Bringing Theory Closer to Practice in Post-quantum and Leakage-resilient Cryptography

THÈSE N^{O} 6807 (2015)

PRÉSENTÉE LE 13 NOVEMBRE 2015

À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS

LABORATOIRE DE SÉCURITÉ ET DE CRYPTOGRAPHIE

PROGRAMME DOCTORAL EN INFORMATIQUE ET COMMUNICATIONS

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Alexandre Raphaël DUC

acceptée sur proposition du jury:

Prof. E. Telatar, président du jury
Prof. S. Vaudenay, directeur de thèse
Prof. F.-X. Standaert, rapporteur
Prof. C. Cid, rapporteur
Prof. A. Lenstra, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2015

Je dédie cette thèse à mes amis et à ma famille qui m'ont soutenu durant cette thèse.

Abstract

Modern cryptography pushed forward the need of having *provable security*. Whereas ancient cryptography was only relying on heuristic assumptions and the secrecy of the designs, nowadays researchers try to make the security of schemes to rely on mathematical problems which are believed hard to solve. When doing these proofs, the capabilities of potential adversaries are modeled formally. For instance, the black-box model assumes that an adversary does not learn anything from the inner-state of a construction. While this assumption makes sense in some practical scenarios, it was shown that one can sometimes learn some information by other means, e.g., by timing how long the computation take. In this thesis, we focus on two different areas of cryptography. In both parts, we take first a theoretical point of view to obtain a result. We try then to adapt our results so that they are easily usable for implementers and for researchers working in practical cryptography.

In the first part of this thesis, we take a look at *post-quantum cryptography*, i.e., at cryptographic primitives that are believed secure even in the case (reasonably big) quantum computers are built. We introduce HELEN, a new public-key cryptosystem based on the hardness of the learning from parity with noise problem (LPN). To make our results more concrete, we suggest some practical instances which make the system easily implementable.
As stated above, the design of cryptographic primitives usually relies on some well-studied hard problems. However, to suggest concrete parameters for these primitives, one needs to know the precise complexity of algorithms solving the underlying hard problem. In this thesis, we focus on two recent hard-problems that became very popular in post-quantum cryptography: the learning with error (LWE) and the learning with rounding problem (LWR). We introduce a new algorithm that solves both problems and provide a careful complexity analysis so that these problems can be used to construct practical cryptographic primitives.

In the second part, we look at *leakage-resilient cryptography* which studies adversaries able to get some side-channel information from a cryptographic primitive. In the past, two main disjoint models were considered. The first one, the threshold probing model, assumes that the adversary can put a limited number of probes in a circuit. He then learns all the values going through these probes. This model was used mostly by theoreticians as it allows very elegant and convenient proofs. The second model, the

noisy-leakage model, assumes that every component of the circuit leaks but that the observed signal is noisy. Typically, some Gaussian noise is added to it. According to experiments, this model depicts closely the real behaviour of circuits. Hence, this model is cherished by the practical cryptographic community. In this thesis, we show that making a proof in the first model implies a proof in the second model which unifies the two models and reconciles both communities. We then look at this result with a more practical point-of-view. We show how it can help in the process of evaluating the security of a chip based solely on the more standard mutual information metric.

Résumé

La cryptographie moderne a développé le besoin de *prouver la sécurité* de ses systèmes. Alors que la sécurité de la cryptographie antique reposait sur des suppositions heuristiques ainsi que sur la confidentialité des constructions, de nos jours, les chercheurs tentent plutôt de faire reposer la sécurité sur des problèmes mathématiques supposés difficiles à résoudre. Afin de pouvoir faire ces preuves de sécurité, il est important de modéliser formellement les capacités de potentiels adversaires. Par exemple, un modèle très courant utilisé en cryptographie est le modèle dit "en boîte noire". Dans ce modèle, nous supposons qu'un adversaire n'apprend rien de l'état interne du système mais connait uniquement ses entrées et sorties. Bien que cette supposition ait du sens dans certains scénarios, des chercheurs ont montré qu'il est parfois possible de dériver des secrets par d'autres moyens, par exemple en mesurant le temps mis par le système pour faire un calcul. Dans cette thèse, nous nous focalisons sur deux branches distinctes de la cryptographie. Dans ces deux parties, nous allons tout d'abord prendre un point de vue théorique. Nous allons ensuite adapter nos résultats afin qu'ils soient facilement utilisables par des chercheurs travaillant dans le côté plus pratique de la cryptographie, par exemple dans l'implémentation.

Dans la première partie de cette thèse, nous étudions la *cryptographie post-quantique*, c'est-à-dire la cryptographie qui est supposée être sûre même si des ordinateurs quantiques suffisamment puissants sont construits. Nous proposons HELEN, un nouveau cryptosystème à clé publique, dont la sécurité est basée sur la difficulté du problème "learning from parity with noise" (LPN). Afin de rendre nos résultats plus concrets, nous proposons aussi des paramètres pratiques qui rendent notre système facile à implémenter.
Comme mentionné ci-dessus, la construction de primitives cryptographiques est basée sur la complexité de problèmes supposés difficiles. Afin de pouvoir proposer des paramètres concrets pour ces primitives, il est nécessaire de connaître la complexité exacte des algorithmes capables de résoudre ces problèmes. Dans cette thèse, nous nous focalisons sur deux problèmes introduits récemment qui sont devenus très populaires en cryptographie post-quantique: le problème "learning with error" (LWE) et le problème "learning with rounding" (LWR). Nous proposons un nouvel algorithme capable de résoudre ces problèmes et nous fournissons une analyse détaillée de sa complexité afin que les problèmes LWE et LWR puissent être utilisés pour construire des instances pratiques de primitives cryptographiques.

Dans la seconde partie de cette thèse, nous étudions la *cryptographie résistante aux attaques par canaux auxiliaires*. Cette branche de la cryptographie étudie le cas où un adversaire est capable d'obtenir des informations supplémentaires sur une primitive cryptographique à l'aide d'un canal auxiliaire. Par le passé, deux modèles principaux ont été considérés dans ce domaine. Le premier, le "threshold probing model", suppose qu'un adversaire puisse mettre un nombre limité de sondes dans un circuit. A l'aide de ces sondes, l'adversaire apprend ensuite toutes les valeurs passant à travers celles-ci. ce modèle a été principalement utilisé par les chercheurs en cryptographie théorique, car il permet de concevoir des preuves plus faciles et plus élégantes. Le second modèle, le "noisy-leakage model", suppose que chaque composant du circuit fuit, mais que les signaux observés sont bruités. Typiquement, le modèle va supposer qu'un bruit Gaussien est ajouté au signal. Des expériences ont montré que ce modèle décrit assez fidèlement le comportement réel des circuits. Il est donc normal que ce modèle soit préféré par les chercheurs travaillant dans le côté pratique de la cryptographie. Dans cette thèse, nous montrons que faire une preuve dans le premier modèle implique une preuve dans le second. Ce résultat unifie les deux modèles et réconcilie les deux communautés. Nous analysons ensuite ce résultat avec un regard plus pratique. Nous montrons comment ce résultat peut aider à estimer la sécurité de puces en utilisant uniquement une métrique basée sur l'information mutuelle.

Acknowledgments

First, I would like to express my gratitude towards my supervisor Prof. Serge Vaudenay. Thank you Serge for trusting in me. I appreciated the freedom you gave me, especially regarding teaching. Besides teaching me how to do research properly and cryptography, you also taught me the importance of formalism in science. Although in the beginning it sounded a bit extreme to me, you managed to convince me that without some formalism the research one produces is usually worthless. Thank you as well for always trying to take decision that are best for your PhD students, even when it was not obvious for us. I enjoyed particularly the atmosphere you managed to give to the lab. Sometimes, it was like a second family for me.

Then, I would like to warmly thank the members of my thesis committee. I thank Prof. Emre Telatar, my jury president. Thank you Emre for your eternal kindness in all circumstances. I also express my gratitude towards Prof. Carlos Cid who made the trip from London to attend my PhD defense. I also want to thank Prof. François-Xavier Standaert. Thanks FX for these fascinating discussions on how to approach cryptography with a more practical point of view and for some endless debates with Sebastian. Finally, I wish to thank Prof. Arjen Lenstra for accepting to be a member of my jury even though some areas of my thesis were not his cup of tea. Thanks Arjen for your support during my PhD when I needed it.

Next, I would like to thank the people with whom I shared my office at EPFL (INF 239). I thank Dr Pouyan Sepehrdad who welcomed me in his office when I started my PhD. Thank you for all the good laughs we had and for all the discussions about (Iranian) marriage. I am also grateful towards my second office mate Sonia M. Bogos. Thank you Sonia for bringing happiness into the office, for sometimes standing my rants with a smile, and for not killing me during our fights. I appreciated also all the sweets you always kindly offered the lab. Finally, I wish to thank Wilson[1] for being all the time quiet and for never letting the third chair of our office empty.

I warmly thank Martine Corval, our secretary. Thank you Martine for being always present when we needed your help. You were especially helpful regarding all the complicated EPFL administrative regulations.

[1] Wilson is Sonia's bag.

Every PhD student seem to have his moment of doubts. I am grateful to Prof. Serge Vaudenay, Prof. Arjen Lenstra, Prof. Martin Vetterli, Prof. Dimitar Jetchev, many friends, and my family for helping me to make the correct decision in these dark days.

Then, I would like to warmly thank all my coauthors: Prof. Serge Vaudenay, Prof. Dimitar Jetchev, Prof. Sebastian Faust, Florian Tramèr, and Prof. Stefan Dziembowski. Dimitar, it was a pleasure for me to work with you on a completely new topic. I also enjoyed all the hikes we did together. Sebastian, thank you for introducing me to the leakage-resilient world. I really enjoyed the amount of new ideas you had. It allowed me to work on a big variety of topics. We had also great laughs together while working on some research topics. Florian, collaborating with you was a pleasure as things went extremely fast. I hope you will have a successful PhD in Standford.

Next, I would like to thank all the members of LASEC I shared some time with during my PhD: Prof. Serge Vaudenay, Martine Corval, Dr Philippe Oechslin, Dr Pouyan Sepehrdad, Dr Atefeh Mashatan, Dr Petr Sušil, Dr Aslı Bay, Prof. Katerina Mitrokotsa, Dr Ioana Boureanu–Carlson, Iosif Spulber, Sonia M. Bogos, Dr Miyako Okubo, Dr Reza Reyhanitabar, Prof. Sebastian Faust, Dr Jialin Huang, Damian Vizár, Florian Tramèr, Dr Divesh Aggarwal, Dr Adeline Langlois–Roux, and Handan Kılınç. Atefeh, thank you for bringing your dynamism into the lab. (Scri)Petr, thank you for never raising your voice, for all your help when managing LASEC machines, for your Bash scripts, and for endless discussions about algebraic attacks. Aslı, thank you for enlightening the lab with your daughter Zeynep. Katerina, thanks for always proposing animated group meetings. Ioana, thank you for your engagement in the lab and congratulation for your great wedding in Romania. Miyako, your stay at LASEC was refreshing and I learned a lot about the Japanese culture. Reza, thank you for your kindness. Damian, thank you for always having time for helping people and for taking everything with a smile. TAing the cryptography course with you was a pleasure. Divesh, thank you for some insightful comments and interesting ideas. Adeline, I am grateful for all the fascinating discussion and gossips we had together. I hope your new position in Rennes will fit you. Handan, you represent the new generation of LASEC: good luck! I would also like to thanks all the students that revolved around LASEC during these four years. Explaining to all of you the beauties of cryptography was always a fun challenge.

I wish to thank all my friends, hiking partners, and travelling partners. Thanks for all the great moments spent outside research. In particular, I thank Régis Blanc and Manohar Jonnalagedda for fruitful coffee breaks at EPFL and endless debates regarding the PhD (among other topics).

Finally I would like to thank my parents Anne-Lise and Jean-Michel Duc for their support during these four years of PhD. Thank you for trusting me during this PhD and for helping me to go through the more difficult moments. I am also grateful to my brother Sébastien and my sister Marie for many fun discussion about EPFL during the past years.

Remerciements

Tout d'abord, je souhaite exprimer ma gratitude envers mon superviseur, le Prof. Serge Vaudenay. Merci Serge pour ta confiance en moi. J'ai beaucoup apprécié la liberté que tu m'as laissée, particulièrement pour l'enseignement. En plus de m'apprendre comment faire de la recherche correctement ainsi que tout ce qu'il faut savoir sur la cryptographie, tu m'as aussi enseigné l'importance du formalisme en science. Il est vrai qu'au début cela m'avait semblé un peu exagéré, mais tu m'as vite convaincu que sans formalisme, la recherche produite était souvent sans valeur. Merci aussi pour toujours essayer de prendre des décisions qui sont pour le mieux pour tes doctorants, même lorsque cela n'était pas évident pour nous. J'ai particulièrement apprécié l'atmosphère que tu as réussi à apporter au laboratoire. Je me sentais parfois quasiment comme dans une seconde famille pour moi.

Ensuite, je souhaite remercier chaleureusement les membres de mon comité de thèse. Je remercie le Prof. Emre Telatar, le président de mon jury. Merci Emre pour votre gentillesse infinie en toutes circonstances. J'aimerais aussi exprimer ma gratitude envers le Prof. Carlos Cid qui a fait le voyage depuis Londres uniquement pour assister à ma thèse. Je remercie aussi le Prof. François-Xavier Standaert. Merci FX pour toutes les discussions fascinante que nous avons eues, dans lesquelles tu m'as montré comme approcher la cryptographie avec une vision plus réaliste. Merci pour tous ces débats quasi infinis que nous avons eus avec Sebastian. Finalement, je souhaite remercier le Prof. Arjen Lenstra pour avoir accepté d'être un membre de mon jury même si certains sujets de ma thèse n'étaient pas sa tasse de thé. Merci Arjen pour ton soutien pendant ma thèse, surtout lorsque j'en avais besoin.

J'aimerais maintenant remercier toutes les personnes avec lesquelles j'ai partagé mon bureau à l'EPFL (INF 239). Je remercie tout d'abord le Dr Pouyan Sepehrdad qui m'a accueilli dans son bureau lorsque j'ai commencé ma thèse. Merci pour tous les fous rires que nous avons eus ainsi que pour toutes les discussions sur le mariage (iranien). Je suis aussi reconnaissant envers ma deuxième collègue de bureau, Sonia M. Bogos. Merci Sonia pour toujours apporter de la joie dans notre bureau, pour parfois supporter mes coups de gueule avec le sourire et pour ne pas m'avoir tué pendant nos bagarres. J'ai aussi beaucoup apprécié toutes les sucreries que tu offres toujours gentiment au labo.

Finalement, je souhaite remercier Wilson[2] pour être toujours silencieux et pour ne jamais laisser libre la troisième chaise de notre bureau.

Je remercie chaleureusement Martine Corval, notre secrétaire. Merci Martine pour toujours être présente lorsque nous avions besoin d'aide. Tu m'as été d'une très grande utilité lorsque j'ai eu affaire à toutes les règles administratives très compliquées de l'EPFL.

Il semble que chaque doctorant traverse à un moment une période de doutes. Je suis reconnaissant envers le Prof. Serge Vaudenay, le Prof. Arjen Lenstra, le Prof. Martin Vetterli, le Prof. Dimitar Jetchev, plusieurs amis et ma famille pour m'avoir aidé à prendre les bonnes décisions pendant ces instants difficiles.

Je voudrais ensuite remercier chaleureusement tous mes coauteurs pendant cette thèse: le Prof. Serge Vaudenay, le Prof. Dimitar Jetchev, le Prof. Sebastian Faust, Florian Tramèr et le Prof. Stefan Dziembowski. Dimitar, cela a été un plaisir pour moi de travailler avec toi sur un sujet complètement nouveau. J'ai eu beaucoup de plaisir durant les randonnées que nous avons faites ensemble. Sebastian, merci de m'avoir fait découvrir la cryptographie résistante aux attaques par canaux auxiliaires. J'ai énormément apprécié toutes les nouvelles idées que tu avais chaque jour. Cela m'a permis de travailler sur une grande variété de sujets. Je me souviens aussi de grands fous rires lorsque nous travaillions sur certains sujets de recherche. Florian, collaborer avec toi a été un plaisir, car tout s'est passé extrêmement rapidement. J'espère que tu feras un brillant doctorat à Standford.

J'aimerais maintenant remercier tous les membres du LASEC avec lesquels j'ai passé un peu de temps durant ma thèse: le Prof. Serge Vaudenay, Martine Corval, le Dr Philippe Oechslin, le Dr Pouyan Sepehrdad, le Dr Atefeh Mashatan, le Dr Petr Sušil, le Dr Aslı Bay, le Prof. Katerina Mitrokotsa, le Dr Ioana Boureanu–Carlson, Iosif Spulber, Sonia M. Bogos, le Dr Miyako Okubo, le Dr Reza Reyhanitabar, le Prof. Sebastian Faust, le Dr Jialin Huang, Damian Vizár, Florian Tramèr, le Dr Divesh Aggarwal, le Dr Adeline Langlois–Roux et Handan Kılınç. Atefeh, merci d'avoir apporté ton dynamisme au labo. (Scri)Petr, merci pour n'avoir jamais élevé la voix, pour toute ton aide lorsque nous devions gérer les ordinateurs du LASEC, pour tous tes scripts en Bash et pour toutes ces discussions infinies sur les attaques algébriques. Aslı, merci d'avoir illuminé le labo avec ta fille Zeynep. Katerina, merci pour avoir toujours organisé des LASEC meeting animés. Ioana, merci pour ton engagement au labo et félicitations pour ton super mariage en Roumanie. Miyako, ton séjour au LASEC était rafraichissant et, grâce à toi, j'ai beaucoup appris sur la culture japonaise. Reza, je te remercie pour ta gentillesse. Damian, merci pour toujours avoir du temps pour aider les autres et pour toujours prendre les choses avec le sourire. Etre un assistant au cours de cryptographie avec toi a été un plaisir. Divesh, merci pour m'avoir donné quelques idées intéressantes. Adeline, j'ai énormément apprécié les discussions fascinantes et les ragots que nous avons

[2]Wilson est le sac de Sonia.

eus ensemble. J'espère que ton nouveau poste à Rennes te plaira. Handan, tu représentes la nouvelle génération du LASEC: bonne chance ! J'aimerais aussi remercier tous les étudiants qui ont révolu autour du LASEC pendant ces quatre ans. Cela a toujours été un challenge amusant de vous expliquer les beautés de la cryptographie.

Je souhaiterais remercier tous mes amis, mes partenaires de randonnées et de voyages. Merci pour tous ces beaux moments passés en dehors de la recherche. Je remercie particulièrement Régis Blanc et Manohar Jonnalagedda pour des pauses café fructueuses et des débats éternels sur le doctorat (parmi tant de sujets).

Finalement, je remercie mes parents Anne-Lise et Jean-Michel Duc pour leur soutien pendant ces quatre ans de thèse. Merci de m'avoir fait confiance pendant ce doctorat et de m'avoir aidé à traverser les moments plus difficiles. Je remercie aussi mon frère Sébastien et ma sœur Marie pour plein de discussions amusantes sur l'EPFL ces dernières années.

Personal Bibliography

In the following list, you will find all the papers that were published during this thesis in reverse chronological order. Entries in bold are included in this dissertation.
Note: Paper [4] is an extended version of Paper [6].

[1] **Alexandre Duc, Florian Tramèr, and Serge Vaudenay. Better Algorithms for LWE and LWR. In Elisabeth Oswald and Marc Fischlin, editors.** ***Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26- 30, 2015. Proceedings, Part I, volume 9056 of Lecture Notes in Computer Science, pages 173-202. Springer, 2015.***

[2] **Alexandre Duc, Sebastian Faust, and François-Xavier Standaert. Making Security Proofs Concrete - Or How to Evaluate the Security of Any Leaking Device. In Elisabeth Oswald and Marc Fischlin, editors.** ***Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26- 30, 2015. Proceedings, Part I, volume 9056 of Lecture Notes in Computer Science, pages 401-429. Springer, 2015.***

[3] **Alexandre Duc, Stefan Dziembowski, and Sebastian Faust. Unifying Leakage Models: From Probing Attacks to Noisy Leakage. In** ***Phong Q. Nguyen and Elisabeth Oswald, editors. Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings, volume 8441 of Lecture Notes in Computer Science, pages 423-440. Springer, 2014.***

[4] **Alexandre Duc and Serge Vaudenay. HELEN: A Public-Key Cryptosystem Based on the LPN and the Decisional Minimal Distance Problems. In Amr Youssef, Abderrahmane Nitaj, and Aboul Ella Hassanien, editors,** ***Progress in Cryptology - AFRICACRYPT 2013, 6th International***

Conference on Cryptology in Africa, Cairo, Egypt, June 22-24, 2013. Proceedings, volume 7918 of Lecture Notes in Computer Science, pages 107–126. Springer, 2013

[5] Alexandre Duc and Serge Vaudenay. TCHo: A Code-Based Cryptosystem. In Evangelos Kranakis, editor, *Advances in Network Analysis and its Applications, volume 18 of Mathematics in Industry, pages 149–179. Springer Berlin Heidelberg, 2013.*

[6] **Alexandre Duc and Serge Vaudenay. HELEN: a Public-key Cryptosystem Based on the LPN and the Decisional Minimal Distance Problems (Extended Abstract). In Yet Another Conference on Cryptography. 2012.**

[7] Alexandre Duc and Dimitar Jetchev. Hardness of Computing Individual Bits for One-Way Functions on Elliptic Curves. In Reihaneh Safavi-Naini and Ran Canetti, editors. *Advances in Cryptology - CRYPTO 2012, volume 7417 of Lecture Notes in Computer Science, pages 832-849. Springer, 2012.*

[8] Alexandre Duc, Jian Guo, Thomas Peyrin, and Lei Wei. Unaligned Rebound Attack: Application to Keccak. In Anne Canteaut, editor. *Fast Software Encryption - 19th International Workshop, FSE 2012, volume 7549 of Lecture Notes in Computer Science, pages 402-421. Springer, 2012.*

Table of Contents

Abstract **iii**

Résumé **v**

Acknowledgments **vii**

Remerciements **ix**

Personal Bibliography **xiii**

Table of Content **xviii**

1 Introduction **1**

1.1 On the Need for Diversity in Public-key Cryptography 1

1.2 Post-quantum Cryptography 2

1.3 On the Concrete Algorithmic Hardness of Hard Problems 3

1.4 Leakage-resilient Cryptography 4

1.5 Unifying Different Leakage Models 4

1.6 Towards a More Practical Interpretation of this Result 5

2 Preliminaries **7**

2.1 Notations and Mathematical Preliminaries 7

2.2 Statistical Distance 8

2.3 Security Notions 9

2.4 Fourier Transforms 11

2.4.1 Discrete Fourier Transform 11

2.4.2 Continuous Fourier Transforms 12

2.5 Various Bounds 13

I Post-quantum Cryptography **15**

3 Hard Problems in Post-Quantum Cryptography **17**
3.1 The Learning from Parity with Noise Problem . . . 17
3.1.1 The Decisional LPN Problem. . . . 18
3.1.2 Algorithms that Solve the LPN Problem . . . 18
3.1.3 Solving LPN with a Polynomial Number of Samples . . . 19
3.2 Finding a Low-weight Codeword in a Random Linear Code . . . 20
3.3 The Learning With Error Problem . . . 21
3.3.1 Gaussian Distributions . . . 22
3.4 The Learning With Rounding Problem . . . 23

4 The BKW Algorithm for LWE **25**
4.1 Sample Reduction . . . 25
4.2 Hypothesis Testing . . . 26
4.3 Back Substitution . . . 27
4.4 The LF1 Algorithm . . . 27

5 The HELEN Cryptosystem **29**
5.1 Related Work . . . 29
5.2 The Cryptosystem . . . 31
5.2.1 Encryption . . . 32
5.2.2 Decryption . . . 32
5.2.3 Key Generation . . . 33
5.3 Security Analysis . . . 34
5.3.1 Link to Random Codes . . . 34
5.3.2 Semantic Security . . . 35
5.4 Selection of Parameters . . . 37
5.5 Encrypting More than One Bit . . . 38
5.5.1 Security . . . 39
5.6 IND-CCA-security . . . 40

6 A New Algorithm Solving the Learning With Error Problem **43**
6.1 Previous Work . . . 43
6.2 Our Contribution . . . 44
6.3 Our LWE Algorithm . . . 44
6.3.1 Sample reduction . . . 45
6.3.2 Hypothesis testing . . . 45
6.3.3 Back Substitution . . . 51
6.4 Complexity of BKW with Multidimensional DFT on LWE . . . 51
6.5 Reducing the Number of Samples . . . 53
6.5.1 Lyubashevsky's Idea . . . 53
6.5.2 The LF2 Heuristic . . . 54

6.6 Results . . . 54

7 A New Algorithm Solving the Learning With Rounding Problem 57
7.1 The LWR-solving Algorithm . . . 58
7.2 Complexity of BKW with multidimensional DFT on LWR . . . 62
7.3 Results . . . 63

II Leakage-resilient Cryptography 65

8 Introduction to Leakage-resilient Cryptography 67
8.1 The Masking Countermeasure . . . 68
8.1.1 Boolean Masking . . . 68
8.1.2 Additive Masking . . . 68
8.1.3 Inner-product Masking . . . 69
8.2 Modeling the Leakage . . . 69
8.3 Threshold Probing Model . . . 70
8.4 Random Probing Model . . . 70
8.5 Noisy Model . . . 71
8.5.1 Modeling the Noise . . . 72
8.5.2 Adversarial Model . . . 74
8.5.3 Other Models . . . 75

9 Unifying Leakage Models 77
9.1 The Work of Prouff and Rivain [PR13] . . . 77
9.2 Our Contribution . . . 79
9.3 Related Work . . . 80
9.4 Noise from Set Elements . . . 81
9.4.1 Simulating Noise by ϵ-identity Functions . . . 81
9.5 Leakage from Vectors . . . 85
9.5.1 Simulating the Noisy Adversary by a Random-probing Adversary 85
9.5.2 Simulating Random-probing by a Threshold-probing . . . 86
9.6 Leakage from Computation . . . 87
9.6.1 Arithmetic Circuits . . . 88
9.6.2 Implementing a Secure Circuit Compiler . . . 91
9.6.3 Security in the Probing Model [ISW03] . . . 93
9.6.4 Resilience to Noisy Leakage from the Wires . . . 94
9.6.5 Resilience to Noisy Leakage from the Gates . . . 95

10 From Theory to Practice 99
10.1 Assessing Security of Concrete Devices . . . 99
10.2 Background . . . 100
10.2.1 Evaluation Metrics . . . 101

10.2.2 Metrics to Quantify the Security Result . . . 103
10.3 Making Proofs Concrete: Theory . . . 103
10.3.1 From Statistical Distance to Mutual Information . . . 104
10.3.2 Security of the Encoding . . . 105
10.3.3 Security of the Whole Circuit . . . 107
10.4 Experimental Validation . . . 107
10.4.1 Intuition Behind the Noise Condition . . . 108
10.4.2 Tightness of the Bounds . . . 110

Conclusion and Further Work **113**
HELEN . . . 113
LWE and LWR . . . 114
Leakage Models . . . 114
Final Words . . . 116

List of Algorithms **119**

List of Definitions **122**

Bibliography **123**

Curriculum Vitae **145**

Chapter 1

Introduction

The cryptographic world is roughly divided into two parts:

- the "theory" side which tries to simulate the reality using theoretical models, design new primitives, and study their asymptotic complexity.
- the "practical" side which consists in implementing primitives, studying precise complexities of schemes, measuring the behaviour of real device and analyse them.

