506–522. Springer, 2004.

[39] P. Xu, X. Gao, W. Wang, W. Susilo, Q. Wu, and H. Jin. Dynamic searchable public-key ciphertexts with fast performance and practical security. Cryptology ePrint Archive, Report 2017/741, 2017. `https://eprint.iacr.org/2017/741`.

[40] J.W. Byun, D.H. Lee, and J. Lim. Efficient conjunctive keyword search on encrypted data storage system. In *Proceedings of the Third European Conference on Public Key Infrastructure: Theory and Practice*, EuroPKI'06, pages 184–196. Springer-Verlag, 2006.

[41] B. Zhang and F. Zhang. An efficient public key encryption with conjunctive-subset keywords search. *Journal of Network and Computer Applications*, 34(1):262–267, 2011.

[42] Y.H. Hwang and P.J. Lee. Public key encryption with conjunctive keyword search and its extension to a multi-user system. In *Proceedings of The first International Conference on Pairing-based Cryptography*, Pairing'07, pages 2–22. Springer Berlin Heidelberg, 2007.

[43] J. Baek, R. Safavi-Naini, and W. Susilo. Public key encryption with keyword search revisited. In *Proceedings of Computational Science and Its Applications*, ICCSA 2008, Part I, pages 1249–1259. Springer, 2008.

[44] J.W. Byun and D.H. Lee. On a security model of conjunctive keyword search over encrypted relational database. *Journal of Systems and Software*, 84(8):1364–1372, 2011.

[45] Y. Chen, J. Zhang, D. Lin, and Z. Zhang. Generic constructions of integrated PKE and PEKS. *Designs, Codes and Cryptography*, 78(2):493–526, 2016.

[46] C. Liu, L. Zhu, M. Wang, and Y.-A. Tan. Search pattern leakage in searchable encryption: Attacks and new construction. *Information Sciences*, 265:176–188, 2014.

[47] H.S. Rhee, J.H. Park, W. Susilo, and D.H. Lee. Trapdoor security in a searchable public-key encryption scheme with a designated tester. *Journal of Systems and Software*, 83(5):763–771, 2010.

[48] T. Wang, M.H. Au, and W. Wu. An efficient secure channel free searchable encryption scheme with multiple keywords. In *Proceedings of the 10th International Conference on Network and System Security*, NSS'16, pages 251–265. Springer International Publishing, 2016.

[49] R. Cramer, I. Damgård, and U. Maurer. General secure multi-party computation from any linear secret-sharing scheme. In *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, EUROCRYPT'00, pages 316–334. Springer-Verlag, 2000.

[50] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[51] G.R. Blakley. Safeguarding cryptographic keys. In *Proceedings of the AFIPS 1979 National Computer Conference*, volume 48, pages 313–317. AFIPS Press, 1979.

[52] M. Ito, A. Saito, and T. Nishizeki. Secret sharing scheme realizing general access structure. *Electronics and Communications in Japan, Part III*, 72(9):56–64, 1989.

[53] J. Benaloh and J. Leichter. Generalized secret sharing and monotone functions. In *Proceedings on Advances in Cryptology*, CRYPTO'88, pages 27–35. Springer-Verlag New York, Inc., 1990.

[54] G. Aggarwal, M. Bawa, P. Ganesan, H. Garcia-Molina, K. Kenthapadi, R. Motwani, U. Srivastava, D. Thomas, and Y. Xu. Two can keep a secret: A distributed architecture for secure database services. In *The Second Biennial Conference on Innovative Data Systems Research (CIDR 2005)*, 2005.

[55] V. Ganapathy, D. Thomas, T. Feder, H. Garcia-Molina, and R. Motwani. Distributing data for secure database services. In *Proceedings of the 4th International Workshop on Privacy and Anonymity in the Information Society*, page 8. ACM, 2011.

[56] V. Ciriani, S. De C. Di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati. Selective data outsourcing for enforcing privacy. *Journal of Computer Security*, 19(3):531–566, 2011.