Both communities have their own conferences (e.g., CHES for the practical community and TCC for the theoreticians) and rarely collaborate together. The recent introduction at Eurocrypt 2015 of two separate tracks — the *ideal track* and the *real track* — can be understood in two different ways. On one hand, this could mean that theoretical conferences will accept more practical papers. On the other hand, this could also be seen as an accentuation in the separation between the two communities. Indeed, in the past, people were encouraged to attend all the sessions. This is less likely to occur with this system as the more theoretical people will remain in the ideal track and vice versa. In this thesis, we develop four different results that try to bring the two sides closer together. More precisely, we start by taking a theoretical point of view on a problem, solve it, and present our results such that someone working on the practical side can make use of it.
This thesis is divided into two parts. In the first we look at the field of *post-quantum* cryptography and in the second at *leakage-resilient* cryptography. In the remaining of this introduction, we motivate our research directions, we describe in a high-level fashion the results we obtained, and we show how we believe these results have a practical impact although most of them are considered as theoretical papers.

1.1 On the Need for Diversity in Public-key Cryptography

In Chapter 5, we propose a new public-key cryptosystem based on some more recent hardness assumption. At first, one could think that we have already good cryptosystems

as they are used every day to secure our communications over the internet. Public-key cryptography initially started with the seminal paper of Diffie and Hellman in 1976 which introduces the Diffie-Hellman key exchange protocol [DH76]. Two years later, Rivest, Shamir, and Adleman introduced the nowadays most famous public-key cryptosystem: the RSA cryptosystem [RSA78]. From these two results, other popular cryptosystems emerged like the Rabin cryptosystem [Rab79] and the ElGamal family of cryptosystems [EG84]. All these systems are relatively efficient in practice, well-studied, and implemented on any thinkable platform.

The need for new cryptosystems rises when one understands how the security of a scheme is proven. When proving a security property, researchers usually relate it to a well-studied mathematical problem which is believed hard to solve. Typical examples of such hard problems are the integer factoring problem or the discrete logarithm problem. However, the number of hard problems used in popular cryptosystems is very small (in fact, the two problems mentioned above are the only used ones in practice). This leads to the following question: "What if one can suddenly easily solve these hard problems using a new clever algorithm?" The answer is simple. It would imply that all the public-key cryptosystems used in practice would be broken.

In particular, it was shown in the nineties that the integer factoring and the discrete logarithm problems could easily be solved on a *quantum computer* using *Shor's algorithm* [Sho97] and its generalizations, e.g., [HV09]. At the time of this writing, it is not yet known if one can ever build quantum computers that can have a large (i.e., meaningful) number of qubits.[1]

To be prepared to this eventuality, we need *crypto diversity*, namely to have cryptosystems that rely on a wide range of hard problems. In particular we want cryptosystems in which we can trust even if quantum computers exist. These systems are called *post-quantum cryptosystems*. Post-quantum cryptography is, since a few years, a hot topic. Dozens of quantum-resistant systems already exist. In Part I, we will focus on post-quantum cryptography.

1.2 Post-quantum Cryptography

Nowadays, recent cryptosystems usually rely on problems that are believed to be hard to solve, even on quantum computers. In this thesis, we will focus on three such problems: the Learning from Parity with Noise problem (LPN) and two closely related problems, the Learning with Error problem (LWE) and the Learning with Rounding problem (LWR). In Chapter 3, we formally describe these problems, discuss previous work and discuss their links with other hard problems.

In particular, we will have a strong interest in the algorithmic complexity of the best algorithms solving these problems. This is especially important when one wants to design a cryptosystem based on these problems. Indeed, once a reduction between a hard problem and the security of the system has been shown, this does not help finding

[1] On a quantum computer, the memory is represented in qubits instead of bits.

practical parameters for the system as reductions are usually asymptotic. Obviously, to be used in practice, the implementers of this system will need such parameters that can be trusted. For this, the typical approach is to survey what the best attacks against the hard problem are and to select parameters such that complexity of this attack is higher than a security bound (usually 2^λ for a security parameter λ).
In Chapter 5, we present HELEN, a new post-quantum cryptosystem, mostly based on the hardness of the LPN problem and the decisional minimum distance problem (a related hard problem introduced in Chapter 3). HELEN was first presented at YACC 2012 [DV12] and later published at Africacrypt 2013 [DV13a]. We show that the resulting cryptosystem achieves indistinguishability under chosen plaintext attacks (IND-CPA security). Using the Fujisaki-Okamoto generic construction [FO99], HELEN achieves IND-CCA security in the random oracle model.
When we submitted [DV12], HELEN was one of the first public-key cryptosystems based on the hardness of LPN. Moreover, in contrast with previous work (e.g., [Ale03]), we were the first to present *concrete parameters* for our system which is in our opinion a primordial step if we expect it to be implemented some day. As mentioned above, the lack of practical instances in most of the public-key cryptosystem proposals annihilates their interest for the industry. This is why we believe that our proposal tries to narrow the gap between theory and practice.

1.3 On the Concrete Algorithmic Hardness of Hard Problems

We already discussed in Section 1.2 the practical importance of a careful algorithmic study of hard problems used in cryptography. The learning with errors and the learning with rounding problem are in particular getting more and more popular for recent applications. For instance, the LWE problem can be used to design a public-key cryptosystem [Reg05] or more complex primitives like identity-based cryptosystems [GPV08] or fully homomorphic encryption [BV11a, Bra12, GSW13], one of the holy grails in cryptography. The BKW algorithm was suggested by Blum et al. [BKW03] as an algorithm to solve the Learning Parity with Noise problem (LPN), a subproblem of LWE. This algorithm was then adapted to LWE [ACF+13] by Albrecht et al. In Chapter 6, we improve the algorithm suggested by Albrecht et al. by using multidimensional Fourier transforms. Our algorithm was, at time of publication, the fastest LWE solving algorithm for some meaningful instances. Compared to the work of Albrecht et al. we greatly simplify the analysis, getting rid of integrals which were hard to evaluate in the final complexity. We also remove some heuristics on rounded Gaussians and some of our analytic results on rounded Gaussians might be of independent interest. Moreover, we also analyze algorithms solving LWE with discrete Gaussian noise.
The LWR problem can be seen as a deterministic counterpart to LWE. This problem is getting more and more attention and is used, for instance, to design pseudorandom functions [BPR12]. We adapt in Chapter 7 our LWE algorithm to the LWR problem

for prime q. To the best of our knowledge, our algorithm is the first algorithm applied directly to LWR. Furthermore, the analysis of LWR contains some technical results of independent interest.
In summary, these two results will help designers of constructions to put forward practical instances of their schemes. These two results were published in Eurocrypt 2015 [DTV15].

1.4 Leakage-resilient Cryptography

In Part II, we focus on a different area of cryptography, namely *leakage-resilient cryptography* and try to narrow the gap between the theoretical world and the practical world. Most of the cryptographic research is done in the *black-box model*. This means that we assume that all the building blocks (e.g., encryption algorithm, signature algorithm, hash function) behave like black boxes, i.e., the adversary can only see their input and output. On the other extreme, there is the *white-box model* in which we assume that the adversary has a complete access to all the inner-states of the algorithms. An intermediate notion is the *grey-box model*, where we assume that the adversary has only limited access to the inner components. For instance, one might think of the following practical scenario in which the adversary has physical access to a chip but in which physical constraints allow him only to read the values going through a limited number of wires in this chip. As we will see later, this model, named the (threshold) *probing model*, is very convenient for proving security. However, a seminal paper by Chari et al. [CJRR99] has shown that the best way to model leakage in practice is to assume that a noisy instance of every wire is given to the adversary. Typically, the noise would follow a Gaussian distribution. A generalization of this model was introduced by Prouff and Rivain [PR13], the *noisy-leakage model*.
The goal of most of the papers mentioned in this thesis as well as in our results is to construct a generic compiler that takes as input a vulnerable circuit and outputs a leakage-resilient circuit in a specific leakage model. Such a secure circuit is obtained by using a *masking scheme* or *secret sharing scheme* that allows to split a critical component into subcomponents that can be recombined when having access to all of them. Their use in leakage-resilient cryptography is straight-forward: we assume that it is hard for an adversary to access to a lot of information in a circuit at the same time. For instance, if the adversary uses probes to read some values in a chip, it might become technically hard for him to probe too many values. In this thesis, we are going to focus on the *additive masking scheme* that is well-studied in the literature.
In Chapter 8, we introduce briefly leakage-resilient cryptography and we show how to model leakage.

1.5 Unifying Different Leakage Models

As discussed in the previous section, the goal of leakage-resilient cryptography is to formally show the leakage resilience of cryptographic implementations in a given leakage

model. One of the most prominent leakage models – the so-called bounded leakage model – assumes that the amount of leakage is a-priori bounded. Unfortunately, it has been pointed out that the assumption of bounded leakages is hard to verify in practice. A more realistic assumption is to assume that leakages are sufficiently noisy, following the engineering observation that real-world physical leakages are inherently noisy. While the noisy leakage assumption has first been studied in the seminal work of Chari et al. (CRYPTO 99), the recent work of Prouff and Rivain (Eurocrypt 2013) provides the first analysis of a protected scheme under a physically motivated noise model. In particular, the authors show that a block-cipher implementation that uses an additive masking scheme is secure against noisy leakages. Unfortunately, the security analysis of Prouff and Rivain has three important shortcomings:

1. It requires leak-free gates, i.e., simple sub-components in which we assume that the adversary has no access.

2. It considers a restricted adversarial model, namely it assumes that the adversary performs random message attacks instead of more traditional attacks, e.g., chosen plaintext attacks.

3. Finally, the security proof has limited application for cryptographic settings.

In Chapter 9, we provide an alternative security proof in the same noisy model that overcomes these three challenges. We achieve this goal by a new reduction from noisy leakage to the important theoretical model of probing adversaries introduced by Ishai et al. [ISW03]. Our work can be viewed as a next step of closing the gap between theory and practice in leakage resilient cryptography: while our security proofs heavily rely on concepts of theoretical cryptography, we solve problems in practically motivated leakage models. This result was published in Eurocrypt 2014 [DDF14] and received one of the two best paper awards.

1.6 Towards a More Practical Interpretation of this Result

While the results presented in the previous section have a great theoretical interest, they might seem to remain far from parameters and techniques used by the more practical community. Hence, in Chapter 10, we investigate the relationships between theoretical studies of leaking cryptographic devices and concrete security evaluations with standard side-channel attacks. First, we connect the formal analysis of the masking countermeasure presented in Chapter 9 with the Eurocrypt 2009 evaluation framework [SMY09] for side-channel key recovery attacks introduced by Standaert et al. In particular, we re-state the main proof of Chapter 9 for the masking countermeasure based on a mutual information metric, which is frequently used in concrete physical security evaluations. Second, we discuss the tightness of the bounds presented in Chapter 9 based on experimental case studies. This allows us to conjecture a simplified link between

the mutual information metric and the success rate of a side-channel adversary, ignoring technical parameters and proof artifacts. These observations[2] enable significant reductions of the evaluation costs for certification bodies. This result was published in Eurocrypt 2015 [DFS15a].

[2] along with two additional contributions presented in [DFS15a] but which are outside the scope of this thesis.

Chapter 2

Preliminaries

In this chapter, we present various notations and basic definitions that will be used in further chapters. In Section 2.1, we give definitions of basic notions in cryptography like negligible functions. In Section 2.2 we define various notions of statistical distance that are going to be use throughout this thesis and especially in Chapters 9 and 10. In Section 2.3, we define formally basic security notions used in cryptography, e.g., IND-CPA security. In Section 2.4, we state core results about both discrete and continuous Fourier transforms that we are going to use in Chapters 6 and 7. Finally, in Section 2.5, we recall a Chernoff and a Hoeffding bound. We want to inform the reader that a very useful list of definitions is given at the end of this thesis.

2.1 Notations and Mathematical Preliminaries

We denote by "log" the logarithm in base two and by "ln" the natural logarithm. The concatenation of two bitstrings x and y is written $x\|y$. We consider vectors as row vectors. The transpose of a vector $\boldsymbol{v}$ is denoted by $\boldsymbol{v}^t$. Given a vector $\boldsymbol{a}$, we denote by $\boldsymbol{a}_j$ its j-th component. We write $\boldsymbol{a}^{(j)}$ to say that we access the j-th vector of a set. We let $\lceil . \rfloor : \mathbb{R} \to \mathbb{Z}$ be the rounding function that rounds to the closest integer.[1] For a predicate $\pi(x)$, we denote by $\mathbb{1}_{\{\pi(x)\}}$ the function which is 1 when $\pi(x)$ is true and 0 otherwise. We denote the Hamming weight of a bitstring x by $\mathsf{Hw}(x)$. We write $x \xleftarrow{U} \mathcal{D}$ if an element x is drawn uniformly at random in a domain $\mathcal{D}$. Finally, we sometimes write $\Pr[y] \coloneqq \Pr[Y = y]$ when clear from context.

Definition 2.1 (Negligible Function)**.** *A function $f(\lambda)$ is* negligible *if for all $d \in \mathbb{R}$ we have $f(\lambda) = O\left(\lambda^{-d}\right)$.*

We denote the Bernoulli distribution with parameter p by $\mathrm{Ber}(p)$, i.e., if $x \leftarrow \mathrm{Ber}(p)$, we have $\Pr[x = 1] = p$ and $\Pr[x = 0] = 1 - p$.

[1] In case of equality, we take the floor.

Definition 2.2 (Sequence of Bernoulli trials (S_p^n))**.** *We write S_p^n to denote the sequence of n independent Bernoulli trials with parameter p. We write $\mathrm{S}_p^n(r)$ when we need to specify the seed r used to generate this sequence.*

Notation. *Given some initial parameters Π and a predicate P, we write*

$$\Pr\left[P(v_1,\ldots,v_m;r_p):\quad \begin{array}{l} v_1 \leftarrow f_1(\Pi;r_1) \\ \vdots \\ v_m \leftarrow f_m(\Pi,v_1,\ldots,v_{m-1};r_m) \end{array}\right]$$

to denote the probability (over the randomnesses $r_1,\ldots,r_m,r_p$) that, when $v_1 \leftarrow f_1(\Pi;r_1)$, $\ldots, v_m \leftarrow f_m(\Pi,v_1,\ldots,v_m;r_m)$, then $P(v_1,\ldots,v_m;r_p)$ holds.

2.2 Statistical Distance

We define the *statistical distance* between two distributions in the following way.

Definition 2.3 (Statistical distance)**.** *Given two discrete distributions $\mathcal{D}_0$ and $\mathcal{D}_1$ over a set $\mathcal{Z}$, we define the* statistical distance *between $\mathcal{D}_0$ and $\mathcal{D}_1$ by*

$$\Delta(\mathcal{D}_0,\mathcal{D}_1) \coloneqq \frac{1}{2}\sum_{z\in\mathcal{Z}}|\mathcal{D}_1(z)-\mathcal{D}_0(z)| = \sum_{z\in\mathcal{Z}}\max\{0,\mathcal{D}_1(z)-\mathcal{D}_0(z)\}\ .$$

If $\mathcal{X},\mathcal{Y}$ are some events then by $\Delta((A|\mathcal{X})\ ;\ (B|\mathcal{Y}))$ we will mean the distance between variables A' and B', distributed according to the conditional distributions $P_{A|\mathcal{X}}$ and $P_{B|\mathcal{Y}}$. If C is a random variable then by $\Delta(A\ ;\ (B|C))$ we mean $\sum \Pr[C=c]\cdot\Delta(A\ ;\ (B|(C=c)))$.
If A, B, and C are random variables then $\Delta((B;C)\mid A)$ denotes $\Delta((B,A);(C,A))$.[2] It is easy to see that it is equal to $\sum_a \Pr[A=a]\cdot\Delta((B|A=a)\ ;\ (C|A=a))$. If $\Delta(A;B)\le\epsilon$ then we say that A and B are *ϵ-close*. The "$\stackrel{d}{=}$" symbol denotes the equality of distributions, i.e., $A\stackrel{d}{=}B$ if and only if $\Delta(A;B)=0$. We will use the following lemma.

Lemma 2.4. *Let A, B be two (possibly correlated) random variables. Let B' be a variable distributed identically to B but independent from A. We have*

$$\Delta(A;(A\mid B)) = \Delta((B;B')\mid A). \tag{2.1}$$

[2] This notation might seem counterintuitive but is used in all the previous works.

Proof. We have

$$\begin{aligned}\Delta(A;(A \mid B)) &= \sum_b \frac{1}{2} \cdot \Pr[B=b] \cdot \sum_a |\Pr[A=a] - \Pr[A=a \mid B=b]| \\ &= \frac{1}{2} \sum_{a,b} |\Pr[B=b] \cdot \Pr[A=a] - \Pr[B=b] \cdot \Pr[A=a \mid B=b]| \\ &= \frac{1}{2} \sum_{a,b} |\Pr[B'=b \wedge A=a] - \Pr[B=b \wedge A=a]| \qquad (2.2) \\ &= \Delta((B;B') \mid A),\end{aligned}$$

where in (2.2) we used the fact that B' is a variable distributed identically to B and is independent from A. □

We state now a basic lemma that will be used in Chapter 9 for some proofs.

Lemma 2.5. *For any random variables A and B and an event $\mathcal{E}$ we have*

$$\Delta(A;B) \leq \Delta((A \mid \neg\mathcal{E});B) + \Pr[\mathcal{E}],$$

where $\neg\mathcal{E}$ denotes the negation of $\mathcal{E}$.

A proof of a very similar lemma is given in [DDV10, Appendix A]

2.3 Security Notions

Definition 2.6 (Public-key Encryption Scheme)**.** *Let $\varphi(\lambda)$ be a function. A $\varphi(\lambda)$-cryptosystem over a given message space $\mathcal{M}$ and random coin space $\mathcal{R}$ consists of three* polynomial-time *algorithms:*

- *a probabilistic key-generation algorithm $\mathsf{Gen}(1^\lambda;\rho_g)$ taking as input some security parameter 1^λ in unary representation and some random coins ρ_g, and producing a secret key K_s and a public key K_p;*
- *a probabilistic encryption algorithm $\mathsf{Enc}(K_p,m;r)$ taking as input a public key K_p and a message $m \in \mathcal{M}$ with some random coins $r \in \mathcal{R}$, and producing a ciphertext y in the ciphertext space $\mathcal{C}$;*
- *a deterministic decryption algorithm $\mathsf{Dec}(K_s,c)$ taking as input a secret key K_s and a ciphertext $c \in \mathcal{C}$, and producing a message or an error.*

The cryptosystem must satisfy the following correctness property:

$$\max_{m\in\mathcal{M}} \Pr\left[\mathsf{Dec}(K_s,\mathsf{Enc}(K_p,m;\rho)) \neq m : \qquad (K_s,K_p) \leftarrow \mathsf{Gen}(1^\lambda;\rho_g)\right] \leq \varphi(\lambda)\,.$$

We will also use the following security notions and acronyms. Adaptive Chosen Ciphertext Attack is denoted CCA, Chosen Plaintext Attack CPA, Indistinguishability IND and one-wayness OW.

Definition 2.7 (IND-CPA-security)**.** *A cryptosystem is said* (t, ε)-IND-CPA-*secure or* (t, ε)-semantically secure *against chosen plaintext attacks if no adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *with running time bounded by* t *can distinguish the encryption of two different plaintexts* m_0 *and* m_1 *with a probability higher than* ε.[3] *More formally, for all* $\mathcal{A}$ *bounded by* t*,*

$$\Pr\left[\mathcal{A}_2(K_p, c; \rho) = b : \begin{array}{l}(K_s, K_p) \leftarrow \mathsf{Gen}(1^\lambda; \rho_g) \\ m_0, m_1 \leftarrow \mathcal{A}_1(K_p; \rho) \\ r \xleftarrow{U} \mathcal{R};\ b \xleftarrow{U} \{0,1\} \\ c \leftarrow \mathsf{Enc}(K_p, m_b; r)\end{array}\right] \leq \frac{1}{2} + \varepsilon\ .$$

Asymptotically, a cryptosystem is IND-CPA-*secure if for any polynomial* $t(\lambda)$ *there exists a negligible function* $\varepsilon(\lambda)$ *such that it is* $(t(\lambda), \varepsilon(\lambda))$-IND-CPA-*secure.*

IND-CPA-security can also be represented in the real-or-random game model [BDJR97b, BDJR97a].[4]

Definition 2.8 (Simple real-or-random IND-CPA game security)**.** *A cryptosystem is* (t, ε)-IND-CPA-*secure in the real-or-random game model if no adversary* $\mathcal{A}$ *with running time bounded by* t *can distinguish the encryption of a chosen plaintext* m_0 *to a random one with a probability higher than* ε*. More formally, for all* $\mathcal{A}$ *bounded by* t*,*

$$\Pr\left[\mathcal{A}_2(K_p, c; \rho) = b : \begin{array}{l}(K_s, K_p) \leftarrow \mathsf{Gen}(1^\lambda; \rho_g) \\ m_0 \leftarrow \mathcal{A}_1(K_p; \rho);\ m_1 \xleftarrow{U} \mathcal{M} \\ r \xleftarrow{U} \mathcal{R};\ b \xleftarrow{U} \{0,1\} \\ c \leftarrow \mathsf{Enc}(K_p, m_b; r)\end{array}\right] \leq \frac{1}{2} + \varepsilon\ .$$

Asymptotically, a cryptosystem is IND-CPA-*secure in the real-or-random game model if for any polynomial* $t(\lambda)$ *there exists a negligible function* $\varepsilon(\lambda)$ *such that it is* $(t(\lambda), \varepsilon(\lambda))$-IND-CPA-*secure in the real-or-random game model.*

A (t, ε)-IND-CPA-secure system in the real-or-random game model is $(t, 2\varepsilon)$-IND-CPA-secure [BDJR97a]. Conversely, a (t, ε)-IND-CPA-secure system is (t, ε)-IND-CPA-secure in the real-or-random game model. Asymptotically, both models are equivalent.

Definition 2.9 (IND-CCA-security)**.** *A cryptosystem is said* (t, ε)-IND-CCA-*secure or* (t, ε)-secure *against adaptive chosen ciphertext attacks if no adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*,*

[3] We include in the running time the size of the code of $\mathcal{A}$ in a fixed RAM model of computation to avoid trivial adversaries.

[4] In our definition of real-or-random game model, we consider only *simple* adversaries, i.e., adversaries who can query the oracle once. This definition is enough to prove the IND-CPA-security of our scheme.

with access to a decryption oracle $\mathcal{O}_{K_s}$ and with running time bounded by t can distinguish the encryption of two different plaintexts m_0 and m_1 with a probability higher than ε. More formally, for all $\mathcal{A}$ bounded by t,

$$\Pr\left[\mathcal{A}_2^{\mathcal{O}_{K_s}}(K_p, c; \rho) = b : \begin{array}{l}(K_s, K_p) \leftarrow \mathsf{Gen}(1^\lambda; \rho_g) \\ m_0, m_1 \leftarrow \mathcal{A}_1^{\mathcal{O}_{K_s}}(K_p; \rho) \\ r \xleftarrow{U} \mathcal{R};\ b \xleftarrow{U} \{0,1\} \\ c \leftarrow \mathsf{Enc}(K_p, m_b; r)\end{array}\right] \leq \frac{1}{2} + \varepsilon\ ,$$

where $\mathcal{O}_{K_s,c}(y) = \mathsf{Dec}(K_s, y)$ for $y \neq c$ and $\mathcal{O}_{K_s,c}(c) = \perp$. Asymptotically, a cryptosystem is IND-CCA*-secure if for any polynomial $t(\lambda)$ there exists a negligible function $\varepsilon(\lambda)$ such that it is $(t(\lambda), \varepsilon(\lambda))$-*IND-CCA*-secure.*

Definition 2.10 (Distinguisher)**.** *Given two distributions $\mathcal{D}_0$ and $\mathcal{D}_1$, a* distinguisher *between them is an algorithm $\mathcal{A}$ that takes as input one sample x from either $\mathcal{D}_0$ or $\mathcal{D}_1$ and has to decide which distribution was used. Its* advantage *is*

$$\mathrm{Adv}_{\mathcal{A}}(\mathcal{D}_0, \mathcal{D}_1) = \Pr\left[\mathcal{A}(x) = 1 \colon x \leftarrow \mathcal{D}_1\right] - \Pr\left[\mathcal{A}(x) = 1 \colon x \leftarrow \mathcal{D}_0\right]\ .$$

We know that for all $\mathcal{A}$, $\mathrm{Adv}_{\mathcal{A}}(\mathcal{D}_0, \mathcal{D}_1) \leq \Delta(\mathcal{D}_0, \mathcal{D}_1)$. Equality can be reached with $\mathcal{A}$ defined by $\mathcal{A}(x) = 1$ iff $\mathcal{D}_1(x) \geq \mathcal{D}_0(x)$.
We say that $\mathcal{D}_0$ and $\mathcal{D}_1$ are ϵ-statistically indistinguishable *if $\Delta(\mathcal{D}_0, \mathcal{D}_1) \leq \epsilon$.*
We say that the two distributions are (t, ε)-computationally indistinguishable *if for any distinguisher $\mathcal{A}$ with running time bounded by t,*

$$|\,\mathrm{Adv}_{\mathcal{A}}(\mathcal{D}_0, \mathcal{D}_1)| \leq \varepsilon\ .$$

Asymptotically, two distributions depending on a parameter λ are computationally indistinguishable *if for any polynomial $t(\lambda)$ there exists a negligible function $\varepsilon(\lambda)$ such that, they are $(t(\lambda), \varepsilon(\lambda))$-computationally indistinguishable.*

2.4 Fourier Transforms

Our results in Chapter 6 and 7 make extensive use of both discrete and continuous Fourier transforms. We summarize in the following some basic properties that we are going to use. We refer the reader to any book on Fourier transforms for proofs and extensions of these results. In this section, we define $\sqrt{-1} = i \in \mathbb{C}$.

2.4.1 Discrete Fourier Transform

Definition 2.11 (Discrete Fourier Transform (DFT))**.** *Let $p_1, \cdots, p_b$ be integers and let $\theta_{p_j} \coloneqq \exp(2\pi i / p_j)$, for $1 \leq j \leq b$ and where $i = \sqrt{-1}$. Define the group $G \coloneqq \mathbb{Z}_{p_1} \times \cdots \times \mathbb{Z}_{p_b}$. We may write an element $x \in G$ as $(x_1, \cdots, x_b)$. The* discrete Fourier

transform (DFT) *of a function* $f\colon G \to \mathbb{C}$ *is a function* $\widehat{f}\colon G \to \mathbb{C}$ *defined as*

$$\widehat{f}(\alpha) \coloneqq \sum_{x \in G} f(x)\theta_{p_1}^{-\alpha_1 x_1} \cdots \theta_{p_b}^{-\alpha_b x_b} \,. \tag{2.3}$$

The discrete Fourier transform can be computed in time $O\left(|G|\log(|G|)\right) =: C_{\mathsf{FFT}} \cdot |G|\log(|G|)$ for a small constant C_{FFT}.

2.4.2 Continuous Fourier Transforms

We use the following definition for continuous Fourier Transforms.

Definition 2.12 (Continuous Fourier transform (FT))**.** *The* continuous Fourier transform (FT) *of a function* $f\colon \mathbb{R} \to \mathbb{C}$ *is a function* $\mathcal{F}(f)\colon \mathbb{R} \to \mathbb{C}$ *defined as*

$$\mathcal{F}(f)(\chi) = \int_{-\infty}^{\infty} f(x)e^{-2\pi i \chi x}\,\mathrm{d}x \,. \tag{2.4}$$

We will use the following well-known properties of the FT.

Linearity.

$$\mathcal{F}(f(x) + g(x))(\chi) = (\mathcal{F}(f) + \mathcal{F}(g))(\chi) \,. \tag{2.5}$$

Translation.

$$\mathcal{F}(f(x-y))(\chi) = e^{-i2\pi y\chi}\mathcal{F}(f)(\chi) \,. \tag{2.6}$$

Convolution.

$$\mathcal{F}(f(x)g(x))(\chi) = (\mathcal{F}(f(x)) * \mathcal{F}(g(x)))(\chi) \,, \tag{2.7}$$

where $*$ denotes the convolution operator which is defined as

$$(u * v)(x) \coloneqq \int_{-\infty}^{\infty} u(y)v(x-y)\,\mathrm{d}y \,.$$

Integration.

$$\mathcal{F}\left(\int_{-\infty}^{x} f(\tau)\,\mathrm{d}\tau\right)(\chi) = \frac{1}{2i\pi\chi}\mathcal{F}(f)(\chi) + \frac{1}{2}\mathcal{F}(f)(0)\delta(\chi) \,, \tag{2.8}$$

where δ is the Dirac delta distribution. We will use the following property of the Dirac delta.

$$\int_{-\infty}^{\infty} f(\tau)\delta(\tau-\ell)\,\mathrm{d}\tau = f(\ell) \,.$$

We refer the reader to, e.g., [GS64, Rud91, Str03] for more information about the Dirac delta distribution or, e.g. [BB86] for a more engineering approach.
We will also use the Poisson summation formula.

Lemma 2.13 (Poisson summation formula (see, e.g., [SW71]))**.** *Let $f(x)\colon \mathbb{R} \to \mathbb{C}$ be a function in the Schwartz space*[5] *and $\mathcal{F}(f)$ its continuous Fourier transform then*

$$\sum_{\ell=-\infty}^{\infty} f(\ell) = \sum_{\chi=-\infty}^{\infty} \mathcal{F}(f)(\chi) \,. \tag{2.9}$$

Useful Fourier Transforms.

$$\mathcal{F}\left(\frac{1}{\sigma\sqrt{2\pi}} e^{-x^2/(2\sigma^2)}\right)(\chi) = e^{-2\pi^2\sigma^2\chi^2} \,. \tag{2.10}$$

Let $\gamma \in \mathbb{R}$. Then

$$\mathcal{F}\left(\cos(\alpha x)\right)(\chi) = \frac{1}{2}\left(\delta\left(\chi - \frac{\gamma}{2\pi}\right) + \delta\left(\chi + \frac{\gamma}{2\pi}\right)\right) \,, \tag{2.11}$$

where δ is the Dirac delta distribution.

2.5 Various Bounds

We will need the following Chernoff bound.

Lemma 2.14 (Chernoff bound)**.** *Let $Z = \sum_{i=1}^{n} Z_i$, where Z_i's are random variables independently distributed over $[0,1]$. Then for every $\xi > 0$ we have*

$$\Pr[Z \geq (1+\xi)\mathbb{E}[Z]] \leq \exp\left(-\frac{\xi^2}{3}\mathbb{E}[Z]\right) \,.$$

We will use also the following Hoeffding bounds.

Theorem 2.15 ([Hoe63])**.** *Let $X_1, X_2, \ldots, X_n$ be n independent random variables with $\alpha_j \leq X_j \leq \beta_j$ for $1 \leq j \leq n$. Let $X \coloneqq X_1 + \ldots + X_n$ and let $\mathbb{E}[X]$ be the expected value of X. We have that*

$$\Pr[X - \mathbb{E}[X] \geq t] \leq \exp\left(\frac{-2t^2}{\sum_{j=1}^{n}(\beta_j - \alpha_j)^2}\right)$$

[5] A function $f(x)$ is in the Schwartz space if $\forall \alpha, \beta \in \mathbb{N}, \exists C_{\alpha,\beta}$ such that $\sup|x^\alpha \partial_x^\beta f(x)| \leq C_{\alpha,\beta}$. A function in C^∞ with compact support is in the Schwartz space.

and

$$\Pr[X - \mathbb{E}[X] \leq -t] \leq \exp\left(\frac{-2t^2}{\sum_{j=1}^{n}(\beta_j - \alpha_j)^2}\right),$$

for any $t > 0$.

Part I

Post-quantum Cryptography

Chapter 3

Hard Problems in Post-Quantum Cryptography

Every public-key cryptosystem relies on problems that are believed computationally hard. The two mostly used problems are the integer factorization problem [RSA78, Rab79] and the discrete logarithm problem [EG84]. However, these two problems can be solved in polynomial time on a quantum computer. It is thus important to develop new cryptosystems that are secure even on quantum computers and to correctly propose some parameters depending on the required security.
In this chapter, we list some hard problems used in cryptography that are believed to be post-quantum, i.e., hard to solve even on a quantum computer. We do not try to be exhaustive and we list only the problem we will use later in our results.

3.1 The Learning from Parity with Noise Problem

The *Learning from Parity with Noise* (LPN) problem[1] has been well studied both in learning theory and in cryptography. The hardness of this problem was first discussed by Kearns in STOC 1993 [Kea93]. The same year, it was used for the first time in cryptography by Blum et al. [BFKL93]. The goal of this problem is to find out an unknown vector $\boldsymbol{u}$, given some noisy versions of its scalar product with some known random vector. More formally

Definition 3.1 (LPN Oracle)**.** *An* LPN oracle $\Pi_{\boldsymbol{u},p}$ *for a hidden vector* $\boldsymbol{u} \in \{0,1\}^k$ *and* $0 < p < \frac{1}{2}$ *is an oracle returning an LPN vector, i.e., vectors of the form*

$$\langle \boldsymbol{a} \xleftarrow{U} \{0,1\}^k , \langle \boldsymbol{a}, \boldsymbol{u} \rangle \oplus \nu \rangle ,$$

where, $\nu \leftarrow \mathrm{Ber}(p)$*. Note that the output is a* $k+1$*-bit vector.*

[1] This problem is also called the Learning Parity with Noise problem in some papers.

Definition 3.2 (Search Learning from Parity with Noise Problem (LPN))**.** *The (k,p)-Learning from Parity with Noise Problem ((k,p)-LPN) consists, given an LPN Oracle $\Pi_{\boldsymbol{u},p}$, to recover the hidden vector $\boldsymbol{u}$.*
We say that an algorithm $\mathcal{A}$ (t,n,δ)-solves the (k,p)-LPN problem if $\mathcal{A}$ runs in time at most t, makes at most n oracle queries and

$$\Pr\left[\boldsymbol{u} \xleftarrow{U} \{0,1\}^k \,:\, \mathcal{A}^{\Pi_{\boldsymbol{u},p}}(1^k) = \boldsymbol{u}\right] \geq \delta \,.$$

3.1.1 The Decisional LPN Problem.

The LPN problem has also a decisional form.

Definition 3.3 (Decisional LPN Problem (D-LPN))**.** *Let U_{k+1} be an oracle returning random $k+1$-bit vectors. Then, an algorithm $\mathcal{A}$ (t,n,δ)-solves the* (k,p)-decisional LPN problem *(D-LPN) if $\mathcal{A}$ runs in time at most t, makes at most n oracle queries and*

$$\left|\Pr\left[\boldsymbol{u} \xleftarrow{U} \{0,1\}^k \,:\, \mathcal{A}^{\Pi_{\boldsymbol{u},p}}(1^k) = 1\right] - \Pr\left[\mathcal{A}^{U_{k+1}}(1^k) = 1\right]\right| \geq \delta \,.$$

It is shown in [KS06, Reg05] that if there exists an algorithm $\mathcal{A}$ that (t,n,δ)-solves the (k,p)-D-LPN problem, then there is an algorithm $\mathcal{A}'$ that $(t',n',\delta/4)$-solves the (k,p)-LPN problem, with $t' := O\left(t \cdot k\delta^{-2}\log k\right)$ and $n' := O\left(n \cdot \delta^{-2}\log k\right)$. Thus, the hardness of the LPN problem implies that the output of the LPN vector oracle is indistinguishable from a random source.
We say that the (k,p)-D-LPN problem is (t,ϵ)-hard, if there is no known algorithm that solves it with running time bounded by t and advantage higher than ϵ.

3.1.2 Algorithms that Solve the LPN Problem

The first subexponential algorithm to solve the LPN problem was given by Blum, Kalai, and Wasserman in [BKW03] and they estimated its complexity to $2^{O(k/\log k)}$. We name this algorithm the BKW algorithm.
The idea of the BKW algorithm is to first query the LPN oracle to obtain a large amount of LPN vectors. It searches then for basis vectors e_j by finding a low amount of vectors that xor to e_j. If the number of vectors that xor to e_j is small, the noise for this vector will be small as well. Using different independent instances that xor to the same e_j, one can recover the jth bit of $\boldsymbol{u}$ with good probability. All this procedure requires using a large amount of queries. We will devote a whole chapter (Chapter 4) to the BKW algorithm as we will show how to apply it to the learning with error problem (LWE), an extension of the LPN problem. The LWE problem will be formally defined in Section 3.3.
The BKW algorithm (when applied to LPN) was analyzed in detail and improved in [LF06, FMI+06, BL12, GJL14]. Very recently, Bogos, Tramèr, and Vaudenay published a detailed survey regarding the BKW problem when applied on the LPN problem [BTV15]. In this survey, Bogos et al. show tighter bounds for all the previously existing algorithms. We are going to give here their modified bound of the Levieil and

Fouque algorithm [LF06] that we will use as a security bound in the cryptosystem we suggest in Chapter 5. As the hardness of the LPN problem is far from being a bottleneck in our proposal, we will use this generic bound to find practical parameters. Note that the bounds from [GJL14] might be slightly better but would require us to find a covering code that suits our application.

Theorem 3.4 ([BTV15, Theorem 5], modified bound from [LF06])**.** *For $a, b \in \mathbb{N}$, two parameters such that $ab \geq k$. Let $\theta \in (0, 1]$ be the probability of success of the algorithm. Let $q \coloneqq 8\ln(2^b/\theta)(1-2p)^{-2^a} + (a-1)2^b$ and let $t \coloneqq kaq + b2^b$. There exists an algorithm that (heuristically) (t, q, θ)-solves the (k, p)-LPN problem.*

Some parameters along with their security are given in [BTV15]. This algorithm requires a subexponential (in k) number of samples.

3.1.3 Solving LPN with a Polynomial Number of Samples

When the number of samples is polynomial (as it will be in Chapter 5), Lyubashevsky showed how one can use a universal family of hash functions and the leftover hash lemma to obtain more samples under the conditions that they are under a higher noise level [Lyu05].

Definition 3.5 (Universal family of hash functions)**.** *Let $\mathcal{H}$ be a set of functions from a set X to a set Y. Let $H \xleftarrow{U} \mathcal{H}$. $\mathcal{H}$ is a* universal family of hash functions *if for all $x_1, x_2 \in X$, $x_1 \neq x_2$,*

$$\Pr[H(x_1) = H(x_2)] \leq \frac{1}{|Y|} .$$

Lemma 3.6 (Leftover hash lemma [IZ89])**.** *Let $\ell, e \in \mathbb{N}$ be parameters. Let $X, Y \subseteq \{0,1\}^k$ be two sets such that $|X| \geq 2^\ell$ and $|Y| = \{0,1\}^{\ell-2e}$. Let U be the uniform distribution over the set Y and let $\mathcal{H}$ be a universal family of hash functions from X to Y. Then, there exists $H \subseteq \mathcal{H}$ with $|H| = (1 - 2^{-e/2})|\mathcal{H}|$ such that for any $h \in H$, and $x \xleftarrow{U} X$, $\Delta(h(x), U) \leq 2^{-e/2}$.*

It is a well-known result (and shown in [Lyu05]) that the family of hash function $\mathcal{H}$ from a set X to a set Y with $X \subseteq \{0,1\}^{k^{1+\epsilon}}$, $|X| \geq 2^{2k}$ and $Y = \{0,1\}^k$ defined as $\mathcal{H} \coloneqq \left\{h_a | a = (a_1, \ldots, a_{k^{1+\epsilon}}), a_i \in \{0,1\}^k\right\}$ with $h_a(x) = x_1 a_1 \oplus \cdots \oplus x_{k^{1+\epsilon}} a_{k^{1+\epsilon}}$. In his paper, Lyubashevsky applies the leftover hash lemma to obtain more queries. For this, he combines queries using $\mathcal{H}$ with X the set of all bit strings with exactly $\lceil 2k/(\epsilon \log k)\rceil$ ones. Letting a be the set of all existing queries and Lemma 3.6 shows that the resulting queries are now close to uniformly distributed.
Lyubashevsky shows that the LPN problem with a polynomial number of samples is solved asymptotically in $2^{O(k/\log\log k)}$. More precisely, one can transform the (k, p)-LPN problem with $k^{1+\epsilon}$ samples in the (k, p')-LPN problem with enough samples to use the

BKW algorithm and with

$$p' = \frac{1}{2} - \frac{1}{2}\left(\frac{1}{4} - \frac{p}{2}\right)^{\left\lceil \frac{2k}{\epsilon \log k} \right\rceil} . \tag{3.1}$$

This new noise can easily be explained as we combine exactly $\lceil 2k/(\epsilon \log k) \rceil$ queries together with the universal hash function. Combining Lyubashevsky's result with Theorem 3.4, we get the following time complexity ($\mathsf{T}_{\mathsf{LPN}}$) for solving LPN and we will use it as a security bound in our cryptosystem.

Theorem 3.7 (LPN with limited number of queries)**.** *For $a, b \in \mathbb{N}$, two parameters such that $ab \geq k$. Let $\theta \in (0, 1]$ be the probability of success of the algorithm. Let q be the maximum number of queries allowed. Let $\epsilon > 0$ be such that $q = k^{1+\epsilon}$. Let $p' = 1/2 - 1/2(1/4 - p/2)^{\lceil 2k/(\epsilon \log k) \rceil}$ as in* (3.1)*. Let*

$$\mathsf{T}_{\mathsf{LPN}} \coloneqq \min_{0<a\leq k} \left(k \times a \times \left(8 \ln \left(\frac{2^{k/a}}{\theta} \right) (1 - 2p')^{-2^a} + (a-1)2^{k/a} \right) + \frac{k}{a} 2^{k/a} \right) . \tag{3.2}$$

There exists an algorithm that $(\mathsf{T}_{\mathsf{LPN}}, q, \theta)$-solves the (k, p)-LPN problem.

3.2 Finding a Low-weight Codeword in a Random Linear Code

In our security proof, we will also need to bound the complexity of finding a low-weight parity-check equation in a random linear code which is the same as finding a low-weight codeword in the dual code. This problem of finding a low-weight codeword is also called the minimum distance problem.

Definition 3.8 (Minimum Distance Problem (MDP))**.** *The* (n, k, w)-decisional minimum distance problem *is the following. Given an $(n-k) \times n$ matrix H drawn uniformly and given $w \in \mathbb{N}, w \geq 0$, is there a* non-zero $\boldsymbol{x} \in \mathbb{F}_2^n$ *with $\mathsf{Hw}(\boldsymbol{x}) \leq w$ such that $\boldsymbol{x}H^t = \mathbf{0}$? The computation counterpart of this problem consists in finding such an x.*

Its hardness remained open for a long time. It was even set the "open problem of the month" in [Joh82]. It was finally shown to be NP-hard by Vardy [Var97] using a reduction from the decisional syndrome decoding problem. Many algorithms solving this problem were developed (e.g. [LB88, Ste88, CC94, CC98, CS98, FS09].)
Finally, a general lower-bound on the complexity of the information set decoding algorithm was derived by Finiasz and Sendrier [FS09] using idealized algorithms. However, it was shown in [BLP11, MMT11, BJMM12] that it is possible to do better than this bound.
A new lower-bound for information set decoding is proposed in [BLP11]. This bound is much simpler and we give it in Assumption 3.9.

Assumption 3.9 ([BLP11])**.** *Let $r \coloneqq n - k$. Given an $[n, k]$-code and given a weight w, if $\binom{n}{w} \leq 2^r$, the cost of finding a parity-check equation of weight w is lower-bounded by*

$$\mathsf{T}_{\mathsf{MDP}}(w, n, k) \coloneqq \min_i \frac{\binom{n}{w}}{2\binom{k}{w-i}\sqrt{\binom{r}{i}}} , \tag{3.3}$$

bit operations.

We will assume this lower-bound for our cryptosystem. Note that a similar analysis for linear codes over a general field $\mathbb{F}_q$ is presented in [Pet10].

3.3 The Learning With Error Problem

The Learning With Error problem (LWE) was introduced by Regev in [Reg05] and can be seen as an extension of the Learning (from) Parity with Noise problem (LPN). Roughly, the adversary is given queries from an LWE oracle, which returns uniformly random vectors $\boldsymbol{a}_j$ in $\mathbb{Z}_q$ and their inner-product with a fixed secret vector $\boldsymbol{s} \in \mathbb{Z}_q^k$ to which some noise was added (typically some discrete Gaussian noise). The goal of the adversary is then to recover the secret $\boldsymbol{s}$. In LPN, $q = 2$ and the noise follows a Bernoulli distribution. In his seminal paper [Reg09], Regev shows a quantum reduction from some well-known Lattice problems like the decisional shortest vector problem (Gap-SVP) or the short independent vector problem (SIVP) to the LWE problem. Later, Peikert and Brakerski et al. showed how to make this reduction classical [BLP+13, Pei09]. The LWE problem was then used to design a wide range of cryptographic primitives. For instance, Gentry et al. showed how to construct a trapdoor function based on LWE and created an identity-based cryptosystem [GPV08]. Applebaum et al. used LWE to design encryption schemes with strong security properties [ACPS09]. However, the biggest breakthrough regarding LWE is its use in the design of (fully) homomorphic encryption schemes (FHE). FHE was first introduced by Gentry in his PhD thesis [Gen09]. While the initial construction was not using the LWE problem, most of the recent designs are, e.g., [BV11a, Bra12, GSW13].
We give now a formal definition of the LWE problem.

Definition 3.10 (LWE Oracle)**.** *Let k, q be positive integers. A* Learning with Error (LWE) *oracle $\Pi_{\boldsymbol{s},\chi}$ for a hidden vector $\boldsymbol{s} \in \mathbb{Z}_q^k$ and a probability distribution χ over $\mathbb{Z}_q$ is an oracle returning*

$$\left(\boldsymbol{a} \xleftarrow{U} \mathbb{Z}_q^k \ , \ \langle \boldsymbol{a}, \boldsymbol{s} \rangle + \nu\right) ,$$

where $\nu \leftarrow \chi$.

Definition 3.11 (Search-LWE)**.** *The* Search-LWE *problem is the problem of recovering the hidden secret $\boldsymbol{s}$ given n queries $(\boldsymbol{a}^{(j)}, c^{(j)}) \in \mathbb{Z}_q^k \times \mathbb{Z}_q$ obtained from $\Pi_{\boldsymbol{s},\chi}$.*

In typical schemes based on LWE, the parameter q is taken to be polynomial in k, and χ follows a discretized Gaussian distribution (see next section).
Like for LPN, one can define analogously a decisional version of the LWE problem, but we will not use this problem in this thesis.

3.3.1 Gaussian Distributions

Let $\mathcal{N}(0, \sigma^2)$ denote the Gaussian distribution of mean 0 and standard deviation σ. We denote its probability density function by $\phi \mapsto p(\phi; \sigma)$, for $\phi \in \mathbb{R}$. Consider the *wrapped* Gaussian distribution $\Psi_{\sigma,q}$ resulting from wrapping the Gaussian distribution around a circle of circumference $q > 0$. Its probability density function $g(\theta; \sigma, q)$ is given by

$$g(\theta; \sigma, q) \coloneqq \sum_{\ell=-\infty}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} \exp\left[\frac{-(\theta + \ell q)^2)}{2\sigma^2}\right], \quad \text{for} \quad \theta \in \left]-\frac{q}{2}, \frac{q}{2}\right] \ . \tag{3.4}$$

Note that $\Psi_{\sigma,2\pi}$ is the standard wrapped normal distribution obtained by wrapping $\mathcal{N}(0, \sigma^2)$ around the unit circle, used, for instance, in directional statistics [MJ09].
LWE schemes use a discretization of a Gaussian over $\mathbb{Z}_q$. There are two variants of LWE that we will consider in this thesis. We will see that we obtain similar results for both distributions. In the initial version by Regev [Reg09], the noise in LWE was a *rounded Gaussian distribution*. This is also what is considered in [ACF+13, GKPV10]. Such a distribution can be obtained by sampling from $\Psi_{\sigma,q}$ and rounding the result to the nearest integer in the interval $]\frac{-q}{2}, \frac{q}{2}]$. We denote this distribution by $\bar{\Psi}_{\sigma,q}$.

Definition 3.12 (Rounded Gaussian distribution ($\bar{\Psi}_{\sigma,q}$))**.** *The probability mass function of a* Rounded Gaussian distribution *is given by*

$$\Pr[x \leftarrow \bar{\Psi}_{\sigma,q}] = \int_{x-\frac{1}{2}}^{x+\frac{1}{2}} g(\theta; \sigma, q)\, \mathrm{d}\theta \ , \tag{3.5}$$

for x an integer in the interval $]\frac{-q}{2}, \frac{q}{2}]$ and where $g(\theta; \sigma, q)$ is given in (3.4)*.*

The LWE problem is believed to be hard when $\sigma \geq \sqrt{k}$ and $q \in \mathrm{Poly}(k)$.
The second Gaussian distribution used for LWE is the *discrete Gaussian distribution* $D_{\sigma,q}$. This distribution is used in most of the applications and in the classical LWE reduction [BLP+13]. We denote this distribution by $D_{\sigma,q}$.

Definition 3.13 (Discrete Gaussian distribution ($D_{\sigma,q}$))**.** *For x an integer in $]-\frac{q}{2}, \frac{q}{2}]$, the* discrete Gaussian distribution $D_{\sigma,q}$ *is defined as*

$$\Pr[x \leftarrow D_{\sigma,q}] = \frac{\exp(-x^2/(2\sigma^2))}{\sum_{y \in]-\frac{q}{2}, \frac{q}{2}]} \exp(-y^2/(2\sigma^2))} \ . \tag{3.6}$$

3.4 The Learning With Rounding Problem

The Learning With Rounding problem (LWR) was introduced by Banerjee, Peikert, and Rosen to construct pseudorandom functions [BPR12]. LWR can be seen as a derandomization of LWE where the random noise is replaced by a rounding modulo $p < q$. This rounding introduces a deterministic error which makes the problem hard to solve. Banerjee et al. showed that LWR is at least as hard as the LWE problem, when $q/p = k^{\omega(1)}$, where k is the length of the secret. The LWR problem was later revisited by Alwen et al. to get rid of this super-polynomial blowup [AKPW13]. However, the number of LWR samples given to the adversary is limited in this case. LWR finds new applications every year. Among them, there is the design of pseudorandom functions [BPR12], lossy trapdoor functions and reusable extractors [AKPW13], or key-homomorphic PRFs [BLMR13].
We formalize now the LWR problem.

Definition 3.14 (Rounding function ($\lceil\cdot\rfloor_p$))**.** *Let $q \geq p \geq 2$ be positive integers. The LWR problem uses the rounding function from $\mathbb{Z}_q = \{0, \ldots, q-1\}$ to $\mathbb{Z}_p = \{0, \ldots, p-1\}$, given by*[2]

$$\lceil\cdot\rfloor_p \ : \ \mathbb{Z}_q \to \mathbb{Z}_p \ : \ x \mapsto \left\lceil \left(\frac{p}{q}\right) \cdot x \right\rfloor ,$$

where the operations are done over $\mathbb{R}$.

Definition 3.15 (LWR Oracle)**.** *Let k and $q \geq p \geq 2$ be positive integers. A* Learning with Rounding (LWR) *oracle $\Lambda_{\boldsymbol{s},p}$ for a hidden vector $\boldsymbol{s} \in \mathbb{Z}_q^k$, $\boldsymbol{s} \neq 0$ is an oracle returning*

$$\left(\boldsymbol{a} \xleftarrow{U} \mathbb{Z}_q^k \ , \ \lceil\langle \boldsymbol{a}, \boldsymbol{s}\rangle\rfloor_p\right) .$$

Definition 3.16 (Search-LWR)**.** *The* Search-LWR *problem is the problem of recovering the hidden secret $\boldsymbol{s}$ given n queries $(\boldsymbol{a}^{(j)}, c^{(j)}) \in \mathbb{Z}_q^k \times \mathbb{Z}_p$ obtained from $\Lambda_{\boldsymbol{s},p}$.*

Two main reductions from LWE to LWR exist: one with exponential parameters and another with a limited number of samples.