[57] V. Ciriani, S. De C. Di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati. Combining fragmentation and encryption to protect privacy in data storage. *ACM Transactions on Information and System Security (TISSEC)*, 13 (3):22:1–22:33, 2010.

[58] V. Ciriani, S. De C. Di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati. Fragmentation and encryption to enforce privacy in data storage. In *European Symposium on Research in Computer Security*, pages 171–186. Springer, 2007.

[59] H. Hacigümüş, B. Iyer, C. Li, and S. Mehrotra. Executing sql over encrypted data in the database-service-provider model. In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, SIGMOD '02, pages 216–227, New York, NY, USA, 2002. ACM. ISBN 1-58113-497-5.

[60] B. Hore, S. Mehrotra, M. Canim, and M. Kantarcioglu. Secure multidimensional range queries over outsourced data. *The VLDB Journal*, 21(3):333–358, 2012.

[61] B. Hore, S. Mehrotra, and G. Tsudik. A privacy-preserving index for range queries. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30*, VLDB '04, pages 720–731. VLDB Endowment, 2004. ISBN 0-12-088469-0.

[62] R. Poddar, T. Boelter, and R.A. Popa. Arx: A strongly encrypted database system. Cryptology ePrint Archive, Report 2016/591, 2016. `https://eprint.iacr.org/2016/591`.

[63] A. Boldyreva, N. Chenette, Y. Lee, and A. O'Neill. Order-preserving symmetric encryption. In *Proceedings of the 28th Annual International Conference on Advances in Cryptology: The Theory and Applications of Cryptographic Techniques*, EUROCRYPT '09, pages 224–241, Berlin, Heidelberg, 2009. Springer-Verlag.

[64] F. Hahn and F. Kerschbaum. Poly-logarithmic range queries on encrypted data with small leakage. In *Proceedings of the 2016 ACM on Cloud Computing Security Workshop*, CCSW '16, pages 23–34, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4572-9.

[65] D. Boneh and B. Waters. Conjunctive, subset, and range queries on encrypted data. In *Proceedings of the 4th Conference on Theory of Cryptography*, TCC'07, pages 535–554. Springer-Verlag, 2007.

[66] E. Shi, J. Bethencourt, T.-H. Chan, D. Song, and A. Perrig. Multidimensional range query over encrypted data. In *Proceedings of the IEEE Symposium on Security and Privacy*, SP'07, pages 350–364, 2007.

[67] R. Li, A.X. Liu, A.L. Wang, and B. Bruhadeshwar. Fast range query processing with strong privacy protection for cloud computing. *Proc. VLDB Endow.*, 7(14): 1953–1964, 2014.

[68] S. Faber, S. Jarecki, H. Krawczyk, Q. Nguyen, M.-C. Rosu, and M. Steiner. Rich queries on encrypted data: Beyond exact matches. In *Proceedings of the 20th European Symposium on Research in Computer Security*, ESORICS'15, Part II, pages 123–145, 2015.

[69] O. Farràs and J. Ribes-González. Searchable encryption for geo-referenced data. In *15th IFIP Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net'16)*, pages 1–8, 2016.

[70] J. Li and E.R. Omiecinski. Efficiency and security trade-off in supporting range queries on encrypted databases. In *Proceedings of the 19th Annual IFIP WG 11.3 Working Conference on Data and Applications Security*, DBSec'05, pages 69–83. Springer-Verlag, 2005.

[71] I. Demertzis, S. Papadopoulos, O. Papapetrou, A. Deligiannakis, and M. Garofalakis. Practical private range search revisited. In *Proceedings of the 2016 International Conference on Management of Data*, SIGMOD '16, pages 185–198, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-3531-7.

[72] F. Kerschbaum and A. Tueno. An efficiently searchable encrypted data structure for range queries. *CoRR*, abs/1709.09314, 2017. URL `http://arxiv.org/abs/1709.09314`.

[73] D.J. Park, J. Cha, and P.J. Lee. Searchable keyword-based encryption, 2005.

[74] D.J. Park, K. Kim, and P.J. Lee. Public key encryption with conjunctive field keyword search. In *Proceedings of the 5th International Workshop on Information Security Applications*, volume 3325 of *WISA'04*, pages 73–86. Springer, 2004.