Theorem 3.17 (Theorem 3.2 in [BPR12])**.** *Let $\beta \in \mathbb{R}_+$ and let χ be any efficiently sampleable distribution over $\mathbb{Z}$ such that $\Pr_{x \leftarrow \chi}[|x| > \beta]$ is negligible. Let $q \geq p \cdot \beta \cdot k^{\omega(1)}$. Then, solving decision-LWR with secrets of size k and parameters p and q is at least as hard as solving decision-LWE over $\mathbb{Z}_q$ with secret of size k and noise distribution χ.*

The second result reduces this explosion in the parameters but limits the number of samples the adversary is allowed to get from the LWR oracle.

[2] For the second component returned by the LWR oracle, we decided to return the rounding of $\langle \boldsymbol{a}, \boldsymbol{s}\rangle$ instead of the usual $\lfloor\langle \boldsymbol{a}, \boldsymbol{s}\rangle\rfloor$. The problem is equivalent (see, e.g., [AKPW13]). However, if we would use the floor operation, the noise in Lemma 7.1 would not have zero mean but mean $(1/2 - \gcd(p, q)/2q)$ and we would have to introduce tedious correcting terms in (7.3).

Theorem 3.18 (Theorem 4.1 from [AKPW13]). *Let λ be the security parameter. Let k, ℓ, m, p, γ be positive integers, $p_{\max}$ be the largest prime divisor of q, and $p_{\max} \geq 2\beta\gamma kmp$. Let χ be a probability distribution over $\mathbb{Z}$ such that $\mathbb{E}[|\chi|] \leq \beta$. Then, if $k \geq (\ell + \lambda + 1)\log(q)/\log(2\gamma) + 2\lambda$ and if $\gcd(q, q/p_{\max}) = 1$, the decision-LWR with secret of size k, parameters p and q and limited to m queries is at least as hard as solving decision-LWE over $\mathbb{Z}_q$ with secrets of size ℓ, noise distribution χ and limited to m queries.*

Chapter 4

The BKW Algorithm for LWE

The BKW algorithm [BKW03], introduced by Blum et al., was the first sub-exponential algorithm given for solving the *Learning Parity with Noise* (LPN) problem. Asymptotically, it has a time and sample complexity of $2^{O(k/\log k)}$. As LPN can be seen as a special case of LWE where we work over $\mathbb{Z}_2$, the BKW algorithm can be adapted to solve Search-LWE over $\mathbb{Z}_q$ with an asymptotic sample and time complexity of $q^{O(k/\log(k))} = 2^{O(k)}$ when the modulus q is polynomial in k [ACF+13, Reg09, Reg10].
The BKW algorithm can be described as a variant of the standard Gaussian elimination procedure, where a row addition results in the elimination of a whole block of elements instead of a single element. The main idea is that by using 'few' row additions and no row multiplications, we limit the size of the noise at the end of the reduction, allowing us to recover a small number of elements of $\boldsymbol{s}$ with high probability through maximum likelihood. The main complexity drawback of the algorithm comes from finding samples colliding on a block of elements such that their addition eliminates multiple elements at once.
The BKW algorithm takes two integer parameters, usually denoted a and b, such that $a = \lceil k/b \rceil$. The algorithm repeatedly eliminates blocks of up to b elements per row addition, over a rounds, to obtain the samples used for recovering elements of $\boldsymbol{s}$. Minimizing the complexity of the algorithm requires a tradeoff between the two parameters. For small a, the reduced samples have low noise and the complexity of recovering elements of $\boldsymbol{s}$ with high probability is reduced. For large b however, the complexity of finding colliding samples increases.
In [ACF+13], Albrecht et al. view the BKW algorithm as a linear system solving algorithm consisting of three stages, denoted *sample reduction*, *hypothesis testing* and *back substitution*. For convenience, we briefly describe each of these stages below.

4.1 Sample Reduction

Given an LWE oracle $\Pi_{\boldsymbol{s},\chi}$, the goal of this stage is to construct a series of oracles $\mathcal{A}_{\boldsymbol{s},\chi,\ell}$, each of which produces samples $(\boldsymbol{a}, c)$, where the first $b \cdot \ell$ elements of $\boldsymbol{a}$ are zero. To

create the oracles $\mathcal{A}_{\boldsymbol{s},\chi,\ell}$ for $0 < \ell < a$, Albrecht et al. make use of a set of tables T^ℓ, which are maintained throughout the execution of the algorithm. To sample from $\mathcal{A}_{\boldsymbol{s},\chi,1}$, we query the oracle $\mathcal{A}_{\boldsymbol{s},\chi,0}$ (which is the original LWE oracle) to obtain samples $(\boldsymbol{a}, c)$ to be stored in table T^1. However, if T^1 already contains a sample $(\boldsymbol{a}', c')$ such that $\boldsymbol{a}$ and $\pm\boldsymbol{a}'$ agree on their first b coordinates, we do not store $(\boldsymbol{a}, c)$ but instead output $(\boldsymbol{a} \mp \boldsymbol{a}', c \mp c')$. If a sample from $\mathcal{A}_{\boldsymbol{s},\chi,0}$ already has its first b elements to be zero, we directly output it as a sample from $\mathcal{A}_{\boldsymbol{s},\chi,1}$.
For $1 < \ell < a$, we proceed recursively by populating a table T^ℓ of non-zero samples from $\mathcal{A}_{\boldsymbol{s},\chi,\ell-1}$ and outputting a query as soon as we get a collision in the table.

Exploiting the symmetry of $\mathbb{Z}_q$ and the fact that we do not need to store queries which are already all-zero on a block, a table T^ℓ contains at most $(q^b-1)/2$ samples. Then, to create m samples from $\mathcal{A}_{\boldsymbol{s},\chi,\ell}$, we will need at most $m + \frac{q^b-1}{2}$ calls to $\mathcal{A}_{\boldsymbol{s},\chi,\ell-1}$. Furthermore, since there is no use in storing the zero elements from reduced samples, table T^ℓ stores samples of size $n - (\ell-1)\cdot b + 1$ elements from $\mathbb{Z}_q$. The description of the oracles $\mathcal{A}_{\boldsymbol{s},\chi,\ell}$ is given in Algorithm 4.1.
In the original BKW algorithm (see [BKW03, LF06]), one would then take samples from $\mathcal{A}_{\boldsymbol{s},\chi,a-1}$, i.e., samples with zeros everywhere except in the first b positions, and delete any sample $(\boldsymbol{a}, c)$ with more than one non-zero coordinate $\boldsymbol{a}_i$. The remaining samples would be used to recover one bit of $\boldsymbol{s}$ at a time.
Albrecht et al. generalized a bit this result. First, they select a parameter $d \leq k-(a-1)b$ which will define the number of non-zero positions we want to keep in the last iteration.[1] Instead of keeping an element only if it has a single non-zero coordinate, they do a final reduction using a new oracle $\mathcal{A}_{\boldsymbol{s},\chi,a}$. This oracle follows exactly the same algorithm as before except that its goal is to obtain vectors that have exactly d non-zero entries in $\boldsymbol{a}$. Like the previous oracles, the oracle $\mathcal{A}_{\boldsymbol{s},\chi,a}$ makes use of a final table T^a. It samples from $\mathcal{A}_{\boldsymbol{s},\chi,a-1}$ and adds (or subtracts) queries $(\boldsymbol{a}, c)$, $(\boldsymbol{a}', c')$, for which $\boldsymbol{a}$ and $\pm\boldsymbol{a}'$ agree on coordinates $(a-1)\cdot b+1$ through $k-d-1$. Albrecht et al. note that, in practice, the obtain the best results when choosing d equal to 1 or 2. Note that $d = 1$ corresponds to the BKW algorithm.[2]

4.2 Hypothesis Testing

After the reduction phase, Albrecht et al. are left with samples $(\boldsymbol{a}, c)$ from $\mathcal{A}_{\boldsymbol{s},\chi,a}$, where $\boldsymbol{a}$ has non-zero elements on d positions. We can view $\mathcal{A}_{\boldsymbol{s},\chi,a}$ as outputting samples in $\mathbb{Z}_q^d \times \mathbb{Z}_q$. Let $\boldsymbol{s}'$ denote the d first elements of $\boldsymbol{s}$. Since $\boldsymbol{a}$ was obtained by summing or subtracting up to 2^a samples from the LWE oracle $\Pi_{\boldsymbol{s},\chi}$ (and considering the fact that χ is symmetric around 0), the noise $(c - \langle \boldsymbol{a}, \boldsymbol{s}' \rangle)$ of the reduced samples follows the distribution of the sum of 2^a noise samples. The problem of recovering $\boldsymbol{s}'$ can then be

[1] If $k = ab$, then $d \leq b$.
[2] The only difference between the two algorithms is that the original BKW algorithm restarts every time $\mathcal{A}_{\boldsymbol{s},\chi,a}$ outputs something.

Algorithm 4.1 Oracle $\mathcal{A}_{\boldsymbol{s},\chi,\ell}$, for $0 < \ell < a$

State: A table T^ℓ (initially empty)
Output: An LWE tuple $(\boldsymbol{a}, c)$ such that $\boldsymbol{a}$ has the first $b \cdot \ell$ elements set to 0.

loop
  Let $(\boldsymbol{a}, c) \leftarrow \mathcal{A}_{\boldsymbol{s},\chi,\ell-1}$.
  if $\boldsymbol{a}$ has the first $b \cdot \ell$ elements set to 0 **then**
    return $(\boldsymbol{a}, c)$.
  end if
  if there is $(\boldsymbol{a}', c') \in T^\ell$ such that $\boldsymbol{a}$ and $\pm\boldsymbol{a}'$ are equal on the first $b \cdot \ell$ positions **then**
    return $(\boldsymbol{a} \mp \boldsymbol{a}', c \mp c')$
  end if
  Insert $(\boldsymbol{a}, c)$ in T^ℓ.
end loop

seen as a problem of distinguishing between the noise distributions for $\boldsymbol{s}'$ and $\boldsymbol{v} \neq \boldsymbol{s}'$.
By performing an exhaustive search over $\mathbb{Z}_q^d$ and making use of the log-likelihood ratio, Albrecht et al. determine the number m of samples from $\mathcal{A}_{\boldsymbol{s},\chi,a}$ which should be required to recover $\boldsymbol{s}'$ with high enough probability.
As already mentioned, the analysis of the solving phase from [ACF+13] makes use of the heuristic assumption that the noise contributions of the samples from $\mathcal{A}_{\boldsymbol{s},\chi,a}$ are independent and that the sum of rounded Gaussians also follows a rounded Gaussian distribution.

4.3 Back Substitution

This stage was not part of the original BKW algorithm for LPN [BKW03, LF06] (which does not make use of the set of tables T defined previously either). It is analogous to the back substitution typically used in Gaussian elimination and is a clever way of reducing the size of the LWE problem after part of the secret $\boldsymbol{s}$ has been recovered.
Indeed, once d elements of $\boldsymbol{s}$ are recovered with high probability, we can perform a back substitution over the set of tables T, zeroing-out d elements in each sample. To recover the next d elements from $\boldsymbol{s}$, we query m new samples from $\Pi_{\boldsymbol{s},\chi}$ and reduce them through the tables T (which are already filled) to obtain samples for hypothesis testing. Note that as soon as we recover all the bits at positions $(\ell - 1) \cdot b$ through $\ell \cdot b - 1$, the oracle $\mathcal{A}_{\boldsymbol{s},\chi,\ell}$ and its corresponding table T^ℓ become superfluous and further samples will need one reduction phase less.

4.4 The LF1 Algorithm

In [LF06], Levieil and Fouque propose an optimisation of the BKW algorithm for LPN, denoted LF1, which recovers a full block of b bits of the secret $\boldsymbol{s}$ at once by cleverly

applying a Walsh-Hadamard transform.
Compared to the original BKW algorithm, their method has the advantage of making use of all the available samples after reduction, instead of having to discard those with more than one non-zero position. Instead of an exhaustive search of complexity of the order $O\left(2^b \cdot m\right)$ (where m is the number of samples left after reduction), they use a fast Walsh-Hadamard transform to recover the most likely secret block in time $O\left(m + b2^b\right)$. The analysis of their algorithm shows that it clearly outperforms the standard BKW, although their asymptotic complexities are the same. Note that the LF1 algorithm uses the exact same reduction phase as the original BKW. This is exactly the algorithm we used to obtain the complexity in Theorem 3.4. We will use a similar idea in Chapter 6 to solve the LWE problem.

Chapter 5

The HELEN Cryptosystem

The work presented in this chapter is a joint work with Prof. S. Vaudenay and was published in [DV13a] and in [DV12] as an extended abstract. However, compared to these two papers, we greatly simplify the security proof in this thesis. In this chapter, we present HELEN, a public-key cryptosystem, the security of which relies on the hardness of the LPN (see Section 3.1) and the *minimum distance problem* (see Section 3.2).[1] Algorithms solving these problems were surveyed in Chapter 3. Note that there is also no known polynomial-time algorithm on quantum computers. In short, the keys in HELEN consists in a low-weight parity check equation h (the private key) which is hidden in a random matrix G (the public key) such that it is indistinguishable from a totally random matrix. The matrix G spans a linear code. Our cryptosystem has some similarities with the Alekhnovich cryptosystem [Ale03]. However, we carefully study its complexity, we further suggest concrete and optimized parameters, and we make incorrectness small.
We encrypt a duplicated bit by hiding it using a random linear codeword as well as a random biased noise vector. For decryption, the random linear codeword is removed by multiplying the ciphertext with h. The noise is removed by majority logic decoding. With a proper parameter choice, the probability of decrypting erroneously the message is small. We show in a further section how to reduce this probability of error as well as how to encrypt multiple bits at the same time using HELEN.

5.1 Related Work

The LPN problem is well studied in the cryptographic community. There is an authentication protocol based on the LPN problem named HB by Hopper and Blum [HB01]. This protocol was later improved into the HB$^+$ protocol by Juels and Weis [JW05]. However, HB$^+$ was shown vulnerable to man-in-the-middle attacks [GRS05]. Several variants were suggested [BCD06, DK07, MP07] but all of them suffer from the same vulnerability [GRS08a]. A new variant HB$^\#$ was introduced by Gilbert, Robshaw and

[1] HELEN stands for Hidden Equation for Linear Encryption with Noise.

Seurin [GRS08b] to improve the transmission cost of the protocol and its securtiy against man-in-the-middle attacks but an attack was also found in this variant [OOV08]. Two more recent versions were introduced based on the hardness of some variant of the LPN problem, namely Ring-LPN [HKL$^+$11, HKL$^+$12] and subspace LPN [KPC$^+$11].

Among other work based on the LPN problem, a PRNG is presented by Blum et al. in [BFKL93] along with a one-way function and a private-key encryption scheme based on some hard learning problems. A private-key encryption scheme named LPN-C was suggested by Gilbert, Robshaw and Seurin [GRS08c]. LPN-C was shown IND-CPA secure.

The construction of HELEN presents some similarities with the trapdoor cipher TCHo by Aumasson et al. [DV13b, AFMV07, FV06] which similarly encrypts a message by adding some random biased noise and some contribution from a linear code. In TCHo, this noise is introduced using an LFSR whose feedback polynomial has a multiple of low weight.

A class of lattice-based cryptosystems introduced by Regev is based on the worst-case complexity of the *learning with errors* (LWE) problem [Reg05, Pei09, LPR10, SSTX09], which is a generalisation of the LPN problem on fields $\mathbb{F}_q$ with $q > 2$. The last two introduce the *ring-LWE* problem, an algebraic variant of the LWE problem. According to the authors, it is the first truly practical lattice-based cryptosystem based on the LWE problem.

Recall that HELEN is somewhat a code-based cryptosystem. This class of systems includes some well-known cryptosystems like the McEliece cryptosystem [McE78] and its dual the Niederreiter cryptosystem [Nie86], which are code-based making use of Goppa codes.

Besides LWE-based cryptosystem, there are other lattice-based cryptosystem, for instance NTRU [HPS98], a system based on the hardness of the shortest vector problem in a particular class of lattices. We refer the reader to [Ber09] for a more exhaustive survey on post-quantum cryptosystems.

More closely-related cryptosystems were suggested. Gentry et al. introduced an LWE-based cryptosystem [GPV08] in which users share a common random matrix and whose private key (resp. public key) consists in a random error vector (resp. its syndrome). Extensions to $p = 2$ have been open so far. Our procedure is different from theirs in the sense that we hide a low-parity check equation in a matrix so that this matrix looks random, whereas they pick a totally random matrix. Similarly, Alekhnovich proposed a scheme based on problem to distinguish $(A, Ax+e)$ with x following uniform distribution and e either in $\binom{n}{n^\delta}$ or $\binom{n}{n^\delta+1}$ with $\delta < 1/2$ which he conjectures to be hard [Ale03]. Our scheme differs with the scheme proposed in [Ale03] in the following ways. First, we encode the bit so that decryption is correct with constant probability ϕ and which is independent from the encrypted bit b (in [Ale03], this probability is just known to be close to one for $b = 0$ and $1/2$ for $b = 1$). Finally, we propose concrete parameters and asymptotic parameters for our scheme. Applebaum et al. proposed a scheme, which is very similar to ours but which uses sparse matrices instead of random ones. Thus,

the security reduces to the less-studied 3LIN problem instead of LPN. This problem is similar to the LPN problem except that queries are done with vectors of weight 3 instead of random vectors. Also, the authors do not provide any concrete parameters [ABW10]. In Asiacrypt 2012, Döttling et al. presented an IND-CCA secure cryptosystem based on Alekhnovich's scheme, but again, no concrete parameters are given [DMQN12]. IND-CCA security is obtained using a technique by Dolev et al. [DDN91] based on one-time signatures and a tool by Rosen and Segev [RS09b]. We want also to mention a result by Damgård and Park who tries to characterize how practical a cryptosystem based on LPN can be [DP12]. At the time of the publication of the paper related to this chapter, to the best of our knowledge, we proposed for the first time a *concrete public key cryptosystem* whose security is based on LPN.

5.2 The Cryptosystem

We will first consider how to encrypt one single bit b. Hence, our message space is $\mathcal{M} = \{0,1\}$. We denote the cryptosystem by HELEN. We generalize the encryption to multiple bits in Section 5.5.

HELEN uses the following parameters which are described below: n, k, p, w, c, and $\mathcal{H}$. We encode first our message bit b with a binary $[n,1]$-error-correcting code C_1, for $n \in \mathbb{N}$. The goal of this code is to be able to recover b when errors occur. Let $c \in \{0,1\}^n$ be the generating matrix of this code (in fact, it is a vector). We encode b as $b \cdot c$. This message is hidden by a random codeword from a random binary linear $[n,k]$-code C_2 which has a low-weight parity-check equation $h \in \{0,1\}^n$ and a generator matrix $G \in \{0,1\}^{k \times n}$. The parameter $k \in \mathbb{N}$ determines the dimension of the codeword space in C_2 and needs to be tuned so that the system has the required security. The parity-check equation h will be the *private key* of our system while G will be the *public key.* As h is a parity check equation of the code generated by G, we have $h \cdot G^t = 0$. We denote the weight of h by w and the set of all possible h by $\mathcal{H}$. In the following, $\mathcal{H}$ will be the set of all vectors of weight w and dimension n but we keep this more general $\mathcal{H}$ for further improvements. We also hide the message further by adding some low weight random noise vector $\nu \in \{0,1\}^n$ produced by a source S_p.

For correct decryption, we require also that $h \cdot c^t = 1$ for all $h \in \mathcal{H}$. When $\mathcal{H}$ contains all the vectors of weight w, this condition implies $c = (1, \ldots, 1)$ (see (5.1) below).

In the following, we describe more precisely the cryptosystem. All algorithms are summarized in Algorithms 5.1, 5.2, and 5.3.

Before formally describing the cryptosystem, we will need the following technical lemma.

Lemma 5.1. *Let X be a random variable defined as the sum modulo 2 of w iid Bernoulli random variables $\nu_1, \cdots \nu_w$ equals to 1 with probability p and to 0 else. Then*

$$\Pr[X = 1] = \frac{1 - (1-2p)^w}{2} .$$

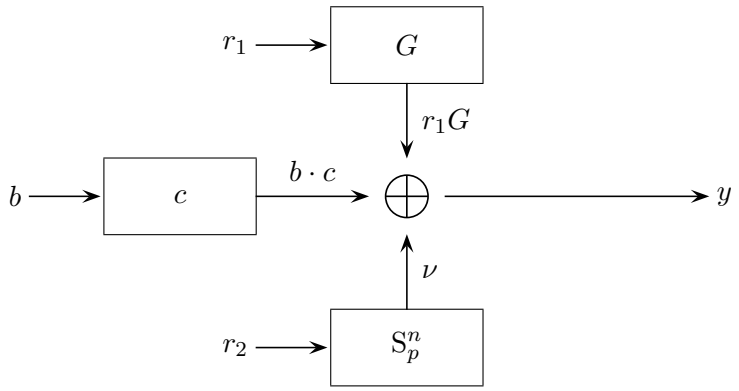


Figure 5.1 – The HELEN encryption

Proof. On one hand, we have

$$\mathbb{E}\left[(-1)^X\right] = \Pr[X = 0] - \Pr[X = 1] = 1 - 2\Pr[X = 1] \,.$$

On the other hand,

$$\mathbb{E}\left[(-1)^X\right] = \prod_{i=1}^{w} \mathbb{E}\left[(-1)^{\nu_i}\right] = (1-2p)^w \,.$$

Combining the two equations shows the result. □

5.2.1 Encryption

A bit $b \in \mathcal{M}$ is encrypted as

$$\mathsf{BEnc}(G, b; r_1 \| r_2) = b \cdot c \oplus r_1 G \oplus \nu \,,$$

where c is the generator vector for C_1, G is the generator matrix for C_2, $r_1 \in \{0,1\}^k$ is random and $\nu \coloneqq \mathrm{S}_p^n(r_2)$, i.e., it is the n first bits generated by the source S_p with random seed r_2. The ciphertext space is, thus, $\mathcal{C} = \{0,1\}^n$. The complexity of encryption is $O(kn)$.[2]

5.2.2 Decryption

We define

$$b' \coloneqq \mathsf{BDec}(h, y) = h \cdot y^t \,.$$

[2]Recently Gwanbae Choi showed [Cho15] that when encrypting many plaintexts at the same time (batch encryption), one could obtain a small improvement in this complexity, using more clever matrix multiplication algorithms (e.g., Strassen's algorithm [Str69]).

Algorithm 5.1 The HELEN encryption algorithm

Input: A bit b to encrypt, a public key G, two random seeds r_1 and r_2, a length n, an n-bit vector c, and a noise parameter p.
Output: A ciphertext y encrypted under the public key G.
1: Let $\nu \coloneqq \mathrm{S}_p^n(r_2)$.
2: **return** $y \leftarrow b \cdot c \oplus r_1 G \oplus \nu$.

Given a ciphertext $y \in \{0,1\}^n$, we recover the original message by first removing the noise due to C_2. This is done by applying h on y since $h \cdot G^t = 0$. Hence, we get

$$b' \coloneqq \mathsf{BDec}(h, y) = h \cdot y^t = (h \cdot c^t \cdot b^t) \oplus \nu' \,,$$

for $\nu' \coloneqq h \cdot \nu$ a noise with

$$\Pr[\nu' = 1] = \frac{1 - (1-2p)^w}{2}$$

by Lemma 5.1. Note that it is necessary that

$$h \cdot c^t = 1 \tag{5.1}$$

for all vector $h \in \mathcal{H}$ if one wants to be able to recover b. When $\mathcal{H}$ includes all vectors of weight w, this condition is equivalent to setting c to the all-one vector and w to an odd number. The resulting bit b' is then different from b with probability φ, which is given in the following theorem.

Theorem 5.2. *HELEN is a φ-cryptosystem, where*

$$\varphi \coloneqq \frac{1 - (1-2p)^w}{2} \,.$$

Note that the complexity of decryption is $O(n)$.

Algorithm 5.2 HELEN decryption algorithm

Decryption:
Input: A ciphertext y and a private key h.
Output: The original plaintext b with probability φ defined in Theorem 5.2.
1: **return** $b' \leftarrow h \cdot y^t$.

5.2.3 Key Generation

We need now to generate a code that is indistinguishable from a random code but that contains a known secret parity-check equation h of low weight. Let w be the required weight of h and let $\mathcal{H}$ be the set of all possible private keys. We propose the following key generation scheme.

1. Draw a random vector h of length n in the set $\mathcal{H}$. This vector will be the private key.
2. Let $0 < u \leq n$ be any index of h such that $h_i = 1$, e.g., $\max\{i \colon h_i = 1\}$.
3. Let $g_{ij} \leftarrow \mathrm{Ber}(\frac{1}{2})$, for $1 \leq i \leq k$ and $1 \leq j \leq n$, $j \neq u$.
4. Let

$$g_{iu} = \sum_{\substack{1 \leq j \leq n \\ j \neq u}} g_{ij} h_j$$

   for $1 \leq i \leq k$, where the sum is taken over $\mathbb{F}_2$.
5. Return the matrix $G \coloneqq [g_{ij}]_{\substack{1 \leq i \leq k \\ 1 \leq j \leq n}}$ and the vector h.

The resulting public key size is $k \times n$ bits, as we have to store the matrix G. The private key is $w \log n$ bits long. The key generation complexity is $O(k \times n)$. Note that we have $hG^t = 0$.

Algorithm 5.3 HELEN key generation algorithm

Input: Lengths k, n and a set $\mathcal{H}$.
Output: A private key h and a public key G.

1: Draw a random vector h of length n in the set $\mathcal{H}$.
2: Let $0 < u \leq n$ be any index of h such that $h_i = 1$, e.g., $\max\{i \colon h_i = 1\}$.
3: Let $g_{ij} \leftarrow \mathrm{Ber}(\frac{1}{2})$, for $1 \leq i \leq k$ and $1 \leq j \leq n$, $j \neq u$.
4: Let

$$g_{iu} = \sum_{\substack{1 \leq j \leq n \\ j \neq u}} g_{ij} h_j$$

   for $1 \leq i \leq k$, where the sum is taken over $\mathbb{F}_2$.
5: **return** the matrix $G \coloneqq [g_{ij}]_{\substack{1 \leq i \leq k \\ 1 \leq j \leq n}}$ and the vector h.

5.3 Security Analysis

We will reduce the security of our scheme to the LPN problem presented in Section 3.1. To do this, we will proceed in two steps. First, we show that the code we construct for C_2 is computationally indistinguishable from a random matrix.

5.3.1 Link to Random Codes

We first show in Lemma 5.3, that our key generation algorithm is equivalent to an algorithm that generates a random matrix such that it has at least one parity-check equation in $\mathcal{H}$. The first generator is our key generation algorithm.

Generator A: Run the key generation algorithm to obtain G and h and return $A \coloneqq G$.

Generator G_1: Draw a random $h \in \mathcal{H}$. Then, draw a random $k \times n$ matrix B until it has h as parity check equation.

Generator B: Return a random $k \times n$ matrix B.

Lemma 5.3.

$$\Delta(A, G_1) = 0 \ .$$

Proof. We have to show that generator A and generator G_1 produce the same distribution. First, note that showing that one line of A and G_1 are generated using the same distribution is enough, as the lines are drawn independently one from another. To finish the proof, note first that in both cases, h is drawn with uniform distribution. Let g_i be one line of G and note that, when $h \neq 0$,

$$\Pr[g_i \leftarrow A \mid h \leftarrow A] = \frac{1}{2^{n-1}} = \Pr[g_i \leftarrow G_1 \mid h \leftarrow G_1] \ .$$

If $h = 0$, any g_i would be accepted in both cases.[3] □

We want now to link this distribution with the distribution of an uniformly distributed $k \times n$ matrix, i.e., a matrix produced by generator B. We will need suitable parameters such that G_1 is *computationally indistinguishable* from B.
The best distinguisher between G_1 and B consists in deciding whether the output of the unknown generator has a parity-check equation in $\mathcal{H}$ or not. As discussed, the decisional problem is believed as hard as the computational problem. Hence, we extend Assumption 3.9 to the following one.

Assumption 5.4. *For any distinguisher between G_1 and B, the complexity over advantage ratio is lower bounded by $\mathsf{T}_{\mathsf{MDP}}(w, n, k)$, which is defined in* (3.3).

So, by selecting parameters such that the $\mathsf{T}_{\mathsf{MDP}}(w, n, k) \geq 2^{\lambda}$, for a security parameter λ, any game involving our cryptosystem produces a computationally indistinguishable outcome when the key generator is replaced by B.

5.3.2 Semantic Security

Now that we have B computationally indistinguishable from A, we can link our cryptosystem with the LPN problem.

[3] Note that this case would imply that $0 \in \mathcal{H}$, which would break the correctness of our cryptosystem.

Theorem 5.5. *If the (n, k, w)-decisional minimum distance problem is (t_1, ε_1)-hard and if the (k, p)-decisional LPN problem is (t_2, ε_2)-hard, then there exists a constant τ such that our cryptosystem is*

$$(\min\{t_1, t_2 - \tau kn\}, 2(\varepsilon_1 + \varepsilon_2))\text{-IND-CPA-}\textit{secure}\ .$$

Proof. We recall the real-or-random IND-CPA game defined in Definition 2.8. We introduce the following three games Γ_0, Γ_1 and Γ_2. Γ_0 is the IND-CPA game for our cryptosystem in the simple real-or-random model. Γ_1 is the IND-CPA game in the same model but using generator B instead of A. Γ_2 is the (k, p)-D-LPN game.

By the assumptions, we know that the best advantage between Γ_0 and Γ_1 is ε_1. For the best advantage between Γ_1 and Γ_2, we do the following. Recall that in the simple real-and-random game this model, the adversary submits first a chosen plaintext b using an algorithm $\mathcal{A}_1^{\mathsf{ror}}(G)$. Then, given a n-bit word u, has to decide using an algorithm $\mathcal{A}_2^{\mathsf{ror}}(G, u)$, whether u is the encryption of b or is a random bitstring. Let $(\mathcal{A}_1^{\mathsf{ror}}(G), \mathcal{A}_2^{\mathsf{ror}}(G, u))$ be an IND-CPA adversary for our cryptosystem when G is generated using generator B.

We show that using this adversary, we can create an adversary $\mathcal{B}$ that solves the D-LPN problem. $\mathcal{B}$ first queries the unknown oracle of the D-LPN problem n times to obtain n-vectors $\alpha_1, \ldots, \alpha_n$. Note that each of these α_i has exactly $k + 1$ bits. He creates then the $k \times n$ matrix $\tilde{G}$ using the first k bits of α_i as column i, for $1 \leq i \leq n$. Using $\mathcal{A}_1^{\mathsf{ror}}(\tilde{G})$, he ca recover a plaintext b. Let $z \coloneqq b \cdot c \oplus (\alpha_{1|k+1} \| \ldots \| \alpha_{n|k+1})$, where $\alpha_{i|k+1}$ denotes the $k + 1$-th bit of α_i. If the unknown oracle returns random bitstrings, then z will be random as well. However, if it is an LPN oracle, then z is a valid ciphertext of b using the public key $\tilde{G}$. Note also that the matrix $\tilde{G}$ follows the same distribution as the output of generator B.

Hence, using $\mathcal{A}_2^{\mathsf{ror}}(\tilde{G}, z)$, we can decide whether z is a ciphertext corresponding to b or not. The complexity of this simulation is τkn for a constant $\tau > 0$ large enough. Thus, $\mathcal{B}$ wins Γ_2 with the same advantage as $(\mathcal{A}_1, \mathcal{A}_2)$ wins Γ_1.

As the D-LPN problem is supposed (t_2, ε_2)-hard, we get that our cryptosystem when we use generator B is $(t_2 - \tau kn, \varepsilon_2)$-IND-CPA-secure in the simple real-or-random model. Similarly, we get that the original cryptosystem is $(\min\{t_1, t_2 - \tau kn\}, \varepsilon_1 + \varepsilon_2)$-IND-CPA-secure in the simple real-or-random model. Thus, our cryptosystem is $(\min\{t_1, t_2 - \tau kn\}, 2(\varepsilon_1 + \varepsilon_2))$-IND-CPA-secure in the standard model [BDJR97b]. □

Hence, we reduced the semantic security of our cryptosystem to the hardness of the decisional LPN problem with n queries and noise parameter p.

Note that since we encrypt one single bit, an IND-CPA adversary has to distinguish $\mathsf{BEnc}(G, 0)$ from $\mathsf{BEnc}(G, 1)$ which is equivalent to OW-CPA security.

5.4 Selection of Parameters

To summarize, we need to tune the following security parameters for HELEN:

- The dimension k of the code C_2 generated by G,
- The ciphertext length n (also the length of the codewords in C_2),
- The weight w of the secret key, and
- The noise probability p.

For our cryptosystem to be semantically secure, we need the parameters to verify Theorem 5.5. In particular, this implies that the D-LPN problem should be hard and that finding a low-weight parity-check equation in the code is hard as well, i.e., that $\mathsf{T}_{\mathsf{MDP}}(w, n, k) \geq 2^{\lambda}$ We need also w to be odd. For the LPN problem, we want $\mathsf{T}_{\mathsf{LPN}} \geq 2^{\lambda}$, where $\mathsf{T}_{\mathsf{LPN}}$ is given in Equation (3.2).
Recall that the probability of decrypting incorrectly a bit is

$$P_{\mathsf{error}} \coloneqq \frac{1-(1-2p)^w}{2} \,. \tag{5.2}$$

Hence, to compare different parameters, we will normalize them with the capacity of a binary symmetric channel (BSC) with parameter P_{error}. Recall that the capacity of the BSC is $C \coloneqq 1 - H_2(P_{\mathsf{error}})$ with $H_2(p) \coloneqq -p\log(p) - (1-p)\log(1-p)$ the binary entropy. We normalize by this factor, as we know that such a rate is achievable by the channel coding theorem (see, e.g., [CT06]). This gives us a good way of comparing the parameters.
We propose two sets of parameters. Some (I) which minimizes the n/C ratio to minimize the number of transmitted bits and some (II) with a smaller kn/C ratio to minimize the encryption/decryption complexity. We give in Table 5.1 concrete parameters for different security parameters λ.

Table 5.1 – Parameters for our cryptosystem

	λ	k	n	w	p	kn	n/C	kn/C	$\mathsf{T}_{\mathsf{MDP}}$	$\mathsf{T}_{\mathsf{LPN}}$	C
I	64	4 500	18 000	33	0.01	$2^{26.3}$	$2^{16.4}$	$2^{28.6}$	$2^{65.3}$	$\geq 2^k$	0.20
II	64	2 200	16 000	23	0.02	$2^{25.0}$	$2^{17.1}$	$2^{28.2}$	$2^{64.7}$	$\geq 2^k$	0.11
I	80	5 600	28 000	35	0.01	$2^{27.2}$	$2^{17.2}$	$2^{29.7}$	$2^{80.5}$	$\geq 2^k$	0.18
II	80	2 800	27 000	25	0.02	$2^{26.2}$	$2^{18.1}$	$2^{29.6}$	$2^{80.4}$	$\geq 2^k$	0.10

In Table 5.2, we compare for concrete parameters HELEN with the code-based McEliece cryptosystem [McE78] and with an LWE-based cryptosystem [LP11]. Note that for encryption and decryption time, we neglect the cost of encoding and decoding.
We propose the following asymptotic parameters for our system:

$$k = \Theta\left(\lambda^2\right) \qquad n = \Theta\left(\lambda^2\right) \qquad w = \Theta\left(\lambda\right) \qquad p = \Theta\left(1/\lambda\right) .$$

Indeed, we obtain $\mathsf{T}_{\mathsf{MDP}}$ and $\mathsf{T}_{\mathsf{LPN}} \geq 2^{\lambda}$, $\mathsf{D}_{\mathsf{A},\mathsf{G}_3} \leq 2^{-\lambda}$, $P_{\mathsf{error}} = \frac{1}{2} - \frac{1}{e^{O(1)}}$, and $C > 0$. In Table 5.3, we compare the asymptotic complexity of HELEN with the complexity of various cryptosystems of which we could find some asymptotic parameters. Analysing this table, we can see that, although HELEN is not as good as NTRU or McEliece, it performs better than RSA and even TCHo.[4]

Table 5.2 – Comparison with other cryptosystems

Name	λ	Message expansion	Pub key size	Encryption time	Decryption time
HELEN I	80	$2^{17.2}$	$2^{27.2}$	$O\left(2^{29.7}\right)$	$O\left(2^{17.2}\right)$
McEliece [BLP08]	80	1.29	$2^{18.8}$	$O\left(2^{21.0}\right)$	$O\left(2^{21.3}\right)$
LWE [LP11]	128	22	$2^{17.5}$	$O\left(2^{24}\right)$	$O\left(2^{18.5}\right)$
Ring-LWE [LP11]	128	22	$\approx 2^{10}$	$O\left(2^{24}\right)$	$O\left(2^{18.5}\right)$

Table 5.3 – Asymptotic comparison with other cryptosystems. The $\Theta(.)$'s have been omitted.

Name	Message expansion	Pub key size	Private key size	Key generation	Encryption	Decryption
HELEN	λ^2	λ^4	$\lambda \log \lambda$	λ^4	λ^4	λ^2
TCHo [DV13b]	λ^2	λ^2	$\lambda \log \lambda$	$\lambda^6 \log \lambda \log\log \lambda$	λ^5	λ^4
McEliece [BLP08]	1	λ^2	λ^2	λ^3	λ^2	$\lambda^2 \log \lambda$
RSA	1	λ^3	λ^3	λ^{12}	λ^6	λ^9
NTRU [HPS98]	1	λ	λ	λ^3	λ^2	λ^2

5.5 Encrypting More than One Bit

In this section, we show how to encrypt more than one bit using HELEN. Taking advantage of an efficient coding scheme, we can also improve the probability of decrypting correctly the message. In addition to the previous parameters n, k, p, w and $\mathcal{H}$, we add a $[\mu, \kappa]$-error-correcting code. Let Encode be this $[\mu, \kappa]$-error-correcting code. Let also Decode be an efficient decoding algorithm corresponding to this code.

Encryption: We encrypt a plaintext $m \in \{0,1\}^{\kappa}$ in two steps. First we compute $b_1 \| \ldots \| b_{\mu} \coloneqq \mathsf{Encode}(m)$. The ciphertext c is then $\mathsf{BEnc}(G, b_1) \| \ldots \| \mathsf{BEnc}(G, b_{\mu})$. The complexity of encryption is $O(\mu k n + T_{\mathsf{Encode}})$, where T_{Encode} is the complexity of the encoding algorithm.

[4] For the comparison with TCHo, only the public key size is worse in HELEN.

Decryption: To decrypt, we first decrypt each block of n bits using BDec to recover $b'_1\| \dots \|b'_\mu$, where each $b'_i \neq b_i$ with probability $(1-(1/2p)^w)/2 =: P_{\mathsf{error}}$. The complexity of decryption is $O(\mu n + T_{\mathsf{Decode}})$, where T_{Decode} is the complexity of the decoding algorithm. Let ρ be the maximum number of errors the error-correcting code can correct. Then, the probability of decrypting incorrectly the message is

$$\sum_{i=\rho+1}^{\mu} \binom{\mu}{i} (P_{\mathsf{error}})^i (1-P_{\mathsf{error}})^{\mu-i} \leq \exp\left[-2\mu\left(\frac{\rho}{\mu} - P_{\mathsf{error}}\right)^2\right] =: \phi \tag{5.3}$$

by the Hoeffding bound (Theorem 2.15).

Theorem 5.6. *HELEN with parameter μ, κ is a ϕ-cryptosystem, where ϕ is given in* (5.3).

Very recently Gwangbae Choi [Cho15] instantiated HELEN using a $[23, 12, 7]$-Golay code and showed a reduction of the ciphertext length by a factor $2/3$ with respect to basic repetition codes.

5.5.1 Security

Theorem 5.7. *Let ε_b be the* IND-CPA *advantage for the elementary cryptosystem HELEN with $\mu = \kappa = 1$. Then, the advantage of an* IND-CPA *adversary against the full cryptosystem HELEN with parameter μ and κ is smaller than $\mu\varepsilon_b$.*

Proof. Let $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$ be an IND-CPA adversary HELEN with parameter μ, κ. Given $i \in \{1, \dots, \mu\}$, we define $\mathcal{B}_i := (\mathcal{B}_{i,1}(G), \mathcal{B}_{i,2}(G, c))$ as follows.

$\mathcal{B}_{i,1}(G)$**:**

1. Let $m_0, m_1 \leftarrow \mathcal{A}_1(G)$
2. Let $b_1^0\| \dots \|b_\mu^0 \leftarrow \mathsf{Encode}(m_0)$, the encoding of m_0
3. Let $b_1^1\| \dots \|b_\mu^1 \leftarrow \mathsf{Encode}(m_1)$, the encoding of m_1
4. Return b_i^0, b_i^1.

$\mathcal{B}_{i,2}(G, c)$**:**

1. Compute $c_1 \leftarrow \mathsf{BEnc}(G, b_1^1), \dots, c_{i-1} \leftarrow \mathsf{BEnc}(G, b_{i-1}^1)$.
2. Let $c_i = c$
3. Compute $c_{i+1} \leftarrow \mathsf{BEnc}(G, b_{i+1}^0), \dots, c_\mu \leftarrow \mathsf{BEnc}(G, b_\mu^0)$.
4. Set $y := c_1\| \dots \|c_\mu$
5. return $\mathcal{A}_2(G, y)$

We know that $\mathrm{Adv}\,\mathcal{B}_i \leq \varepsilon_b$. We have

$$\Pr\left[\mathcal{A} \to 0 \mid m_0 \text{ encrypted}\right] = \Pr\left[\mathcal{B}_1 \to 0 \mid b_1^0 \text{ encrypted}\right]$$

and

$$\Pr\left[\mathcal{A} \to 0 \mid m_1 \text{ encrypted}\right] = \Pr\left[\mathcal{B}_\mu \to 0 \mid b_\mu^1 \text{ encrypted}\right] \ .$$

Also,

$$\Pr\left[\mathcal{B}_i \to 0 \mid b_i^1 \text{ encrypted}\right] = \Pr\left[\mathcal{B}_{i+1} \to 0 \mid b_{i+1}^0 \text{ encrypted}\right] \ .$$

Hence,

$$\begin{aligned}\mathrm{Adv}\,\mathcal{A} &= \left(\Pr\left[\mathcal{A} \to 0 \mid m_0 \text{ encrypted}\right] - \Pr\left[\mathcal{A} \to 0 \mid m_1 \text{ encrypted}\right]\right) \\ &= \sum_{i=1}^{\mu} \left(\Pr\left[\mathcal{B} \to 0 \mid b_i^0 \text{ encrypted}\right] - \Pr\left[\mathcal{B} \to 0 \mid b_i^1 \text{ encrypted}\right]\right) \leq \mu\varepsilon_b \ .\end{aligned}$$

□

5.6 IND-CCA-security

Obviously HELEN is not IND-CCA-secure, as it is clearly malleable. It suffices to change one single bit of the ciphertext and to submit it to the decryption oracle to decrypt the plaintext with good probability. To achieve IND-CCA security, one can use well-known construction like the Fujisaki-Okamoto hybrid construction [FO99]. This construction uses two random oracles H_1 and H_2 as well as a symmetric encryption scheme. However, such a construction work only if the cryptosystem is Γ-uniform.

Definition 5.8 (Γ-uniformity)**.** *Let* Enc *be an asymmetric encryption scheme, with key generation algorithm* $\mathsf{Gen}(1^\lambda)$ *and encryption algorithm* $\mathsf{Enc}(K_p, m; r)$ *over the message space* $\mathcal{M}$ *and the random coins space* $\mathcal{R}$*.* Enc *is* Γ-uniform *if for any plaintext* $m \in \mathcal{M}$*, for any keys drawn by* Gen *and for any* $y \in \{0,1\}^*$*, we have*

$$\Pr\left[h \xleftarrow{U} \mathcal{R} : y = \mathsf{Enc}(K_p, m; h)\right] \leq \Gamma \ ,$$

i.e., the probability that a plaintext and a ciphertext match is bounded.

Lemma 5.9. *HELEN is* $(1-p)^n$*-uniform.*

Proof. Recall that the HELEN encryption of b is $y = b \cdot c \oplus r_1 G \oplus \mathrm{S}_p^n(r_2)$, for random coins r_1 and r_2. We need to bound the probability (taken over r_1 and r_2) that a given plaintext x and ciphertext y match. As in HELEN we consider only $p < \frac{1}{2}$, the most probable ciphertext corresponds to $y = b \cdot c \oplus r_1 G$, i.e., when S_p^n is the zero bitstring. This happens with probability $(1-p)^n$. When we take the average over the possible r_1, this probability can only decrease. Hence, HELEN is $(1-p)^n$-uniform. □

Theorem 5.10. *Let q_1(resp. q_2) be the number of queries an adversary makes to H_1 (resp. H_2). Let q_d be the number of queries performed to the decryption oracle. Then, if HELEN is (t, ϵ)-IND-CPA-secure, the Fujisaki-Okamoto hybrid construction using a one-time pad for symmetric encryption with key length ℓ is (t_1, ϵ_1)-IND-CCA-secure in the random oracle model, where*

$$t_1 \coloneqq t - O\left((q_1 + q_2) \times (k + \ell)\right)$$
$$\epsilon_1 \coloneqq (2(q_1 + q_2)\epsilon + 1)(1 - (1 - p)^n - 2^{-\ell})^{-q_d} - 1 \ .$$

Proof. Since HELEN is OW-CPA secure and $(1-p)^n$-uniform (see Lemma 5.9), the result follows from [FO99, Theorem 14]. □

Chapter 6

A New Algorithm Solving the Learning With Error Problem

The work presented in this chapter is part of a joint work with F. Tramèr and Prof. S. Vaudenay and was published in [DTV15].

Notation. *In this chapter as well as in the next one, we define* $\sqrt{-1} = i \in \mathbb{C}$.

6.1 Previous Work

Algorithms solving LWE can be divided into two categories: those finding short vectors in a lattice using, e.g., Regev's [Reg09] or Brakerski et al.'s [BLP+13] reduction and those attacking the LWE problem directly. The first type of algorithms is extensively studied (see, e.g., [BGJ14, LP11, CN11, Ngu11, HPS11b, HPS11a, GNR10, NS09b]). However, there is still no precise complexity analysis for large dimensions. In this chapter, we focus only on the second type of algorithms the study of which started with the LPN problem and the BKW Algorithm [BKW03] with complexity $2^{O(k/\log k)}$ where k is the length of the secret vector (see Chapter 4 for more background regarding the BKW algorithm).
In ICALP 2011, Arora and Ge publish the first algorithm targeting a specific version of LWE, namely when the Gaussian noise is low [AG11] using algebraic attacks. This result was later improved by Albrecht et al. [ACF+14]. Using BKW for LWE was first mentioned by Regev [Reg09]. However, it is only in 2013 that the first detailed analysis of a generic algorithm targeting LWE is published by Albrecht et al. [ACF+13]. It is an adaptation of the original BKW algorithm with some clever improvements of the memory usage and achieves complexity $2^{O(k)}$. Their analysis is extremely detailed and we already presented their result in Chapter 4. Finally, Albrecht et al. presented in PKC 2014 an algorithm targeting LWE when the secret vector has small components (typically binary). Using BKW along with modulus switching techniques, they managed to reduce the complexity for solving the LWE problem in these cases [AFFP14].

6.2 Our Contribution

We contributed in the following:

- First we suggest a *new algorithm for LWE*, which is better than the current state of the art. Our new algorithm replaces the log-likelihood part from [ACF+13] by a multidimensional Fourier transform. We also put forward a heuristic adapted from LF2 [LF06] to reduce the number of oracle queries even further.

- Albrecht et al. in [ACF+13] were relying on the heuristic that the sum of rounded Gaussian variables remains a rounded Gaussian. We *remove this heuristic* by a careful analysis. In particular, we give good bounds on the expected value of the cosine of the rounded Gaussian distribution. Our algorithm relies solely on the common heuristic stating that after having performed all the XORs in the BKW algorithm, all the noises are independent. This heuristic is already used in most of the LPN-solving algorithms (e.g. [LF06, FMI+06, ACF+13]).

- In [ACF+13], only the rounded Gaussian distribution for the noise in LWE is considered. While this distribution was initially used by Regev [Reg09], more recent papers tend to use the discrete Gaussian distribution instead. We perform our analysis for both distributions.

- Albrecht et al.'s complexity is rather difficult to estimate when $\sqrt{2^a}\sigma > q/2$ (see for instance [ACF+13, Theorem 2]). Indeed, their result contains a parameter which they could express only using an integral and the erf function. Our detailed analysis allows us to bound the Fourier coefficients of the rounded Gaussian distribution in all the cases and, hence, all our complexities are simple to evaluate.

- Finally, we adapt Lyubashevsky's idea for LPN that we presented in Section 3.1.3 to LWE and show that for LWE, the minimum number of queries required using his method is $k^{1+(\log q+1)/\log k}$.

6.3 Our LWE Algorithm

In this section, we present our new LWE-solving algorithm. Following the structure from [ACF+13], our algorithm will also consist of the sample reduction, hypothesis testing and back substitution phases. However, we change the hypothesis testing phase with an idea similar to the LF1 algorithm [LF06]. Indeed, as the Walsh-Hadamard transform can be seen as a multidimensional discrete Fourier transform in $\mathbb{Z}_2$, it would seem plausible that a similar optimization could be achieved over $\mathbb{Z}_q$ for LWE. As we have seen, the BKW algorithm for LWE from [ACF+13] differs slightly from the original BKW algorithm in its reduction phase. Recall that after reducing samples to a block of size $k' \leq b$, Albrecht et al. further reduce the samples to d elements. Our idea is to remove this last reduction to d elements and recover directly the k' elements of $\boldsymbol{s}$

using a DFT. Thus, the samples we use for the DFT would have noise sampled from the sum of 2^{a-1} discretized Gaussians instead of 2^a, which might also lead to a significant improvement. As for most other works on LPN or LWE solving algorithms, we will make use of an heuristic assumption of independence for the noise of the reduced samples.
Finally, note that the LF1 algorithm uses the exact same reduction phase as the original BKW. Similarly, our algorithm will use (nearly) the same reduction phase as in [ACF+13], combined with a different hypothesis testing phase. The major differences in our reduction phase will be that we perform one reduction round less, and that we decide to store and re-use samples for solving successive blocks of $\boldsymbol{s}$.

6.3.1 Sample reduction

As mentioned previously, our algorithm uses the same reduction phase as the BKW algorithm from [ACF+13], except that we always stop the reduction as soon as we reach a block of $k' \leq b$ non-zero elements. We will construct the oracles $\mathcal{A}_{\boldsymbol{s},\chi,\ell}$ and the tables T^ℓ only for $1 \leq \ell \leq a-1$. It is thus fairly trivial to adapt the results from [ACF+13] to bound the complexity of our algorithm's reduction phase.

Lemma 6.1 (Lemma 2 and 3 from [ACF+13])**.** *Let k, q be positive integers and $\Pi_{\boldsymbol{s},\chi}$ be an LWE oracle, where $\boldsymbol{s} \in \mathbb{Z}_q^k$. Let $a \in \mathbb{Z}$ with $1 \leq a \leq k$, let b be such that $ab \leq k$, and let $k' = k - (a-1)b$. The worst case cost of obtaining m samples $(\boldsymbol{a}_i, c_i)$ from the oracle $\mathcal{A}_{\boldsymbol{s},\chi,a-1}$, where the $\boldsymbol{a}_i$ are zero for all but the first k' elements, is upper bounded by*

$$\left(\frac{q^b-1}{2}\right)\left(\frac{(a-1)\cdot(a-2)}{2}(k+1) - \frac{ab\cdot(a-1)\cdot(a-2)}{6}\right) + m\left(\frac{a-1}{2}(k+2)\right)$$

additions in $\mathbb{Z}_q$ and $(a-1)\cdot\frac{q^b-1}{2} + m$ calls to $\Pi_{\boldsymbol{s},\chi}$.
The memory required in the worst case to store the set of tables T^1 through T^{a-1}, expressed in elements of $\mathbb{Z}_q$ is upper bounded by

$$\left(\frac{q^b-1}{2}\cdot(a-1)\cdot\left(k+1-b\frac{a-2}{2}\right)\right) .$$

Proof. The proof follows exactly the one from [ACF+13], with the exception that we do not use any table T^a. □

6.3.2 Hypothesis testing

At the end of the reduction phase, we are left with m samples $(\boldsymbol{a}^{(j)}, c^{(j)})$ from the oracle $\mathcal{A}_{\boldsymbol{s},\chi,a-1}$, where each $\boldsymbol{a}^{(j)}$ has all elements equal to zero except for a block of size $k' = k - (a-1)\cdot b$. Let $\boldsymbol{s}'$ denote the corresponding block of the secret $\boldsymbol{s}$. We can view the oracle $\mathcal{A}_{\boldsymbol{s},\chi,a-1}$ as returning samples in $\mathbb{Z}_q^{k'} \times \mathbb{Z}_q$. We will consider that each such sample is the sum of 2^{a-1} samples (or their negation) from the LWE oracle $\Pi_{\boldsymbol{s},\chi}$. Then,

the noise $\langle \boldsymbol{a}^{(j)}, \boldsymbol{s}' \rangle - c^{(j)}$ will correspond to the sum of 2^{a-1} independent samples from the distribution χ, multiplied by ± 1, and taken modulo q. We perform our analysis when χ is the discrete Gaussian distribution (3.6) and when χ is the rounded Gaussian distribution (3.5) which are used in most of the LWE research, i.e., we let $\chi = D_{\sigma,q}$ or $\chi = \bar{\Psi}_{\sigma,q}$.