[75] InGeoCloudS: inspired geo-data cloud services. `https://www.ingeoclouds.eu/`, 2016. Accessed: 11/12/2016.

[76] B. Tugrul and H. Polat. Estimating Kriging-based predictions with privacy. *International Journal of Innovative Computing, Information and Control*, 2013.

[77] B. Tugrul and H. Polat. Privacy-preserving Kriging interpolation on partitioned data. *Knowledge-Based Systems*, 62:38–46, 2014.

[78] M. Karchmer and A. Wigderson. On span programs. In *Proceedings of the 8th IEEE Structure in Complexity Theory*, SCT'93, pages 102–111. IEEE Computer Society Press, 1993.

[79] E.F. Brickell. Some ideal secret sharing schemes. In *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, EUROCRYPT'89, pages 468–475. Springer Berlin Heidelberg, 1990.

[80] O. Farràs, J. Martí-Farré, and C. Padró. Ideal multipartite secret sharing schemes. *Journal of Cryptology*, 25(3):434–463, 2012.

[81] O. Farràs, J.R. Metcalf-Burton, C. Padró, and L. Vázquez. On the optimization of bipartite secret sharing schemes. *Designs, Codes and Cryptography*, 63(2):255–271, 2012.

[82] C. Padró, L. Vázquez, and A. Yang. Finding lower bounds on the complexity of secret sharing schemes by linear programming. *Discrete Applied Mathematics*, 161 (7-8):1072–1084, 2013.

[83] L. Csirmaz. Secret sharing on the d-dimensional cube. *Designs, Codes and Cryptography*, 74(3):719–729, 2015.

[84] O. Farràs, J. Ribes-González, and S. Ricci. Local bounds for the optimal information ratio of secret sharing schemes. *Designs, Codes and Cryptography*, 2018. URL `https://doi.org/10.1007/s10623-018-0529-7`.

UNIVERSITAT ROVIRA I VIRGILI
CRYPTOGRAPHIC TECHNIQUES FOR SECURING DATA IN THE CLOUD
Jordi Ribes González

[85] J. Alderman, B.R. Curtis, O. Farràs, K.M. Martin, and J. Ribes-González. Private outsourced Kriging interpolation. In *Financial Cryptography and Data Security (FC'17) International Workshops: 4th Workshop on Encrypted Computing and Applied Homomorphic Cryptography (WAHC'17), Revised Selected Papers*, pages 75–90. Springer International Publishing, 2017.

[86] O. Farràs, J. Ribes-González, and S. Ricci. Privacy-preserving data splitting: A combinatorial approach. *Submitted, preprint at arXiv:1801.05974*, 2018.

[87] O. Farràs and J. Ribes-González. Provably secure public key encryption with conjunctive and subset keyword search. *Submitted to the International Journal on Information Security*, 2017.

[88] J. Katz and Y. Lindell. *Introduction to Modern Cryptography, Second Edition*. Chapman & Hall/CRC, 2nd edition, 2014. ISBN 1466570261, 9781466570269.

[89] A.J. Menezes, S.A. Vanstone, and P.C. Van Oorschot. *Handbook of Applied Cryptography*. CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 1996. ISBN 0849385237.

[90] J. Daemen and V. Rijmen. *The Design of Rijndael*. Springer-Verlag New York, Inc., 2002. ISBN 3540425802.

[91] P.R. Karn, W.A. Simpson, and P.E. Metzger. The ESP Triple DES Transform. RFC 1851, 1995. URL `https://rfc-editor.org/rfc/rfc1851.txt`.

[92] B. Schneier. Description of a new variable-length key, 64-bit block cipher (blowfish). In *Fast Software Encryption, Cambridge Security Workshop*, pages 191–204. Springer-Verlag, 1994. ISBN 3-540-58108-1.

[93] D. Cash, S. Jarecki, C. Jutla, H. Krawczyk, M. Rosu, and M. Steiner. Highly-scalable searchable symmetric encryption with support for boolean queries, 2013.

[94] Y. Lindell. How to simulate it-a tutorial on the simulation proof technique. Technical report, IACR Cryptology ePrint Archive, 2016: 46, 2016.

[95] S. Goldwasser and S. Micali. Probabilistic encryption; how to play mental poker keeping secret all partial information. In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, STOC '82, pages 365–377. ACM, 1982.

[96] V.S. Miller. Short programs for functions on curves. In *IBM Thomas J. Watson Research Center*, 1986.

[97] S.D. Galbraith, K.G. Paterson, and N.P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.

[98] P.S.L.M. Barreto, H.Y. Kim, B. Lynn, and M. Scott. Efficient algorithms for pairing-based cryptosystems. In *Proceedings of Advances in Cryptology*, volume 2442 of *CRYPTO'09*, pages 354–368. Springer, 2002.

[99] M. Scott and P.S.L.M. Barreto. Compressed pairings. In *Proceedings of Advances in Cryptology*, CRYPTO'04, pages 140–156. Springer Berlin Heidelberg, 2004.

[100] ECRYPT II. D.MAYA.6: final report on main computational assumptions in cryptography. ECRYPT II Project co-funded by the European Commission within the 7th Framework Programme, ICT-2007-216676, 2013. URL http://cordis.europa.eu/docs/projects/cnect/6/216676/080/deliverables/001-DMAYA6.pdf.

[101] D. Boneh. The decision diffie-hellman problem. In *Proceedings of the Third International Symposium on Algorithmic Number Theory*, ANTS-III, pages 48–63. Springer-Verlag, 1998.

[102] D. Boneh and X. Boyen. Efficient selective identity-based encryption without random oracles. *Journal of Cryptology*, 24(4):659–693, 2011. early version in Eurocrypt 2004.

[103] A. Joux. A one round protocol for tripartite diffie-hellman. In *Proceedings of the 4th International Symposium on Algorithmic Number Theory*, ANTS-IV'00, pages 385–394. Springer-Verlag, 2000.

[104] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security*, CCS'93, pages 62–73, 1993.

[105] S. Frankel and S.G. Kelly. Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec. RFC 4868, 2007. URL https://rfc-editor.org/rfc/rfc4868.txt.

[106] T. Iwata, J. Song, J. Lee, and R. Poovendran. The AES-CMAC Algorithm. RFC 4493, 2006. URL https://rfc-editor.org/rfc/rfc4493.txt.

[107] M. Tibouchi. A note on hashing to BN curves. In *Proceedings of the 29th Japanese Symposium on Cryptography and Information Security*, SCIS'12, IEICE, 2012.

[108] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, EUROCRYPT'99, pages 223–238. Springer, 1999.

[109] C. Bösch, P. Hartel, W. Jonker, and A. Peter. A survey of provably secure searchable encryption. *ACM Computing Surveys*, 47(2):18:1–18:51, 2014.

[110] S. Kamara, C. Papamanthou, and T. Roeder. Dynamic searchable symmetric encryption. In *Proceedings of the 2012 ACM conference on Computer and Communications Security*, pages 965–976. ACM, 2012.

[111] Q. Zheng, S. Xu, and G. Ateniese. VABKS: verifiable attribute-based keyword search over outsourced encrypted data. In *Infocom, 2014 proceedings IEEE*, pages 522–530. IEEE, 2014.

[112] E.-J. Goh. Secure indexes, 2003. URL http://eprint.iacr.org/.

[113] Y.-C. Chang and M. Mitzenmacher. Privacy preserving keyword searches on remote encrypted data. In *Proceedings of the Third International Conference on Applied Cryptography and Network Security*, ACNS'05, pages 442–455. Springer-Verlag, 2005.

[114] M. Chase and S. Kamara. Structured encryption and controlled disclosure. *IACR Cryptology ePrint Archive*, 2011. URL http://eprint.iacr.org/2011/010.

[115] M. Fredman, J. Komlós, and E. Szemerédi. Storing a sparse table with $o(1)$ worst case access time. *Journal of the ACM (JACM)*, 31(3):538–544, 1984.