We represent our m samples as a matrix $\boldsymbol{A} \in \mathbb{Z}_q^{m \times k'}$ with rows $\boldsymbol{A}_j$ and a vector $\boldsymbol{c} \in \mathbb{Z}_q^m$. Recall that $\theta_q \coloneqq \exp(2\pi i/q)$. Let us consider the function

$$f(\boldsymbol{x}) \coloneqq \sum_{j=1}^{m} \mathbb{1}_{\{\boldsymbol{A}_j = \boldsymbol{x}\}} \, \theta_q^{\boldsymbol{c}_j} \ , \quad \forall \boldsymbol{x} \in \mathbb{Z}_q^{k'} \ . \tag{6.1}$$

The discrete Fourier transform of f is

$$\widehat{f}(\boldsymbol{\alpha}) \coloneqq \sum_{\boldsymbol{x} \in \mathbb{Z}_q^{k'}} f(\boldsymbol{x}) \theta_q^{-\langle \boldsymbol{x}, \boldsymbol{\alpha} \rangle} = \sum_{\boldsymbol{x} \in \mathbb{Z}_q^{k'}} \sum_{j=1}^{m} \mathbb{1}_{\{\boldsymbol{A}_j = \boldsymbol{x}\}} \, \theta_q^{\boldsymbol{c}_j} \theta_q^{-\langle \boldsymbol{x}, \boldsymbol{\alpha} \rangle} = \sum_{j=1}^{m} \theta_q^{-(\langle \boldsymbol{A}_j, \boldsymbol{\alpha} \rangle - \boldsymbol{c}_j)} \ .$$

In particular, note that

$$\widehat{f}(\boldsymbol{s}') = \sum_{j=1}^{m} \theta_q^{-(\langle \boldsymbol{A}_j, \boldsymbol{s}' \rangle - c_j)} = \sum_{j=1}^{m} \theta_q^{-(\nu_{j,1} \pm \cdots \pm \nu_{j,2^{a-1}})} \ , \tag{6.2}$$

where the $\nu_{j,l}$ are independent samples from χ. Note that we dropped the reduction of the sum of the ν modulo q, since $\theta_q^{kq} = 1$, for $k \in \mathbb{Z}$.

We will now show, through a series of lemmas, that for appropriate values for m and a, the maximum value of the function $\mathrm{Re}(\widehat{f}(\boldsymbol{\alpha}))$ is reached by $\boldsymbol{s}'$ with high probability. Our algorithm for recovering $\boldsymbol{s}'$ will thus consist in finding the highest peak of the real part of the DFT of $f(\boldsymbol{x})$.

We start first with two technical lemmas regarding Gaussian distributions which might be of independent interest.

Lemma 6.2. *For q an odd integer, let $X \sim \bar{\Psi}_{\sigma,q}$ and let $Y \sim 2\pi X/q$. Then*

$$\mathbb{E}[\cos(Y)] \geq \frac{q}{\pi} \sin\left(\frac{\pi}{q}\right) e^{-2\pi^2\sigma^2/q^2} \quad \textit{and} \quad \mathbb{E}[\sin(Y)] = 0 \ .$$

Proof. Let S_ℓ be the set of integers in $]-q/2 + \ell q, q/2 + \ell q]$. Using (3.4) and (3.5), we

can write

$$\mathbb{E}[\cos(Y)] = \sum_{x\in S_0} \cos\left(\frac{2\pi}{q}x\right) \sum_{\ell=-\infty}^{\infty} \int_{x-1/2}^{x+1/2} p(\theta+\ell q;\sigma)\,\mathrm{d}\theta \tag{6.3}$$

$$= \sum_{\ell=-\infty}^{\infty} \sum_{x\in S_0} \cos\left(\frac{2\pi}{q}x + 2\pi\ell\right) \int_{x-1/2}^{x+1/2} p(\theta+\ell q;\sigma)\,\mathrm{d}\theta \tag{6.4}$$

$$= \sum_{\ell=-\infty}^{\infty} \sum_{x\in S_0} \cos\left(\frac{2\pi}{q}(x+\ell q)\right) \int_{x-1/2+\ell q}^{x+1/2+\ell q} p(\theta;\sigma)\,\mathrm{d}\theta \tag{6.5}$$

$$= \sum_{\ell=-\infty}^{\infty} \sum_{x'\in S_\ell} \cos\left(\frac{2\pi}{q}x'\right) \int_{x'-1/2}^{x'+1/2} p(\theta;\sigma)\,\mathrm{d}\theta \tag{6.6}$$

$$= \sum_{x'=-\infty}^{\infty} \cos\left(\frac{2\pi}{q}x'\right) \int_{x'-1/2}^{x'+1/2} p(\theta;\sigma)\,\mathrm{d}\theta \tag{6.7}$$

$$= \sum_{\chi=-\infty}^{\infty} \mathcal{F}\left(\cos\left(x\frac{2\pi}{q}\right) \int_{x-1/2}^{x+1/2} p(\theta;\sigma)\,\mathrm{d}\theta\right)(\chi)\ , \tag{6.8}$$

where, for (6.6), we used $x' \coloneqq x+\ell q$ and, for (6.8), we used Lemma 2.13. More generally, basics about continuous Fourier transforms and the Fourier transforms of $\cos(2\pi x/q)$ and $1/(\sigma\sqrt{2\pi})\exp[-x/(2\sigma^2)]$ can be found in Section 2.4.2. We are now ready to prove the lemma (we drop some (χ) for readability). For integer values of χ, we have

$$\mathcal{F}\left(\cos\left(x\frac{2\pi}{q}\right) \int_{x-1/2}^{x+1/2} \frac{1}{\sigma\sqrt{2\pi}} e^{-\theta^2/(2\sigma^2)}\,\mathrm{d}\theta\right) \tag{6.9}$$

$$= \mathcal{F}\left(\cos\left(x\frac{2\pi}{q}\right)\right) * \left(\mathcal{F}\left(\int_{-\infty}^{x+\frac{1}{2}} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{\theta^2}{2\sigma^2}}\,\mathrm{d}\theta\right) - \mathcal{F}\left(\int_{-\infty}^{x-\frac{1}{2}} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{\theta^2}{2\sigma^2}}\,\mathrm{d}\theta\right)\right) \tag{6.10}$$

$$= \mathcal{F}\left(\cos\left(x\frac{2\pi}{q}\right)\right) * \left(\left(e^{\pi i\chi} - e^{-\pi i\chi}\right)\mathcal{F}\left(\int_{-\infty}^{x} \frac{1}{\sigma\sqrt{2\pi}} e^{-\theta^2/(2\sigma^2)}\,\mathrm{d}\theta\right)\right) \tag{6.11}$$

$$= \frac{1}{2}\left(\delta\left(\chi-\frac{1}{q}\right) + \delta\left(\chi+\frac{1}{q}\right)\right) * \left(\left(e^{\pi i\chi} - e^{-\pi i\chi}\right)\left(\frac{1}{2\pi i\chi} e^{-2\pi^2\sigma^2\chi^2} + \frac{1}{2}\delta(\chi)\right)\right) \tag{6.12}$$

$$= \frac{1}{2}\left(\delta\left(\chi-\frac{1}{q}\right) + \delta\left(\chi+\frac{1}{q}\right)\right) * \left(\sin(\pi\chi)\left(\frac{1}{\pi\chi} e^{-2\pi^2\sigma^2\chi^2}\right)\right) \tag{6.13}$$

$$= \frac{q}{2\pi}\sin\left(\frac{\pi}{q}\right)(-1)^{\chi}\left(\frac{e^{-2\pi^2\sigma^2(q\chi+1)^2/q^2}}{q\chi+1} - \frac{e^{-2\pi^2\sigma^2(q\chi-1)^2/q^2}}{q\chi-1}\right)\ , \tag{6.14}$$

where (6.10) is the convolution property of the FT, (6.11) comes from the translation property of the FT, (6.12) comes from the integration property of the FT, and

(6.13) holds as $\delta(\chi \pm 1/q) * \delta(\chi) = 0$ for integer values of χ. We can write (6.8) as

$$\frac{q}{\pi}\sin\left(\frac{\pi}{q}\right)\exp^{-2\pi^2\sigma^2/q^2} + \sum_{\chi=1}^{\infty}\frac{q}{\pi}\sin\left(\frac{\pi}{q}\right)(-1)^{\chi}\left(\frac{e^{-2\pi^2\sigma^2(q\chi+1)^2/q^2}}{q\chi+1} - \frac{e^{-2\pi^2\sigma^2(q\chi-1)^2/q^2}}{q\chi-1}\right) .$$

Notice that the sum term in this equation is alternating and decreasing in absolute value when χ grows (derivative is negative). Notice also that the first term (when $\chi = 1$) is positive. Hence this sum is greater than 0 and we get our result for $\mathbb{E}[\cos(Y)]$.
For $\mathbb{E}[\sin(Y)]$, note that when q is odd, X and Y are perfectly symmetric around 0. The result then follows trivially from the symmetry of the sine function. □

Lemma 6.3. *For q an odd integer, let $X \sim D_{\sigma,q}$ and let $Y \sim 2\pi X/q$. Then*

$$\mathbb{E}[\cos(Y)] \geq 1 - \frac{2\pi^2\sigma^2}{q^2} \quad \textit{and} \quad \mathbb{E}[\sin(Y)] = 0 .$$

Proof. Using [Ban93, Lemma 1.3] with $a = 1/(2\sigma^2)$, we have that $\mathbb{E}[X^2] \leq \sigma^2$. Hence, using $\cos(x) \geq 1 - x^2/2$,

$$\mathbb{E}[\cos(2\pi X/q)] \geq 1 - 2\pi^2\mathbb{E}[X^2]/q^2 = 1 - 2\pi^2\sigma^2/q^2 .$$

For $\mathbb{E}[\sin(Y)]$, note that when q is odd, X and Y are perfectly symmetric around 0. The result then follows trivially from the symmetry of the sine function. □

Definition 6.4 ($R_{\sigma,q,\chi}$)**.** *In the following, let $R_{\sigma,q,\chi} \coloneqq \mathbb{E}[\cos(\chi)]$, i.e.,*

$$R_{\sigma,q,\chi} \coloneqq \begin{cases} \frac{q}{\pi}\sin\left(\frac{\pi}{q}\right)e^{-2\pi^2\sigma^2/q^2} & \textit{when } \chi = \bar{\Psi}_{q,\sigma} \\ 1 - \frac{2\pi^2\sigma^2}{q^2} & \textit{when } \chi = D_{q,\sigma} \end{cases}$$

Lemma 6.5. $\mathbb{E}\left[\mathrm{Re}(\widehat{f}(\boldsymbol{s}'))\right] \geq m \cdot \left(R_{\sigma,q,\chi}\right)^{2^{a-1}}$.

Proof. From (6.2), we get

$$\mathbb{E}\left[\mathrm{Re}(\widehat{f}(\boldsymbol{s}'))\right] = \mathrm{Re}\left(\sum_{j=1}^{m}\mathbb{E}\left[\theta_q^{-(\nu_{j,1}\pm\cdots\pm\nu_{j,2^{a-1}})}\right]\right) = \mathrm{Re}\left(\sum_{j=1}^{m}\mathbb{E}\left[\cos\left(\frac{2\pi}{q}\nu_{j,1}\right)\right]^{2^{a-1}}\right) ,$$

using the independence of the noise samples $\nu_{j,\ell}$ and $\mathbb{E}[\theta_q^{\pm\nu_{j,\ell}}] = \mathbb{E}[\cos(2\pi\nu_{j,\ell}/q)]$ (which follows from Lemmas 6.2 and 6.3). Using Lemmas 6.2 and 6.3 again, we have that

$\mathbb{E}[\cos(2\pi\nu_{j,\ell}/q)] \geq R_{\sigma,q,\chi}$. Hence, we get that

$$\mathbb{E}\left[\mathrm{Re}(\widehat{f}(\boldsymbol{s}'))\right] \geq \sum_{j=1}^{m} \left(R_{\sigma,q,\chi}\right)^{2^{a-1}} = m \cdot \left(R_{\sigma,q,\chi}\right)^{2^{a-1}} \ . \tag{6.15}$$

□

Lemma 6.6. *Let $G \subseteq \mathbb{Z}_q$ be a subgroup of $\mathbb{Z}_q$, let $X \xleftarrow{U} G$ and let $e \in \mathbb{Z}_q$ be independent from X. Then, $\mathbb{E}\left[\theta_q^{X+e}\right] = 0$.*

Proof. Define $Y = \frac{2\pi}{q}X$. Then Y is a random variable following a discrete uniform distribution on the unit circle. Then $\mathbb{E}[\theta_q^X] = 0$ follows from the analysis of discrete circular uniform distributions (see e.g. [AP13]). Now, as X and e are independent, $\mathbb{E}[\theta_q^{X+e}] = \mathbb{E}[\theta_q^X]\mathbb{E}[\theta_q^e] = 0$. □

Lemma 6.7. $\arg\max_{\boldsymbol{\alpha}} \mathrm{Re}(\widehat{f}(\boldsymbol{\alpha})) = \boldsymbol{s}'$ *with probability greater than*[1]

$$1 - q^{k'} \cdot \exp\left(-\frac{m}{8} \cdot \left(R_{\sigma,q,\chi}\right)^{2^a}\right) \ .$$

Proof. A similar proof is given for LPN in [BTV15]. We are looking to upper bound the probability that there is some $\boldsymbol{\alpha} \neq \boldsymbol{s}'$ such that $\mathrm{Re}(\widehat{f}(\boldsymbol{\alpha})) \geq \mathrm{Re}(\widehat{f}(\boldsymbol{s}'))$. Using a union bound, we may upper bound this by $q^{k'}$ times the probability that $\mathrm{Re}(\widehat{f}(\boldsymbol{\alpha})) \geq \mathrm{Re}(\widehat{f}(\boldsymbol{s}'))$ for some fixed vector $\boldsymbol{\alpha} \in \mathbb{Z}_q^{k'}$, $\boldsymbol{\alpha} \neq \boldsymbol{s}'$ which is the probability that

$$\sum_{j=1}^{m} \left(\mathrm{Re}\left(\theta_q^{-(\langle \boldsymbol{A}_j, \boldsymbol{s}'\rangle - \boldsymbol{c}_j)}\right) - \mathrm{Re}\left(\theta_q^{-(\langle \boldsymbol{A}_j, \boldsymbol{\alpha}\rangle - \boldsymbol{c}_j)}\right)\right) \leq 0 \ .$$

Let $\boldsymbol{y} = \boldsymbol{\alpha} - \boldsymbol{s}' \in \mathbb{Z}_q^{k'}$. Also, define $e_j \coloneqq \langle \boldsymbol{A}_j, \boldsymbol{s}'\rangle - \boldsymbol{c}_j$, for $1 \leq j \leq m$. Then, $\langle \boldsymbol{A}_j, \boldsymbol{\alpha}\rangle - \boldsymbol{c}_j = \langle \boldsymbol{A}_j, \boldsymbol{y}\rangle + e_j$. Note that as $\boldsymbol{A}_j$ is uniformly distributed at random, independently from e_j, and $\boldsymbol{y}$ is fixed and non-zero, $\langle \boldsymbol{A}_j, \boldsymbol{y}\rangle$ is uniformly distributed in a subgroup of $\mathbb{Z}_q$, and thus so is $\langle \boldsymbol{A}_j, \boldsymbol{\alpha}\rangle - \boldsymbol{c}_j$. Hence, we can apply Lemma 6.6.
From our heuristic assumption, we will consider $X_1, X_2, \ldots, X_m$ to be independent random variables with $X_j = u_j - v_j$, where

$$u_j = \mathrm{Re}\left(\theta_q^{-(\langle \boldsymbol{A}_j, \boldsymbol{s}'\rangle - \boldsymbol{c}_j)}\right) \qquad \text{and} \qquad v_j = \mathrm{Re}\left(\theta_q^{-(\langle \boldsymbol{A}_j, \boldsymbol{\alpha}\rangle - \boldsymbol{c}_j)}\right) \ . \tag{6.16}$$

Note that $X_j \in [-2, 2]$ for all j. Furthermore, let $X = \sum_{j=1}^{m} X_j$. Using Lemmas 6.5 (for the u_j's) and 6.6 (for the v_j's), we get that

$$\mathbb{E}\left[X\right] \geq m \cdot \left(R_{\sigma,q,\chi}\right)^{2^{a-1}} \ . \tag{6.17}$$

[1] A recent result [BV15] seems to show that if we replace the Hoeffding bound with the Central Limit theorem, the factor 1/8 can be replaced by a factor 1/4.

We will bound the probability that $X \leq 0$ using Hoeffding's inequality (Theorem 2.15). Let $t = \mathbb{E}[X] > 0$. Then,

$$\begin{aligned} \Pr[X \leq 0] &= \Pr\left[\,(X - \mathbb{E}[X]) \leq -\mathbb{E}[X]\right] \leq \exp\left(\frac{-2(\mathbb{E}[X])^2}{16m}\right) \\ &\leq \exp\left(-\frac{m}{8} \cdot (R_{\sigma,q,\chi})^{2^a}\right) . \end{aligned} \tag{6.18}$$

Applying the aforementioned union-bound, we get the desired result. □

We are now ready to derive the number of samples m required to recover the correct secret block $\boldsymbol{s}'$ with high probability.

Theorem 6.8. *Let k, q be positive integers and $\Pi_{\boldsymbol{s},\chi}$ be an LWE oracle, where $\boldsymbol{s} \in \mathbb{Z}_q^k$. Let $a \in \mathbb{Z}$ with $1 \leq a \leq k$, let b be such that $ab \leq k$, and let $k' = k - (a-1)b$. Let $\mathcal{A}_{\boldsymbol{s},\chi,a-1}$ be the oracle returning samples $(\boldsymbol{a}_i, c_i)$ where the $\boldsymbol{a}_i$ are zero for all but the first k' elements. Denote the vector consisting of the first k' elements of $\boldsymbol{s}$ as $\boldsymbol{s}'$. Fix an $\epsilon \in (0,1)$. Then, the number of independent samples m^{LWE} from $\mathcal{A}_{\boldsymbol{s},\chi,a-1}$, which are required such that we fail to recover the secret block $\boldsymbol{s}'$ with probability at most ϵ satisfies*

$$m^{LWE} \geq \begin{cases} 8 \cdot k' \cdot \log\left(\frac{q}{\epsilon}\right) \cdot \left(\frac{q}{\pi} \sin\left(\frac{\pi}{q}\right) e^{-2\pi^2\sigma^2/q^2}\right)^{-2^a} & \text{when } \chi = \bar{\Psi}_{\sigma,q} \\ 8 \cdot k' \cdot \log\left(\frac{q}{\epsilon}\right) \cdot \left(1 - \frac{2\pi^2\sigma^2}{q^2}\right)^{-2^a} & \text{when } \chi = D_{\sigma,q} \, . \end{cases}$$

Furthermore, the hypothesis testing phase (the FFT phase in Algorithm 6.1) that recovers $\boldsymbol{s}'$ requires $2m^{LWE} + C_{\mathsf{FFT}} \cdot k' \cdot q^{k'} \cdot \log q$ operations in $\mathbb{C}$ and requires storage for $q^{k'}$ complex numbers, where C_{FFT} is the small constant in the complexity of the FFT.[2]

Proof. For a fixed m, we get

$$\epsilon = \Pr\left[\exists\ \boldsymbol{\alpha} \neq \boldsymbol{s}' \ :\ \mathrm{Re}(\widehat{f}(\boldsymbol{\alpha})) \geq \mathrm{Re}(\widehat{f}(\boldsymbol{s}'))\right] < q^{k'} \cdot \exp\left(-\frac{m}{8} \cdot (R_{\sigma,q,\chi})^{2^a}\right) .$$

Solving for m, we get the desired result.
Concerning the algorithmic and memory complexities, we need to store the values of the function $f(\boldsymbol{x})$ as $q^{k'}$ elements from $\mathbb{C}$. For each of the m^{LWE} samples we receive from $\mathcal{A}_{\boldsymbol{s},\chi,a-1}$, we compute an exponentiation and an addition in $\mathbb{C}$ to update $f(\boldsymbol{x})$ and then discard the sample. Finally, computing the discrete Fourier transform of f can be achieved with $C_{\mathsf{FFT}} \cdot k' \cdot q^{k'} \cdot \log q$ complex operations, and no additional memory, using an in-place FFT algorithm. □

The hypothesis testing part of the algorithm is summarized in Algorithm 6.1.

[2] One might comment on the required precision needed to compute the DFT. For this, we set our precision to $O\left(\log(m(R_{\sigma,q,\chi})^{2^a})\right)$ bits which is the expected size of our highest peak in the DFT. Using this result along with some standard results about the exact complexity to compute a DFT with a given precision (see, e.g., [BSS00]), the ratio between our (binary) complexities and the binary complexities of [ACF+13] remain the same.

Algorithm 6.1 Hypothesis testing algorithm for LWE.

Input: m independent LWE samples with only $k' \coloneqq k-(a-1)b$ non-zero components in $\boldsymbol{a}$. We represent our samples as a matrix $\boldsymbol{A} \in \mathbb{Z}_q^{m\times k'}$ and a vector $\boldsymbol{c} \in \mathbb{Z}_q^m$.

Output: A vector consisting of the k' elements of $\boldsymbol{s}$ that are at the non-zero positions of $\boldsymbol{a}$

1: Compute the fast Fourier Transform $\widehat{f}(\boldsymbol{\alpha})$ of the function $f(\boldsymbol{x})\colon \sum_{j=1}^{m} \mathbb{1}_{\boldsymbol{A}_j=\boldsymbol{x}}\theta_q^{\boldsymbol{c}_j}$

2: **return** $\arg\max_{\boldsymbol{\alpha}\in\mathbb{Z}_q^{k'}} \widehat{f}(\boldsymbol{\alpha})$

6.3.3 Back Substitution

We use a similar back substitution mechanism as the one described in [ACF+13]. Note that we have to apply back substitution on one table less, because we performed only $a-1$ reductions. Furthermore, because we recovered a complete block of $\boldsymbol{s}$, the table T^{a-1} would be completely zeroed-out by back substitution and can therefore simply be dropped after the hypothesis testing phase. Finally, we do not discard the m^{LWE} queries from $\Pi_{\boldsymbol{s},\chi}$, which were reduced and then used for the solving phase. Instead, we store these m^{LWE} original queries and re-use $m' < m^{\mathrm{LWE}}$ of these queries for the next block of $\boldsymbol{s}$.

6.4 Complexity of BKW with Multidimensional DFT on LWE

We now have all the results we need to state the total complexity of solving SEARCH-LWE with our algorithm. For ease of notation, we will consider from here on that the parameters a and b are chosen such that $k = a \cdot b$. Note that the general case, where $k = (a-1)\cdot b + k'$, follows similarly from our previous results.

Theorem 6.9 (Complexity of SEARCH-LWE)**.** *Let k, q be positive integers and $\Pi_{\boldsymbol{s},\chi}$ be an LWE oracle, where $\boldsymbol{s} \in \mathbb{Z}_q^k$. Let $a, b \in \mathbb{N}$ be such that $a \cdot b = k$. Let C_{FFT} be the small constant in the complexity of the fast Fourier transform computation. Let $0 < \epsilon < 1$ be a targeted success rate and define $\epsilon' \coloneqq (1-\epsilon)/a$. For $0 \le j \le a-1$, let*

$$m_{j,\epsilon}^{LWE} \coloneqq \begin{cases} 8\cdot b\cdot \log\left(\frac{q}{\epsilon}\right)\cdot\left(\frac{q}{\pi}\sin\left(\frac{\pi}{q}\right)e^{-2\pi^2\sigma^2/q^2}\right)^{-2^{a-j}} & \text{when } \chi = \bar{\Psi}_{\sigma,q} \\ 8\cdot b\cdot \log\left(\frac{q}{\epsilon}\right)\cdot\left(1-\frac{2\pi^2\sigma^2}{q^2}\right)^{-2^{a-j}} & \text{when } \chi = D_{\sigma,q}\,. \end{cases}$$

Under the standard heuristic that all the samples after reduction are independent (which was also used in the previous work), the time complexity of our algorithm to recover the secret $\boldsymbol{s}$ with probability at least ϵ is $c_1 + c_2 + c_3 + c_4$, where

$$c_1 \coloneqq \left(\frac{q^b-1}{2}\right)\cdot\left(\frac{(a-1)\cdot(a-2)}{2}(k+1) - \frac{b}{6}(a\cdot(a-1)\cdot(a-2))\right) \tag{6.19}$$

is the number of additions in $\mathbb{Z}_q$ to produce all tables T^j, $0 \leq j \leq a-1$,

$$c_2 \coloneqq \sum_{j=0}^{a-1} m_{j,\epsilon'}^{LWE} \cdot \frac{a-1-j}{2} \cdot (k+2) \tag{6.20}$$

is the number of additions in $\mathbb{Z}_q$ to produce the samples required to recover all blocks of $\boldsymbol{s}$ with probability ϵ,

$$c_3 \coloneqq 2\left(\sum_{j=0}^{a-1} m_{j,\epsilon'}^{LWE}\right) + C_{\mathsf{FFT}} \cdot k \cdot q^b \cdot \log(q) \tag{6.21}$$

is the number of operations in $\mathbb{C}$ to prepare and compute the DFTs,and

$$c_4 \coloneqq (a-1) \cdot (a-2) \cdot b \cdot \frac{q^b-1}{2} \tag{6.22}$$

is the number of operations in $\mathbb{Z}_q$ for back substitution.
The number of calls to the oracle $\Pi_{\boldsymbol{s},\chi}$ is

$$(a-1) \cdot \frac{q^b-1}{2} + m_{0,\epsilon}^{LWE}\,. \tag{6.23}$$

Finally, the memory complexity in number of elements from $\mathbb{Z}_q$ and $\mathbb{C}$ are respectively

$$\left(\frac{q^b-1}{2} \cdot (a-1) \cdot \left(k+1-b\frac{a-2}{2}\right)\right) + m_{0,\epsilon}^{LWE} \quad \textit{and} \quad q^b\,. \tag{6.24}$$

Proof. To recover $\boldsymbol{s}$, we need to recover each block of $\boldsymbol{s}$ successfully. Since we are making use of the same set of tables T and reduced queries for each block, these events are not independent. Using a union bound, and a failure probability bounded by $(1-\epsilon)/a$ for each of the a blocks thus leads to a overall success probability of at least ϵ.

- The cost of constructing the set of tables T in (6.19) is given by Lemma 6.1. Note that theses tables are constructed only once and maintained throughout the execution of the algorithm.