[116] L. Ballard, S. Kamara, and F. Monrose. Achieving efficient conjunctive keyword searches over encrypted data. In *Information and Communications Security: 7th International Conference*, ICICS'05, pages 414–426. Springer, 2005.

[117] I.R. Jeong, J.O. Kwon, D. Hong, and D.H. Lee. Constructing PEKS schemes secure against keyword guessing attacks is possible? *Computer Communications*, 32(2):394–396, 2009.

[118] H.S. Rhee, J.H. Park, W. Susilo, and D.H. Lee. Improved searchable public key encryption with designated tester. In *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, ASIACCS'09, pages 376–379. ACM, 2009.

[119] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi. Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. In *Proceedings of Advances in Cryptology*, CRYPTO'05, pages 205–222. Springer, 2005.

[120] D. Boneh and M.K. Franklin. Identity-based encryption from the weil pairing. In *Proceedings of Advances in Cryptology*, CRYPTO'01, pages 213–229. Springer-Verlag, 2001.

[121] A. Beimel. Secret-sharing schemes: A survey. In *Proceedings of the 3rd international conference on Coding and Cryptology*, IWCC'11, pages 11–46. Springer-Verlag, 2011.

[122] C. Padró. Lecture notes in secret sharing. *IACR Cryptology ePrint Archive*, 2012: 674, 2012.

[123] H. Lipmaa. First CPIR protocol with data-dependent computation. In *ICISC*, volume 5984 of *Lecture Notes in Computer Science*, pages 193–210. Springer, 2009.

[124] Y. Shen, W. Yang, L. Li, and L. Huang. Achieving fully privacy-preserving private range queries over outsourced cloud data. *Pervasive and Mobile Computing*, 39: 36–51, 2017.

[125] B. Waters, D. Balfanz, G. Durfee, and D.K. Smetters. Building an encrypted and searchable audit log. In *Proceedings of the 11th Annual Network and Distributed System Security Symposium*, NDSS'04, 2004.

[126] E. Kiltz and D. Galindo. Direct chosen-ciphertext secure identity-based key encapsulation without random oracles. In *Proceedings of the 11th Information Security and Privacy Australasian Conference*, ACISP'06, pages 336–347. Springer Berlin Heidelberg, 2006.

[127] J. Katz, A. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, volume 4965 of *EUROCRYPT'08*, pages 146–162. Springer, 2008.

[128] B. Waters. Efficient identity-based encryption without random oracles. In *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, EUROCRYPT'05, pages 114–127. Springer Berlin Heidelberg, 2005.

[129] A. Shikfa, M. Önen, and R. Molva. Privacy and confidentiality in context-based and epidemic forwarding. *Computer Communications*, 33(13), 2010.

[130] P. Golle, J. Staddon, and B. Waters. Secure conjunctive keyword search over encrypted data. In *Proceedings of the second International Conference on Applied Cryptography and Network Security*, ACNS'04, pages 31–45. Springer Berlin Heidelberg, 2004.

[131] Z. Chen, C. Wu, D. Wang, and S. Li. Conjunctive keywords searchable encryption with efficient pairing, constant ciphertext and short trapdoor. In *Proceedings of Pacific Asia Workshop on Intelligence and Security Informatics*, PAISI'12, pages 176–189. Springer Berlin Heidelberg, 2012.

[132] D.F. Aranha, P.S.L.M. Barreto, P. Longa, and J.E. Ricardini. The realm of the pairings. In *Selected Areas in Cryptography*, volume 8282 of *Lecture Notes in Computer Science*, pages 3–25. Springer, 2013.

[133] R. Barbulescu, P. Gaudry, A. Joux, and E. Thomé. A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. In *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, volume 8441 of *EUROCRYPT'14*, pages 1–16. Springer, 2014.

[134] B. Libert and J.-J. Quisquater. On constructing certificateless cryptosystems from identity based encryption. In *Proceedings of the 9th International Conference on Theory and Practice of Public-Key Cryptography*, volume 3958 of *PKC'06*, pages 474–490. Springer, 2006.

[135] B. Lynn. PBC library. The pairing-based cryptography library, 2016. URL `http://crypto.stanford.edu/pbc/`.

[136] EU Parliament. Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an infrastructure for spatial information in the European Community (INSPIRE). *Official Journal of the European Union*, 50(L108), 2007.

[137] SEAL: Simple encrypted arithmetic library, cryptography research group, microsoft research. `http://sealcrypto.codeplex.com/`, 2016. Accessed: 11/12/2016.

[138] G. Matheron. *Traité de géostatistique appliquée*. Mémoires du Bureau de Recherches Géologiques et Minières. Éditions Technip, 1962-63.

[139] python-paillier: a library for partially homomorphic encryption in python, Data61|CSIRO. `https://github.com/NICTA/python-paillier`, 2016. Accessed: 11/12/2016.

[140] P.A. Burrough, R.A. McDonnell, and C.D. Lloyd. *Principles of geographical information systems*. Oxford University Press, 2015.

[141] L. Csirmaz. The size of a share must be large. *Journal of Cryptology*, 10(4): 223–231, 1997.

[142] R. Robere, T. Pitassi, B. Rossman, and S.A. Cook. Exponential lower bounds for monotone span programs. In *IEEE 57th Annual Symposium on Foundations of Computer Science*, FOCS'16, pages 406–415, 2016.

[143] E.F. Brickell and D.M. Davenport. On the classification of ideal secret sharing schemes. *Journal of Cryptology*, 4(2):123–134, 1991.

[144] I. Komargodski, M. Naor, and E. Yogev. Secret-sharing for NP. *Journal of Cryptology*, 30(2):444–469, 2017.

[145] V. Vaikuntanathan and P.N. Vasudevan. Secret sharing and statistical zero knowledge. In *Proceedings of the 21st Annual International Conference on the Theory and Application of Cryptology and Information Security*, ASIACRYPT'15, pages 656–680. Springer, 2015.

[146] J. Martí-Farré and C. Padró. On secret sharing schemes, matroids and polymatroids. In *Proceedings of the Theory of Cryptography Conference*, TCC'07, pages 273–290. Springer-Verlag, 2007.

[147] A.A. Razborov. Applications of matrix methods to the theory of lower bounds in computational complexity. *Combinatorica*, 10(1):81–93, 1990.

[148] A. Beimel, A. Gál, and M. Paterson. Lower bounds for monotone span programs. In *36th Annual Symposium on Foundations of Computer Science*, pages 674–681, 1995.

[149] A. Beimel, O. Farràs, and Y. Mintz. Secret sharing schemes for very dense graphs. In *Proceedings of Advances in Cryptology*, volume 7417 of *CRYPTO'12*, pages 144–161. Springer, 2012.

[150] P. Frankl. *Handbook of Combinatorics (Chapter 24, Extremal set systems)*. Elsevier, 1995.

[151] M. Bellare and P. Rogaway. Robust computational secret sharing and a unified account of classical secret-sharing goals. In *Proceedings of the 14th ACM Conference on Computer and Communications Security*, CCS'07, pages 172–184. ACM, 2007.

[152] M. Jha and S. Raskhodnikova. Testing and reconstruction of lipschitz functions with applications to data privacy. *SIAM Journal on Computing*, 42(2):700–731, 2013.

[153] L. Babai, A. Gál, and A. Wigderson. Superpolynomial lower bounds for monotone span programs. *Combinatorica*, 19:301–319, 1999.

[154] A. Beimel, O. Farràs, Y. Mintz, and N. Peter. Linear secret-sharing schemes for forbidden graph access structures. In *TCC (2)*, volume 10678 of *Lecture Notes in Computer Science*, pages 394–423. Springer, 2017.

[155] T. Liu, V. Vaikuntanathan, and H. Wee. Conditional disclosure of secrets via non-linear reconstruction. In *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference Proceedings, Part I*, pages 758–790, 2017.

[156] A. Beimel, O. Farràs, and N. Peter. Secret sharing schemes for dense forbidden graphs. In *Proceedings of the 10th Conference on Security and Cryptography for Networks*, SCN'16, pages 509–528, 2016.