- As per Lemma 6.1, the cost of obtaining m samples from the oracle $\mathcal{A}_{\boldsymbol{s},\chi,a-1}$ is upper bounded by $m \cdot \frac{a-1}{2} \cdot (k+2)$. Noting that after solving the jth block, the table T^j is dropped, the result in (6.20) follows.

- The DFT has to be applied a times, for each block of size b. Since the number of samples required is updated for each block, we get equation (6.21).

- After solving the first block, back substitution has to be applied to $a-2$ tables (table T^{a-1} can be dropped). Per table, the substitution has cost $2b$ for each of the $\frac{q^b-1}{2}$ rows. In total, we get a cost of $\sum_{j=1}^{a-2} 2 \cdot b \cdot \left(i \cdot \frac{q^b-1}{2}\right)$, as in (6.22).

- The required number of oracle samples follows from Lemma 6.1. Note that the samples needed to fill up the tables are required only once and that the $m_{0,\epsilon}^{\mathrm{LWE}}$ additional queries are stored and can be reused for each block of $\boldsymbol{s}$ since $m_{0,\epsilon}^{\mathrm{LWE}} > m_{j,\epsilon}^{\mathrm{LWE}}$ for $j > 0$. This gives us the total from (6.23).

- Finally, the storage cost for the tables follows from Lemma 6.1. In addition, we need an array of size q^b to store the complex function on which we apply the DFT (we assume an in-place DFT algorithm requiring no extra storage). We also store the $m_{0,\epsilon}^{\mathrm{LWE}}$ samples queried to solve the first block. Combining these results gives us (6.24).

□

6.5 Reducing the Number of Samples

In this section, we show how we can reduce the number of queries required by our algorithm using two different ideas. The first is due to Lyubashevsky [Lyu05] and the second is due to Levieil and Fouque [LF06].

6.5.1 Lyubashevsky's Idea

If the number of queries to the LWE oracle is limited we can adapt Lyuabshevsky's idea that we presented in the context of LPN in Section 3.1.3. Recall that his idea is to use a universal family of hash function to combine samples to create new ones and that these samples will have higher noise.

Theorem 6.10. *Let $\epsilon \geq (\log q+1)/\log k$. Then, one can convert an LWE instance $\Pi_{\boldsymbol{s},\chi}$ where χ is $\bar{\Psi}_{\sigma,q}$ (resp. $D_{\sigma,q}$) and using $k^{1+\epsilon}$ samples into an LWE instance $\Pi_{\boldsymbol{s},\chi'}$ where χ' is $\bar{\Psi}_{\sigma\lceil(\log q+1)k/(\epsilon\log k)\rceil,q}$ (resp. $D_{\sigma\lceil(\log q+1)k/(\epsilon\log k)\rceil,q}$) without any sample limit.*

Proof (sketch). The proof is exactly the same as in [Lyu05] except for few differences that we state here. We let our samples be $A = \boldsymbol{a}^{(1)}, \cdots \boldsymbol{a}^{(k^{1+\epsilon})} \in \mathbb{Z}_q^k$. Let also $X \subset \{0,1\}^{k^{1+\epsilon}}$ with $x \in X$ if $\sum_j x_j = \lceil(\log(q)+1)k/(\epsilon\log k)\rceil$. We use the following universal family of hash function $H \coloneqq \left\{h_A \colon X \leftarrow \mathbb{Z}_q^k\right\}$ where A is defined above and $h_A(x) \coloneqq x_1 a^{(1)} + \cdots + x_{k^{1+\epsilon}} a^{(k^{1+\epsilon})}$. By the Leftover Hash Lemma (Lemma 3.6), when A and x are uniformly distributed, with probability greater than $1 - 2^{-k/4}$, $\Delta(h_A(x), U) \leq 2^{-n/4}$, where U is the uniform probability distribution over Z_q^k. Note that the Leftover Hash Lemma holds because

$$|X| \geq \left(\frac{k^{1+\epsilon}}{\lceil(\log q+1)k/(\epsilon\log k)\rceil}\right)^{\lceil(\log q+1)k/(\epsilon\log k)\rceil} \geq q^k \ ,$$

when $\epsilon \geq (\log q+1)/\log k$. □

6.5.2 The LF2 Heuristic

In [LF06], Levieil and Fouque propose LF2, an heuristic improvement for the reduction phase of their LPN solving algorithm LF1. The main idea of LF2 is to compute the sum (or difference) of *any* pair of samples $(\boldsymbol{a}, c)$ and $(\boldsymbol{a}', c')$, which agree on b particular coordinates. Thus, in an entry of a reduction table T^i, we would store not only one, but all samples agreeing (up to negation) on b coordinates. Then, when reducing a sample $(\boldsymbol{a}, c)$, we could output $(\boldsymbol{a} \pm \boldsymbol{a}', c \pm c')$ for *each* sample $(\boldsymbol{a}', c')$ in the corresponding table entry. Note that if we have x samples agreeing on b positions, we can output $\binom{x}{2}$ reduced samples.
An interesting case arises when we take exactly $3 \cdot q^b/2$ oracle samples. In the worst case, we get exactly 3 samples per entry in table T^1.[3] Then, applying all the pairwise reductions, we again get $3 \cdot q^b/2$ samples to be stored in table T^2 and so forth. Hence, if we take

$$\max\left\{m_{0,\epsilon'}^{\mathrm{LWE}}, 3 \cdot q^b/2\right\} \tag{6.25}$$

oracle queries, we are ensured to have enough samples for the Fourier transform. We could, thus, solve the LWE problem using fewer oracle samples than in Theorem 6.9 and with a similar time complexity, at the expense of a higher memory complexity (to store multiple samples per table entry).

6.6 Results

We computed the number of operations needed in $\mathbb{Z}_q$ to solve the LWE problem for various values of k when the parameters are chosen according to Regev's cryptosystem [Reg09] and $\epsilon = 0.99$. In this scheme, q is a prime bigger than k^2 and $\sigma = q/(\sqrt{k}\log^2(k)\sqrt{2\pi})$. For our table, we took q to be the smallest prime greater than k^2. Our results are displayed in Table 6.1.[4] To simplify our result, we considered operations over $\mathbb{C}$ to have the same complexity as operations over $\mathbb{Z}_q$. We also took $C_{\mathsf{FFT}} = 1$ which is the best one can hope to obtain for a FFT. Regarding the noise distribution, we obtained the same results for both $D_{\sigma,q}$ and $\bar{\Psi}_{\sigma,q}$. If we compare our results with [ACF+13, Table1], we see that we are better in all the cases.[5] This improvement with respect to log likelihood comes from the fact that we do one reduction less in our reduction phase as we recover a full block instead of a single element in $\mathbb{Z}_q$. This implies that our noise is going to be smaller and, hence, we will need a lower number of queries. However, we still achieve the same asymptotic complexity.

[3] Bogos and Vaudenay did an average case analysis and showed that $2 + q^b$ samples would be enough [BV15].

[4] The code used to compute these value is available on our website `http://lasec.epfl.ch/lwe/`

[5] Albrecht et al. simplified their complexity by considering non-integer a which explains why the difference between our results varies depending on k.

k	q	a	$\log(\#\mathbb{Z}_q)$	$\log(m)$	$\log(m)$ for LF2	$\log(\#\mathbb{Z}_q)$ in [ACF+13]
64	4 099	19	52.62	43.61	41.01	54.85
80	6 421	20	63.23	53.85	51.18	65.78
96	9 221	21	73.72	63.95	61.98	76.75
112	12 547	21	85.86	75.94	73.20	87.72
128	16 411	22	95.03	84.86	82.05	98.67
160	25 601	23	115.87	105.33	102.46	120.43
224	50 177	24	160.34	149.26	146.32	163.76
256	65 537	25	178.74	167.43	164.43	185.35
384	147 457	26	269.18	257.23	254.17	–
512	262 147	27	357.45	345.03	341.92	–

Table 6.1 – We write $\#\mathbb{Z}_q$ for the worst case cost (in operations over $\mathbb{Z}_q$) of solving Search-LWE for various parameters for the Regev cryptosystem [Reg09] when $\epsilon = 0.99$ according to Theorem 6.9. We provide also the value of a that minimizes the complexity, the number of queries (m) according to (6.23), and the number of queries (m) when we apply the LF2 heuristic (6.25).

Chapter 7

A New Algorithm Solving the Learning With Rounding Problem

The work presented in this chapter is part of a joint work with F. Tramèr and Prof. S. Vaudenay and was published in [DTV15].

Notation. *In this chapter, we define* $\sqrt{-1} = i \in \mathbb{C}$.

We try now to adapt the algorithm presented in Chapter 6 to the Learning With Rounding problem (see Section 3.4). More precisely, we present the *first algorithmic analysis of the LWR problem* when q is prime. While our proposal requires a subexponential number of samples, our detailed analysis contains many results of independent interest. In the remaining of this chapter, we will always consider q to be *prime*.

Lemma 7.1. *Let k and $q > p \geq 2$ be positive integers, q prime. Let $(\boldsymbol{a}, c)$ be a random sample from an LWR oracle $\Lambda_{\boldsymbol{s},p}$. Then, the "rounding error", given by $\xi = (p/q)\langle \boldsymbol{a}, \boldsymbol{s}\rangle - c$, follows a uniform distribution in a discrete subset of $[-1/2, 1/2]$ with mean zero. Furthermore, for $\gamma \in \mathbb{R}_{\neq 0}$,*

$$\mathbb{E}\left[e^{\pm i\xi\gamma}\right] = \frac{1}{q} \cdot \frac{\sin(\frac{\gamma}{2})}{\sin(\frac{\gamma}{2q})} \,. \tag{7.1}$$

Proof. We first prove the first part of the lemma. We will prove that for any $\alpha \in [\frac{-q+1}{2}, \ldots, \frac{q-1}{2}]$, ξ takes the value α/q with probability $1/q$. We have $p \cdot \langle \boldsymbol{a}, \boldsymbol{s}\rangle \equiv \xi q \pmod q$. So, $\alpha = \xi q = ((p \cdot \langle \boldsymbol{a}, \boldsymbol{s}\rangle + (q-1)/2) \bmod q) - (q-1)/2$. As $\langle \boldsymbol{a}, \boldsymbol{s}\rangle$ is uniform in $\mathbb{Z}_q$ (for $\boldsymbol{s} \neq 0$), α is uniform in $-(q+1)/2, \cdots, (q-1)/2$ and has mean zero. Hence, so has ξ.

We now prove the second part of our lemma. Let $X = q \cdot \xi$ be a random variable following a discrete uniform distribution on the set of integers $\{(-q+1)/2, \ldots, (q-1)/2\}$. Then,

from the characteristic function of X, for any $t \in \mathbb{R}$ we have

$$\mathbb{E}\left[e^{itX}\right] = \frac{e^{-it(q-1)/2} - e^{it(q+1)/2}}{q \cdot (1 - e^{it})} \ . \tag{7.2}$$

By simple arithmetic, we obtain

$$\mathbb{E}\left[e^{i\xi\gamma}\right] = \mathbb{E}\left[e^{i\gamma q^{-1}X}\right] = \frac{e^{i\gamma/(2q)}\left(e^{-i\gamma/2} - e^{i\gamma/2}\right)}{q\left(1 - e^{i\gamma/q}\right)} = \frac{-\sin(\gamma/2) \cdot 2i}{q\left(e^{-\gamma i/(2q)} - e^{\gamma i/(2q)}\right)}$$

which gives our result. □

In our case, q is an odd prime and different from p. Hence, $\mathbb{E}[e^{i\xi\gamma}]$tends to $\frac{2}{\gamma}\sin(\gamma/2)$ as q grows to infinity. We will be interested in the value $\gamma = 2\pi/p$. Then, for small $p = \{2, 3, 4, 5, \ldots\}$, $\mathbb{E}[e^{i\xi\gamma}]$ is $\{0.6366, 0.8270, 0.9003, 0.9355, \ldots\}$.

7.1 The LWR-solving Algorithm

From the similarity of the LWR and LWE problems, it should not seem surprising that we would use the same sample reduction and back substation phases, but we need an alternative "hypothesis testing phase" (which we call *solving phase*) to account for the difference in error distributions.

As for LWE, we choose some $a, b \leq k$ such that $ab \leq k$ and we let $k' = k - (a-1)b$. We will view the reduction phase of our algorithm as producing a series of oracles $\mathcal{B}_{\boldsymbol{s},p,\ell}$ for $0 \leq \ell \leq a-1$, where $\mathcal{B}_{\boldsymbol{s},p,0}$ is the original LWR oracle $\Lambda_{\boldsymbol{s},p}$. The final oracle $\mathcal{B}_{\boldsymbol{s},p,a-1}$ produces samples $(\boldsymbol{a}, c)$ where $\boldsymbol{a}$ is non-zero only on the first k' elements.

Solving Phase

We consider the samples from $\mathcal{B}_{\boldsymbol{s},p,a-1}$ as belonging to $\mathbb{Z}_q^{k'} \times \mathbb{Z}_p$. We assume we have m such samples and represent them as a matrix $\boldsymbol{A} \in \mathbb{Z}_q^{m \times k'}$ with rows $\boldsymbol{A}_i$ and a vector $\boldsymbol{c} \in \mathbb{Z}_p^m$. The corresponding block of k' elements of the secret $\boldsymbol{s}$ is denoted $\boldsymbol{s}'$.

Additionally, we assume that each sample $(\boldsymbol{a}^{(j)}, c^{(j)})$ from $\mathcal{B}_{\boldsymbol{s},p,a-1}$ is the sum of 2^{a-1} samples (or their negation) from the LWR oracle. The 'noise' $\langle \boldsymbol{a}^{(j)}, \boldsymbol{s}' \rangle \frac{p}{q} - c^{(j)}$ will then correspond to the sum of 2^{a-1} independent "rounding errors" (or their negation) from the original samples.

For $\theta_u \coloneqq \exp(2\pi i/u)$, we consider the function

$$f_{\mathsf{lwr}}(\boldsymbol{x}) \coloneqq \sum_{j=1}^{m} \mathbb{1}_{\{\boldsymbol{A}_j = \boldsymbol{x}\}} \, \theta_p^{\boldsymbol{c}_j} \ , \quad \forall \boldsymbol{x} \in \mathbb{Z}_q^{k'} \ . \tag{7.3}$$

The discrete Fourier transform of f_{lwr} is

$$\widehat{f}_{\mathsf{lwr}}(\boldsymbol{\alpha}) \coloneqq \sum_{\boldsymbol{x}\in\mathbb{Z}_q^{k'}} f_{\mathsf{lwr}}(\boldsymbol{x})\theta_q^{-\langle \boldsymbol{x},\boldsymbol{\alpha}\rangle} = \sum_{j=1}^{m} \theta_p^{-(\langle \boldsymbol{A}_j,\boldsymbol{\alpha}\rangle \frac{p}{q} - \boldsymbol{c}_j)} \ . \tag{7.4}$$

In particular, note that

$$\widehat{f}_{\mathsf{lwr}}(\boldsymbol{s}') = \sum_{j=1}^{m} \theta_p^{-(\langle \boldsymbol{s}',\boldsymbol{A}_j\rangle \frac{p}{q} - \boldsymbol{c}_j)} = \sum_{j=1}^{m} \theta_p^{-(\pm\xi_{j,1}\pm\cdots\pm\xi_{j,2^{a-1}})} \ , \tag{7.5}$$

where the $\xi_{j,\ell}$ are the independent rounding errors from the original LWR samples. Note that it is irrelevant whether the noise has been reduced modulo p, since $\theta_p^{-up} = 1$ for $u \in \mathbb{Z}$.
As for LWE, we can now derive an explicit formula for the number of samples m, which are required to recover $\boldsymbol{s}'$ with high probability.

Lemma 7.2. *For $q > p \geq 2$, q prime,* $\mathbb{E}\left[\mathrm{Re}(\widehat{f}_{\mathsf{lwr}}(\boldsymbol{s}'))\right] = m \cdot \left(\frac{1}{q} \cdot \frac{\sin(\frac{\pi}{p})}{\sin(\frac{\pi}{pq})}\right)^{2^{a-1}}$.

Proof. Let ξ be the random variable defined in Lemma 7.1. Because the original rounding errors are independent, using Lemma 7.1, we may write

$$\mathbb{E}\left[\mathrm{Re}(\widehat{f}_{\mathsf{lwr}}(\boldsymbol{s}'))\right] = m \cdot \mathrm{Re}\left(\mathbb{E}\left[e^{\mp i\xi\frac{2\pi}{p}}\right]^{2^{a-1}}\right) = m \cdot \left(\frac{1}{q} \cdot \frac{\sin(\frac{\pi}{p})}{\sin(\frac{\pi}{pq})}\right)^{2^{a-1}} \ . \tag{7.6}$$

□

We need also to bound the values of $\widehat{f}$ when not evaluated at $\boldsymbol{s}'$.

Lemma 7.3. *Let $\boldsymbol{\alpha} \neq \boldsymbol{s}'$. Then*

$$\mathbb{E}\left[\mathrm{Re}(\widehat{f}_{\mathsf{lwr}}(\boldsymbol{\alpha}))\right] \leq m\left(\frac{2}{p} + \frac{1}{p}\cos\left(\frac{\pi}{p}\right)\right)^{2^{a-1}} \leq m\left(\frac{3}{p}\right)^{2^{a-1}} \ .$$

Proof. Like in the previous lemma, we can write, for $\boldsymbol{a}$ uniformly distributed,

$$\mathbb{E}\left[\mathrm{Re}(\widehat{f}_{\mathsf{lwr}}(\boldsymbol{\alpha}))\right] = m \cdot \mathrm{Re}\left(\mathbb{E}\left[e^{\mp i(2\pi\langle \boldsymbol{a},\boldsymbol{\alpha}\rangle/q - 2\pi \boldsymbol{c}/p)}\right]^{2^{a-1}}\right) \ . \tag{7.7}$$

However, unlike in the LWE case, we cannot use the independence of $\boldsymbol{a}$ and the noise to obtain a zero expected value. This occurs because the errors are computed deterministically from the vectors $\boldsymbol{a}$ in LWR. In fact, experiments showed that the error is strongly correlated to $\boldsymbol{a}$ and that the expected value is not zero. Thus, we will instead bound this expected value. To do this, we write

$$\mathbb{E}\left[e^{\mp i(2\pi\langle \boldsymbol{a},\boldsymbol{\alpha}\rangle/q - 2\pi c/p)}\right] = \mathbb{E}\left[\cos\left(\frac{2\pi\langle \boldsymbol{a},\boldsymbol{\alpha}\rangle}{q} - \frac{2\pi c}{p}\right)\right] \pm i \cdot \mathbb{E}\left[\sin\left(-\frac{2\pi\langle \boldsymbol{a},\boldsymbol{\alpha}\rangle}{q} + \frac{2\pi c}{p}\right)\right]$$

and we bound both the sine and the cosine term.

- We first show that the contribution of the sine is zero, i.e., that for $\boldsymbol{\alpha} \neq \boldsymbol{s}'$ fixed,[1]

$$\mathbb{E}\left[\sin\left(2\pi\langle \boldsymbol{a}, \boldsymbol{\alpha}\rangle/q - 2\pi c/p\right)\right] = 0 \ . \tag{7.8}$$

  Let $w(\boldsymbol{a}) \coloneqq \sin\left(2\pi\langle \boldsymbol{a}, \boldsymbol{\alpha}\rangle/q - 2\pi\lceil\langle \boldsymbol{a}, \boldsymbol{s}'\rangle(p/q)\rfloor/p\right)$. First, note that for $\boldsymbol{a} = \boldsymbol{0}$, $c = 0$. For $\boldsymbol{a} \neq \boldsymbol{0}$, the contribution in the expected value is $w(\boldsymbol{a})$. We have

$$\begin{aligned} w(-\boldsymbol{a}) &= \sin\left(2\pi\langle -\boldsymbol{a}, \boldsymbol{\alpha}\rangle/q - 2\pi\lceil\langle -\boldsymbol{a}, \boldsymbol{s}'\rangle(p/q)\rfloor/p\right) \\ &= \sin\left(-2\pi\langle \boldsymbol{a}, \boldsymbol{\alpha}\rangle/q - 2\pi\lceil-\langle \boldsymbol{a}, \boldsymbol{s}'\rangle(p/q)\rfloor/p\right) = -w(\boldsymbol{a}) \ . \end{aligned}$$

  As q is odd, $-\boldsymbol{a} \neq \boldsymbol{a}$ and, thus, in the expected value, the contribution of any $\boldsymbol{a} \neq 0$ is cancelled. Hence, the result.

- For the cosine, as in Lemma 6.7, we let $\boldsymbol{y} = \boldsymbol{\alpha} - \boldsymbol{s}' \in \mathbb{Z}_q^{k'}$. We get,

$$\begin{aligned} \cos\left(\frac{2\pi\langle \boldsymbol{a}, \boldsymbol{\alpha}\rangle}{q} - \frac{2\pi c}{p}\right) &= \cos\left(\frac{2\pi\langle \boldsymbol{a}, \boldsymbol{y}\rangle}{q} + \frac{2\pi(\langle \boldsymbol{a}, \boldsymbol{s}'\rangle p/q - c)}{p}\right) \\ &= \cos\left(\frac{2\pi\langle \boldsymbol{a}, \boldsymbol{y}\rangle}{q} + \frac{2\pi\xi}{p}\right) \ , \end{aligned} \tag{7.9}$$

  where $\xi \in [-1/2, 1/2]$ is the rounding error from Lemma 7.1. We are looking for an upper-bound and, hence, we assume that $\xi \in [-1/2, 1/2]$ will always be such that $\cos(2\pi\langle \boldsymbol{a}, \boldsymbol{y}\rangle/q + 2\pi\xi/p)$ is maximized. Figure 7.1 might help with the reading. We divide the circle into sets of the form

$$\mathcal{S}_\ell \coloneqq \left[\frac{\ell\pi}{p}, \frac{(\ell+1)\pi}{p}\right] \cup \left[\frac{-\ell\pi}{p}, \frac{-(\ell+1)\pi}{p}\right] \ , \quad \ell \in [0, p-1] \ .$$

  Note that this covers the whole circle. The hashed surface in Figure 7.1 is such a set.

  When $2\pi\langle \boldsymbol{a}, \boldsymbol{y}\rangle/q \in \mathcal{S}_\ell$ for $\ell \neq 0$, we upper-bound (7.9) by $\cos((\ell-1)\pi/p)$ (the bold line in Figure 7.1). Indeed, $|2\pi\xi/p| \leq \pi/p$. When $2\pi\langle \boldsymbol{a}, \boldsymbol{y}\rangle/q \in \mathcal{S}_0$, we upper-bound (7.9) by $\cos(0) = 1$.

  Note that $\Pr[2\pi\langle \boldsymbol{a}, \boldsymbol{y}\rangle/q \in \mathcal{S}_\ell]] = 1/p$ because $\langle \boldsymbol{a}, \boldsymbol{y}\rangle$ is uniformly distributed in $\mathbb{Z}_q$ and $p \leq q$. Hence,

$$\begin{aligned} \mathbb{E}\left[\cos\left(2\pi\langle \boldsymbol{a}, \boldsymbol{y}\rangle + 2\pi\xi/p\right)\right] &\leq \frac{1}{p} + \frac{1}{p}\sum_{\ell=1}^{p-1} \cos\left(\frac{(\ell-1)\pi}{p}\right) \\ &= \frac{1}{p} + \frac{1}{p}\cos(0) - \frac{1}{p}\cos\left(\frac{(p-1)\pi}{p}\right) = \frac{2}{p} + \frac{1}{p}\cos\left(\frac{\pi}{p}\right) \leq \frac{3}{p} \ . \end{aligned} \tag{7.10}$$

[1] This is where the round function instead of the floor function in the definition of LWR becomes handy.

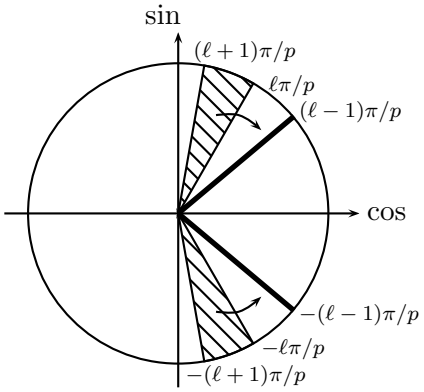


Figure 7.1 – Figure for the proof of Lemma 7.3.

Plugging the values of the sine and the upper-bound for the cosine in (7.7) finishes the proof. □

Lemma 7.4. *When $q > p \geq 4$ and q is prime,* $\arg\max_{\boldsymbol{\alpha}} \mathrm{Re}(\widehat{f}_{\mathsf{lwr}}(\boldsymbol{\alpha})) = \boldsymbol{s}'$ *with probability greater than*[2]

$$1 - q^{k'} \cdot \exp\left(-\frac{m}{8} \cdot \left(\left(\frac{1}{q} \cdot \frac{\sin(\frac{\pi}{p})}{\sin(\frac{\pi}{pq})}\right)^{2^{a-1}} - \left(\frac{3}{p}\right)^{2^{a-1}}\right)^2\right) .$$

Proof. We first want the probability that $\mathrm{Re}(\widehat{f}(\boldsymbol{x})) \geq \mathrm{Re}(\widehat{f}(\boldsymbol{s}'))$ for some fixed vector $\boldsymbol{x} \in \mathbb{Z}_q^{k'}$, $\boldsymbol{x} \neq \boldsymbol{s}'$. Applying the same heuristic argument as for LWE, we consider $X_1, X_2, \ldots, X_m$ to be independent random variables with $X_j = u_j - v_j$, where

$$u_j = \mathrm{Re}\left(\theta_p^{-(\langle \boldsymbol{A}_j, \boldsymbol{s}'\rangle \frac{p}{q} - \boldsymbol{c}_j)}\right) \qquad \text{and} \qquad v_j = \mathrm{Re}\left(\theta_p^{-(\langle \boldsymbol{A}_j, \boldsymbol{x}\rangle \frac{p}{q} - \boldsymbol{c}_j)}\right) . \tag{7.11}$$

Note that $X_j \in [-2, 2]$ for all j. Furthermore, let $X = \sum_{j=1}^{m} X_j$. Using Lemmas 7.2 and 7.3, we get that

$$\mathbb{E}[X] \geq m \cdot \left(\left(\frac{1}{q} \cdot \frac{\sin(\frac{\pi}{p})}{\sin(\frac{\pi}{pq})}\right)^{2^{a-1}} - \left(\frac{3}{p}\right)^{2^{a-1}}\right) \geq 0 . \tag{7.12}$$

We will again bound the probability that $X \leq 0$ using Hoeffding's inequality. Let

[2] Again, a recent result [BV15] seem to show that if we replace the Hoeffding bound with the Central Limit theorem, the factor 1/8 can be replaced by a factor 1/4.