[157] N. Alon and J.H. Spencer. *The Probabilistic Method*. Wiley Publishing, 4th edition, 2016.

[158] O. Farràs, T. Kaced, S. Martín Molleví, and C. Padró. Improving the linear programming technique in the search for lower bounds in secret sharing. In *EUROCRYPT (1)*, volume 10820 of *Lecture Notes in Computer Science*, pages 597–621. Springer, 2018.

[159] A. Beimel and I. Orlov. Secret sharing and non-shannon information inequalities. *IEEE Transactions on Information Theory*, 57(9):5634–5649, 2011.

[160] S. Martín, C. Padró, and A. Yang. Secret sharing, rank inequalities and information inequalities. In *Proceedings of Advances in Cryptology*, CRYPTO'13, Part II, pages 277–288. Springer Berlin Heidelberg, 2013.

[161] O. Farràs, T. Hansen, T. Kaced, and C. Padró. On the information ratio of non-perfect secret sharing schemes. *Algorithmica*, 2016.

[162] A. Gál. A characterization of span program size and improved lower bounds for monotone span programs. *Computational Complexity*, 10(4):277–296, 2001.

[163] T. Pitassi and R. Robere. Lifting nullstellensatz to monotone span programs over any field. In *STOC*, pages 1207–1219. ACM, 2018.

[164] S. Jukna. Boolean function complexity advances and frontiers. *Bulletin of the EATCS*, 113, 2014.

[165] I. Wegener. The complexity of symmetric boolean functions. In *Computation Theory and Logic*, volume 270 of *Lecture Notes in Computer Science*, pages 433–442. Springer, 1987.

[166] A.A. Razborov. On submodular complexity measures. In *Proceedings of the London Mathematical Society Symposium on Boolean Function Complexity*, pages 76–83. Cambridge University Press, 1992.

[167] A. Gaye, Y. Marcon, J. Isaeva, P. LaFlamme, A. Turner, E.M. Jones, J. Minion, A.W. Boyd, C.J. Newby, M.-L. Nuotio, and Marja-Liisa et al. DataSHIELD: taking the analysis to the data, not the data to the analysis. *International Journal of Epidemiology*, 43(6):1929–1944, 2014.

[168] M. Kantarcioglu. A survey of privacy-preserving methods across horizontally partitioned data. *Privacy-preserving data mining*, pages 313–335, 2008.

[169] H. Abu-Libdeh, L. Princehouse, and H. Weatherspoon. RACS: a case for cloud storage diversity. In *Proceedings of the 1st ACM Symposium on Cloud Computing*, SoCC '10, pages 229–240. ACM, 2010.

[170] H. Dev, T. Sen, M. Basak, and M.E. Ali. An approach to protect the privacy of cloud data from data mining based attacks. In *High Performance Computing, Networking, Storage and Analysis (SCC)*, pages 1106–1115. IEEE, 2012.

[171] Z. Wei, S. Xinwei, and X. Tao. Data privacy protection using multiple cloud storages. In *Proceedings of the 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC)*, pages 1768–1772. IEEE, 2013.

[172] R. Brinkman, S. Maubach, and W. Jonker. A lucky dip as a secure data store. In *Proceedings of Workshop on Information and System Security*, 2006.

[173] Q. Yang and X. Wu. 10 challenging problems in data mining research. *International Journal of Information Technology & Decision Making*, 5(04):597–604, 2006.

[174] W. Du, Y.S. Han, and S. Chen. Privacy-preserving multivariate statistical analysis: Linear regression and classification. In *Proceedings of the 2004 SIAM international conference on data mining*, pages 222–233. SIAM, 2004.

[175] A.F. Karr, X. Lin, A.P. Sanil, and J.P. Reiter. Privacy-preserving analysis of vertically partitioned data using secure matrix products. *Journal of Official Statistics*, 25(1):125, 2009.

[176] S. Ricci, J. Domingo-Ferrer, and D. Sánchez. Privacy-preserving cloud-based statistical analyses on sensitive categorical data. In *Modeling Decisions for Artificial Intelligence*, pages 227–238. Springer, 2016.

[177] A. Calviño, S. Ricci, and J. Domingo-Ferrer. Privacy-preserving distributed statistical computation to a semi-honest multi-cloud. In *2015 IEEE Conference on Communications and Network Security, CNS*, pages 506–514, 2015.

[178] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M.Y. Zhu. Tools for privacy preserving distributed data mining. *ACM Sigkdd Explorations Newsletter*, 4(2): 28–34, 2002.

[179] B. Goethals, S. Laur, H. Lipmaa, and T. Mielikäinen. On private scalar product computation for privacy-preserving data mining. In *ICISC*, volume 3506, pages 104–120. Springer, 2004.

[180] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou. Privacy-preserving multi-keyword ranked search over encrypted cloud data. *IEEE Transactions on Parallel and Distributed Systems*, 25(1):222–233, 2014.

[181] V. Guruswami, J. Hastad, and M. Sudan. Hardness of approximate hypergraph coloring. *SIAM Journal on Computing*, 31(6):1663–1686, 2002.

[182] D.J.A. Welsh and M.B. Powell. An upper bound for the chromatic number of a graph and its application to timetabling problems. *The Computer Journal*, 10(1): 85–86, 1967.

[183] F.T. Leighton. A graph coloring algorithm for large scheduling problems. *Journal of research of the national bureau of standards*, 84(6):489–506, 1979.

[184] D. Brélaz. New methods to color the vertices of a graph. *Communications of the ACM*, 22(4):251–256, 1979.

[185] M.W. Carter. A survey of practical applications of examination timetabling algorithms. *Operations research*, 34(2):193–202, 1986.

[186] J.A. de Loera. Gröbner bases and graph colorings. *Beiträge zur algebra und geometrie*, 36(1):89–96, 1995.

[187] J. De Loera, S. Margulies, M. Pernpeintner, E. Riedl, D. Rolnick, G. Spencer, D. Stasi, and J. Swenson. Graph-coloring ideals: Nullstellensatz certificates, gröbner bases for chordal graphs, and hardness of gröbner bases. In *Proceedings of the 2015 ACM on International Symposium on Symbolic and Algebraic Computation*, pages 133–140. ACM, 2015.

[188] C.J. Hillar and T. Windfeldt. Algebraic characterization of uniquely vertex colorable graphs. *Journal of Combinatorial Theory, Series B*, 98(2):400–414, 2008.

[189] F. Levy dit Vehel, M.G. Marinari, L. Perret, and C. Traverso. A survey on polly cracker systems. *Gröbner Bases, Coding, and Cryptography*, pages 285–305, 2009.

[190] D. Cox, J. Little, and D. O'shea. *Ideals, varieties, and algorithms*, volume 3. Springer, 1992.

[191] J.C. Faugere, P. Gianni, D. Lazard, and T. Mora. Efficient computation of zero-dimensional gröbner bases by change of ordering. *Journal of Symbolic Computation*, 16(4):329–344, 1993.

[192] J.A. Loera, J. Lee, S. Margulies, and S. Onn. Expressing combinatorial problems by systems of polynomial equations and Hilbert's Nullstellensatz. *Combinatorics, Probability and Computing*, 18(4):551–582, 2009.

[193] M. Hall. A problem in partitions. *Bulletin of the American Mathematical Society*, 47(10):804–807, 1941.

[194] P. Erdős, A.W. Goodman, and L. Pósa. The representation of a graph by set intersections. *Canadian Journal of Mathematics*, 18(106-112):86, 1966.

[195] J. Spencer and J.H. Spencer. *Ten lectures on the probabilistic method*, volume 52. Society for Industrial and Applied Mathematics Philadelphia, PA, 1987.

[196] The Sage mathematical software system. `http://www.sagemath.org/`, 2016.

[197] L.Sweeney. Simple demographics often identify people uniquely. *Health (San Francisco)*, 671:1–34, 2000.



UNIVERSITAT
ROVIRA i VIRGILI