$t = \mathbb{E}[X] > 0$. Then,

$$\Pr[X \leq 0] = \Pr\left[\,(X - \mathbb{E}[X]) \leq -\mathbb{E}[X]\right] \leq \exp\left(\frac{-2(\mathbb{E}[X])^2}{16m}\right)$$
$$\leq \exp\left(-\frac{m}{8} \cdot \left(\left(\frac{1}{q} \cdot \frac{\sin(\frac{\pi}{p})}{\sin(\frac{\pi}{pq})}\right)^{2^{a-1}} - \left(\frac{3}{p}\right)^{2^{a-1}}\right)^2\right) . \tag{7.13}$$

The final result follows by applying a union bound over all possible values of $\boldsymbol{x}$. □

As for LWE, we may now deduce the number m of reduced samples that are required to recover a block $\boldsymbol{s}'$.

Theorem 7.5. *Let k and $q > p \geq 4$ be positive integers, q prime, and $\Lambda_{\boldsymbol{s},p}$ be an LWR oracle, where $\boldsymbol{s} \in \mathbb{Z}_q^k$. Let $a \in \mathbb{Z}$ with $1 \leq a \leq k$, let b be such that $ab \leq k$, and let $k' = k - (a-1)b$. Let $\mathcal{B}_{\boldsymbol{s},p,a-1}$ be the oracle returning samples $(\boldsymbol{a}_i, c_i)$ where the $\boldsymbol{a}_i$ are zero for all but the first k' elements. Denote the vector consisting of the first k' elements of $\boldsymbol{s}$ as $\boldsymbol{s}'$. Fix an $\epsilon \in (0,1)$. Then, the number of samples m from $\mathcal{B}_{\boldsymbol{s},p,a-1}$, which are required such that we fail to recover the secret block $\boldsymbol{s}'$ with probability at most ϵ satisfies*

$$m^{LWR} \geq 8 \cdot k' \cdot \log\left(\frac{q}{\epsilon}\right) \cdot \left(\left(\frac{1}{q} \cdot \frac{\sin(\frac{\pi}{p})}{\sin(\frac{\pi}{pq})}\right)^{2^{a-1}} - \left(\frac{3}{p}\right)^{2^{a-1}}\right)^{-2} .$$

Furthermore, recovering $\boldsymbol{s}'$ in the solving phase (the FFT phase) requires $2m^{LWR} + C_{\mathsf{FFT}} \cdot k' \cdot q^{k'} \cdot \log q$ operations in $\mathbb{C}$, as well as storage for $q^{k'}$ complex numbers.

7.2 Complexity of BKW with multidimensional DFT on LWR

We now summarize the complexity of our algorithm in the following theorem (the proof of which is analogous to the proof of Theorem 6.9).

Theorem 7.6 (Complexity of SEARCH-LWR)**.** *Let k, q be positive integers and $\Lambda_{\boldsymbol{s},p}$ be an LWR oracle, where $\boldsymbol{s} \in \mathbb{Z}_q^k$. Let $a, b \in \mathbb{N}$ be such that $a \cdot b = k$. Let C_{FFT} be the small constant in the complexity of the fast Fourier transform computation. Let $0 < \epsilon < 1$ be a targeted success rate and define $\epsilon' := (1-\epsilon)/a$. For $0 \leq j \leq a-1$, let*

$$m_{j,\epsilon}^{LWR} := 8 \cdot b \cdot \log\left(\frac{q}{\epsilon}\right) \cdot \left(\left(\frac{1}{q} \cdot \frac{\sin(\frac{\pi}{p})}{\sin(\frac{\pi}{pq})}\right)^{2^{a-1-j}} - \left(\frac{3}{p}\right)^{2^{a-1-j}}\right)^{-2} .$$

Under the standard heuristic that all the samples after reduction are independent (which was also used in the previous work), the time complexity of our algorithm to recover the

secret $\boldsymbol{s}$ *with probability at least* ϵ *is* $c_1 + c_2 + c_3 + c_4$*, where*

$$c_1 \coloneqq \left(\frac{q^b - 1}{2}\right) \cdot \left(\frac{(a-1)\cdot(a-2)}{2}(k+1) - \frac{b}{6}(a\cdot(a-1)\cdot(a-2))\right) \tag{7.14}$$

is the number of additions in $\mathbb{Z}_q$ *to produce all tables* T^j*,* $0 \leq j \leq a-1$*,*

$$c_2 \coloneqq \sum_{j=0}^{a-1} m_{j,\epsilon'}^{LWR} \cdot \frac{a-1-j}{2} \cdot (k+2) \tag{7.15}$$

is the number of additions in $\mathbb{Z}_q$ *to produce the samples required to recover all blocks of* $\boldsymbol{s}$ *with probability* ϵ*,*

$$c_3 \coloneqq 2\left(\sum_{j=0}^{a-1} m_{j,\epsilon'}^{LWR}\right) + C_{\mathsf{FFT}} \cdot k \cdot q^b \cdot \log(q) \tag{7.16}$$

is the number of operations in $\mathbb{C}$ *to prepare and compute the DFTs,and*

$$c_4 \coloneqq (a-1)\cdot(a-2)\cdot b \cdot \frac{q^b-1}{2} \tag{7.17}$$

is the number of operations in $\mathbb{Z}_q$ *for back substitution.*
The number of calls to the oracle $\Lambda_{\boldsymbol{s},p}$ *is*

$$(a-1)\cdot\frac{q^b-1}{2} + m_{0,\epsilon}^{LWR}\,. \tag{7.18}$$

Finally, the memory complexity in number of elements from $\mathbb{Z}_q$ *and* $\mathbb{C}$ *are respectively*

$$\left(\frac{q^b-1}{2}\cdot(a-1)\cdot\left(k+1-b\frac{a-2}{2}\right)\right) + m_{0,\epsilon}^{LWR} \quad \textit{and} \quad q^b\,. \tag{7.19}$$

7.3 Results

The current hardness results for LWR require either a parameter q exponential in k or a bound m on the number of oracle samples that an adversary may query. It is an open problem ([AKPW13]) to assess the hardness of LWR with polynomial parameters when the adversary has no sample limit. In such a case, for $a = O(\log k)$ and $b = \lceil k/a \rceil$, our algorithm would solve LWR in time $2^{O(k)}$, as for LWE.
However, the bound on the number of oracle samples in Theorem 3.18 is much lower than the amount of samples required by our algorithm. Using an idea from Lyubashevsky [Lyu05], we can generate additional samples with higher noise (see Theorem 6.10). Yet, even this method requires at least $k^{1+\epsilon}$ samples for $\epsilon \geq (\log q + 1)/\log k$, which is incompatible with the constraints of Theorem 3.18, for a q polynomial in k.
In [AKPW13, Corollary 4.2], two types of parameters are given: parameters maximizing

k	q	p	a	$\log(\#\mathbb{Z}_q)$	$\log(m)$	type
64	383 056 211	733	23	92.20	82.80	(a)
80	1 492 443 083	1 151	25	110.91	101.11	(a)
96	$\approx 2^{32}$	1 663	26	132.26	122.15	(a)
112	$\approx 2^{33}$	2 287	28	148.08	137.68	(a)
128	$\approx 2^{34}$	3 023	29	167.52	156.87	(a)
64	9 461	13	12	81.61	72.90	(b)
80	14 867	13	12	103.89	94.86	(b)
96	21 611	13	12	126.97	117.66	(b)
112	29 717	13	13	140.21	130.60	(b)
128	39 241	13	13	162.63	152.84	(b)

Table 7.1 – Worst case cost (in operations over $\mathbb{Z}_q$) of solving Search-LWR for various parameters for the Regev cryptosystem [Reg09] when $\epsilon = 0.99$ according to Theorem 7.6. We provide also the value of a that minimizes the complexity, the number of queries (m) according to (6.23).

efficiency (a) and parameters minimizing the Modulus/Error ratio (b). For completeness, we show in Table 7.1 the complexity of our algorithm applied to these parameters. More precisely, we took for the underlying LWE problem Regev's parameters and *ignored the constrains on the number of samples*. For the type (a) parameters, we took

$$\sigma = \frac{k^2}{\sqrt{k}\log^2(k)\sqrt{2\pi}} \qquad q = \mathsf{nextprime}(\lceil (2\sigma k)^3 \rceil) \qquad p = \mathsf{nextprime}(\lceil \sqrt[3]{q} \rceil)$$

and for the type (b) parameters

$$\sigma = \frac{k^2}{\sqrt{k}\log^2(k)\sqrt{2\pi}} \qquad p = 13 \qquad q = \mathsf{nextprime}(\lceil 2\sigma k p \rceil) \ .$$

Table 7.1 shows that the parameters given in [AKPW13] seem secure even if we remove the constrain on the number of samples as the complexities are still quite high.

Part II

Leakage-resilient Cryptography

Chapter 8

Introduction to Leakage-resilient Cryptography

The study of side-channel attacks started to receive a lot of attention with the work of Kocher [Koc96, KJJ99] in the nineties. At Crypto'96 [Koc96], Kocher showed how one could completely break the RSA cryptosystem [RSA78] and the Diffie-Hellman key-exchange protocol [DH76] by simply performing a *timing attack*. A timing attack requires the attacker to measure the time taken by an implementation to perform some operations. From this information, one can recover secret information and eventually break the system. An example of such an attack is the timing attack against RSA proposed by Kocher [Koc96] against the square-and-multiply algorithm used to perform modular exponentiation, an operation required in the RSA decryption algorithm. In a naive implementation of the square-and-multiply algorithm, the Hamming weight of the exponent will impact the duration of the algorithm, thus, leaking some information. Combining this with some knowledge on the ciphertext allowed Kocher to completely recover the secret key.

In the same paper [Koc96], Kocher mentions the fact that the study of power consumption would similarly allow to break the cryptosystems. In Crypto'99 [KJJ99], together with Jaffe and Jun, he introduced the concept of *differential power analysis* in which they study in details how the analysis of the power consumption might lead to devastating attacks.

At that point, the popularity of side-channel attacks rose and many new ways of physically attacking an implementation appeared, including acoustic attacks (e.g., [GST13]), or electromagnetic attacks (e.g., [QS01]).

In parallel of the search of new side-channel attacks, a large body of both applied and theoretical research tried to model the information an adversary obtains from the leakage and design countermeasures to prevent such attacks [CJRR99, ISW03, MR04, AGV09, DP08, SVCO$^+$10, SPY13]. In this thesis, we focus on the masking countermeasure which is one of the most studied countermeasure. More specifically, we focus on *additive*

masking.

8.1 The Masking Countermeasure

A large body of work on cryptographic engineering has developed countermeasures to defeat side-channel attacks (see, e.g., [MOP07] for an overview). While many countermeasures are specifically tailored to protect particular cryptographic implementations (e.g., key updates or shielded hardware), a method that generically works for most cryptographic schemes is *masking* [GP99, BGK04, OMPR05, SVCO+10].
The basic idea of a masking scheme is to secret share all sensitive information, including the secret key and all intermediate values that depend on it, thereby making the leakage independent of the secret data. The main issue that occurs when designing masking techniques is to be able to use these masks to compute on the encoded data while still ensuring that all the intermediate values are still protected. Below, we shortly discuss about the most common masking schemes.

8.1.1 Boolean Masking

The most prominent masking scheme is the *Boolean masking*: a bit b is encoded into an d-bit random bit string in the following way:

1. First $d-1$ bits $b_1, \dots b_{d-1}$ are drawn independently and uniformly at random.
2. Finally, we let $b_d \coloneqq b_1 \oplus \cdots \oplus b_{d_1} \oplus b$.

That way, we know that $b = b_1 \oplus \cdots \oplus b_d$ and that the knowledge of $d-1$ bits leaks no information about the encoded bit b. Note that this additive secret sharing technique can easily be extended to any group G by replacing the XORs with the group operation and taking some inverses. Such an encoding then trivially protects against so-called $(d-1)$-*threshold-probing* adversaries that we will define more formally in Section 8.2. Roughly, in our example, such an adversary is allowed to learn the value of $d-1$ wires (which in our case correspond to bits) and, hence, learning no information about the encoded bit b.

8.1.2 Additive Masking

Boolean masking trivially generalizes to the *additive masking scheme*. Let G be a group with additive notation. In this scheme the idea is the same as in the previous section except that instead of encoding a single bit, a group element x is split into d random group elements in the following way:

1. First $d-1$ group elements $x_1, \dots x_{d-1}$ are drawn independently and uniformly at random.
2. Finally, we let $x_d \coloneqq x_1 \oplus \cdots \oplus x_{d_1} \oplus x$.

That way, we know that $x = x_1 \oplus \cdots \oplus x_d$ and that the knowledge of $d-1$ group elements leaks no information about the encoded element x. Note that in most of the literature, G will have a field structure which will be useful when performing computations. The following chapters will be dedicated exclusively to this masking scheme.

8.1.3 Inner-product Masking

The idea of *inner-product masking* was first introduced by Davì, Dziembowski, and Venturi for Leakage-Resilient storage [DDV10]. It was then further studied in the work of Dziembowski and Faust [DF11]. In this scheme, the secret is split in two separate vectors that are supposed to *leak independently*. More precisely, let $\mathbb{F}$ be a finite field, let $x \in \mathbb{F}$ be an element to encode, and let $n \in \mathbb{N}$ be a parameter. The scheme works in the following way:

1. First, draw a random vector L in $\mathbb{F}^n \setminus \{0\}^n$.
2. Then, sample $R \in \mathbb{F}^n$ at random such that $\langle L, R\rangle = x$, where $\langle L, R\rangle$ denotes the inner-product between L and R. A simple way to sample R is to sample $R_1, \ldots, R_{n-1}$ independently at random and fix R_n such that $\langle L, R\rangle = x$.[1]

A recent result [BFG15] shows that the inner-product masking is more secure than the additive masking shown above but slightly slower.

8.2 Modeling the Leakage

A first step, while formalizing side-channel attacks, is to find a model for the leakage. For this, one has to find a model that represents well the leakage with respect to what an attacker would see when performing his attack. On the other hand, this leakage should be restrictive to allow to prove some results. Indeed, assume that we have a secret value s and that we let the leakage be any function f applied on s. If we do not add any restriction, then it is easy to show impossibility of leakage-resilient constructions in this model, as f could just reveal s with probability one. In such a model, we would have to assume that an adversary can read the value of any secret, which trivially breaks any system.
In this thesis, we will stick to the *noisy-leakage model* in which the adversary sees only noisy versions of each wire, to the *random-probing model* in which an adversary recovers an intermediate value with some probability, and to the *threshold-probing model* in which the adversary is allowed to see only a limited number of intermediate values.
We will model a system with a set of values $(x_1, \ldots, x_\ell) \in \mathcal{X}^\ell$, where $\mathcal{X}$ is a finite set and $\ell \in \mathbb{N}$. One can see the x_i's as the set of wires in the system that the adversary can probe. As explained above, an adversary $\mathcal{A}$ will only be able to obtain some partial information about $(x_1, \ldots, x_\ell)$. Note that we do not specify the computational power of

[1] We assume here, wlog, that $L_n \neq 0$.

$\mathcal{A}$ as the definitions below make sense for both computationally-bounded and infinitely powerful $\mathcal{A}$.

8.3 Threshold Probing Model

The threshold probing model can be seen as an adversary having at its disposal a limited number of probes that can be used on wires of his choice. We define now formally the adversary's capability in this model.

Definition 8.1 (t-threshold-probing adversary)**.** *For $t = 0, \ldots, \ell$ a t-*threshold-probing adversary on $\mathcal{X}^\ell$ *is an algorithm $\mathcal{A}$ that plays the following scenario against an oracle that knows $(x_1, \ldots, x_\ell) \in \mathcal{X}^\ell$:*

1. *$\mathcal{A}$ specifies a set $\mathcal{I} = \{i_1, \ldots, i_{|\mathcal{I}|}\} \subseteq \{1, \ldots, \ell\}$ such that $|\mathcal{I}| \leq t$,*
2. *$\mathcal{A}$ receives $(x_{i_1}, \ldots, x_{i_{|\mathcal{I}|}})$ and outputs some value denoted by $\mathsf{out}_{\mathcal{A}}(x_1, \ldots, x_\ell)$.*[2]

Note that in most of the previous work, this model is simply denoted as "probing model". We decided to append the word "threshold" to distinguish it with the random probing model defined in the next subsection. This model is the easiest to work with when trying to prove security. It is in this model that Ishai, Sahai, and Wagner proved their seminal work [ISW03]. In there, Ishai et al. construct a *circuit compiler* that transforms a circuit in a leakage-resilient one. More precisely, a circuit compiler takes as input the description of a cryptographic scheme C with secret key K, e.g., a circuit that describes a block cipher, and outputs a transformed circuit C' and corresponding key K'. The circuit $C'[K']$ shall implement the same functionality as C running with key K, but additionally is resilient to certain well-defined classes of leakage. Notice that while the framework of [ISW03] talks about circuits the same approach applies to software implementations, and we only follow this notation to abstract our description. We will describe in more details their construction in Section 9.6.2.

8.4 Random Probing Model

We will use this model as an intermediate model in our proof. Before formally defining our adversary in this model, we define ϵ-identity functions.

Definition 8.2 (ϵ-identity function)**.** *A randomized function $\varphi : \mathcal{X} \to \mathcal{X} \cup \{\perp\}$ is an* ϵ-identity function *if for every x we have that either $\varphi(x) = x$ or $\varphi(x) = \perp$ and $\Pr[\varphi(x) \neq \perp] = \epsilon$, where the probability is taken over the random coins of φ.*

We use the special symbol $\perp$ to denote that the adversary was unable to make a successful probe. We can now formally define our adversary:

[2] The goal of $\mathcal{A}$ will typically be to derive some information about the encoded value.

Definition 8.3 (ϵ-random-probing adversary)**.** *For $\epsilon \geq 0$ an ϵ-random-probing adversary on $\mathcal{X}^\ell$ is an algorithm $\mathcal{A}$ that plays the following scenario against an oracle that knows $(x_1, \ldots, x_\ell) \in \mathcal{X}^\ell$:*

1. *$\mathcal{A}$ specifies a sequence $(\epsilon_1, \ldots, \epsilon_\ell)$ such that each $\epsilon_i \leq \epsilon$.*
2. *$\mathcal{A}$ receives $\varphi_1(x_1), \ldots, \varphi_\ell(x_\ell)$ and outputs some value denoted by $\mathsf{out}_{\mathcal{A}}(x_1, \ldots, x_\ell)$, where each φ_i is the ϵ_i-identity function with mutually independent randomness.*

Note that here, $\mathsf{out}_{\mathcal{A}}(x_1, \ldots, x_\ell)$ is a random variable combining $\mathcal{A}$ and the randomness of the functions φ_i.

A similar model was introduced in the work of Ishai, Sahai and Wagner [ISW03] to obtain a circuit compiler that blows-up the size of the circuit linearly in the security parameter d. Also, the work of Ajtai [Ajt11] considers the random probing model, and constructs a compiler that for sufficiently large security parameter d achieves security in the random probing model for a small (but constant) probability ϵ. Reference [Ajt11] however does not give concrete parameters for ϵ and d, and circuits produced by the compiler of [Ajt11] result into a huge circuit size blow-up ($O(d^4)$ with large hidden constants).

8.5 Noisy Model

We finally describe the noisy leakage model, a model in which the adversary sees a noisy version of each value of the circuit. The idea of noisy leakage was first introduced by Chari, Jutla, Rao, and Rohatgi in CRYPTO'99 [CJRR99]. In this seminal work, the authors consider a model, where each share b_i of an encoding is perturbed by Gaussian noise and show that the number of noisy samples needed to recover the encoded secret bit b grows exponential with the number of shares. As stated in [CJRR99], this model matches real-world physical leakages that inherently are noisy. Moreover, many practical solutions exist to amplify leakage noise (see for instance the works of [CK10, CCD00, MOP07]).

One limitation of the security analysis given in [CJRR99] is the fact that it does not consider leakage emitting from masked computation. This shortcoming has been addressed in the recent important work of Prouff and Rivain [PR13], who extend at Eurocrypt 2013 the noisy leakage model of Chari et al. [CJRR99] to also include leakage from the masked operations. Specifically, they show that a variant of the construction of Ishai et al. [ISW03] is secure even when there is noisy leakage from all the intermediate values that are produced during the computation. The authors of [PR13] also generalize the noisy leakage model of Chari et al. [CJRR99] to a wider range of leakage functions instead of considering only the Gaussian one. While clearly noisy leakage is closer to physical leakage occurring in real world, the security analysis of [PR13] has a number of shortcomings which puts strong limitations in which settings the masking countermeasure can be used and achieves the proved security statements. In particular, like earlier works on leakage resilient cryptography [DF12, FRR+10], the security analysis of Prouff

and Rivain relies on so-called *leak-free* gates. Moreover, security is shown in a restricted adversarial model that assumes that plaintexts are chosen uniformly during an attack and the adversary does not exploit joint information from the leakages and, e.g., the ciphertext. We discuss these shortcomings in more detail in Section 9.1.

8.5.1 Modeling the Noise

We can now formalize our noise model based on the work of Prouff and Rivain [PR13]. Let us start with a discussion defining what it means that a randomized function $\mathsf{L}_{\mathsf{Noisy}} : \mathcal{X} \to \mathcal{Y}$ is "noisy".[3] We will assume that $\mathcal{X}$ is finite and rather small: typical choices for $\mathcal{X}$ would be GF(2) (the "Boolean case"), or GF(2^8), if we want to deal with the AES circuit. The set $\mathcal{Y}$ corresponds to the set of all possible noise measurements and may be infinite, except when we require the "efficient simulation" (we discuss it further at the end of this section). In [PR13], the authors introduced the notion of a *bias*, which informally says that the statistical distance between the distribution of X and the conditional distribution $X|\mathsf{L}_{\mathsf{Noisy}}(X)$ is bounded by some parameter. We base our noise definition on their idea and we define it more formally as follows:

Definition 8.4 (δ-noisy function)**.** *Let $\mathsf{L}_{\mathsf{Noisy}} \colon \mathcal{X} \to \mathcal{Y}$ be a randomized function. We say that the function $\mathsf{L}_{\mathsf{Noisy}}$ is* δ-noisy *if*

$$\delta = \Delta(X; (X \mid \mathsf{L}_{\mathsf{Noisy}}(X))) \,, \tag{8.1}$$

where Δ denotes the statistical distance (see Section 2.2).

Of course for (8.1) to be well-defined, we need to specify the distribution of X. The idea to define noisy functions by comparing the distributions of X and "X conditioned on $\mathsf{L}_{\mathsf{Noisy}}(X)$" comes from [PR13], where it is argued that the most natural choice for X is a random variable distributed uniformly over $\mathcal{X}$. We also adopt this convention and assume that $X \leftarrow \mathcal{X}$. We would like to stress, however, that in our proofs we will apply $\mathsf{L}_{\mathsf{Noisy}}$ to inputs $\widehat{X}$ that are not necessarily uniform and in this case the value of $\Delta(\widehat{X}; (\widehat{X}|\mathsf{L}_{\mathsf{Noisy}}(\widehat{X}))$ may obviously be some non-trivial function of δ. Of course if $X \leftarrow \mathcal{X}$ and $X' \leftarrow \mathcal{X}$ then $\mathsf{L}_{\mathsf{Noisy}}(X')$ is distributed identically to $\mathsf{L}_{\mathsf{Noisy}}(X)$, and hence, by Lemma 2.4, (8.1) is equivalent to:

$$\delta = \Delta((\mathsf{L}_{\mathsf{Noisy}}(X); \mathsf{L}_{\mathsf{Noisy}}(X')) \mid X) \,, \tag{8.2}$$

where X and X' are uniform over $\mathcal{X}$. Note that at the beginning, this definition may be a bit counter-intuitive, as *smaller* δ means *more* noise: in particular we achieve "full noise" if $\delta = 0$, and "no noise" if $\delta \approx 1$. Let us compare this definition with the definition of [PR13]. In a nutshell: the definition of [PR13] is similar to ours, the only difference being that instead of the statistical distance Δ in [PR13] the authors use a distance based on the Euclidean norm. More precisely, they start with defining d_2 as:

[3] In the following, we will not write the random coins used to compute the $\mathsf{L}_{\mathsf{Noisy}}$ function to simplify the notation.

Definition 8.5 (Euclidean Norm)**.**

$$\mathrm{d}_2(X;Y) := \sqrt{\sum_{x\in\mathcal{X}} (\Pr[X=x] - \Pr[Y=y])^2}\ .$$

Using this notion, they define β as:

$$\beta(X \mid \mathsf{L}_{\mathsf{Noisy}}(X)) := \sum_{y\in\mathcal{Y}} \Pr[\mathsf{L}_{\mathsf{Noisy}}(X)=y] \cdot \mathrm{d}_2(X\ ;\ (X \mid \mathsf{L}_{\mathsf{Noisy}}(X)=y))\ ,$$

where X is uniform. In the terminology of [PR13], a function $\mathsf{L}_{\mathsf{Noisy}}$ is "δ-noisy" if $\delta = \beta(X|\mathsf{L}_{\mathsf{Noisy}}(X))$. Observe that the right hand side of our noise definition in (8.1) can be rewritten as:

$$\sum_{y\in\mathcal{Y}} \Pr[\mathsf{L}_{\mathsf{Noisy}}(X)=y] \cdot \Delta(X\ ;\ (X \mid \mathsf{L}_{\mathsf{Noisy}}(X)=y))\ .$$

Hence the only difference between their approach and ours is that we use Δ where they use the distance d_2. The authors do not explain why they choose this particular measure. We believe that our choice to use the standard definition of statistical distance Δ is more natural in this setting, as, unlike the "d_2" distance, it has been used in hundreds of cryptographic papers in the past. The popularity of the Δ distance comes from the fact that it corresponds to an intuitive concept of the "indistinguishability of distributions". It is well-known, and simple to verify, that $\Delta(X;Y) \leq \delta$ if and only if no adversary can distinguish between X and Y with advantage better than δ.[4] Hence, e.g., (8.2) can be interpreted as:

> δ is the maximum probability, over all adversaries $\mathcal{A}$, that $\mathcal{A}$ distinguishes between the noise from a uniform X that is *known to him*, and a uniform X' that is *unknown to him*.

It is unclear to us if a d_2 distance has a similar interpretation. We emphasize, however, that the choice whether to use Δ or β is not too important, as the following inequalities between these measures hold for every X and Y distributed over $\mathcal{X}$ (cf. [PR13]):

$$\frac{1}{2}\cdot \mathrm{d}_2(X;Y) \leq \Delta(X;Y) \leq \frac{\sqrt{|\mathcal{X}|}}{2}\cdot \mathrm{d}_2(X;Y)\ ,$$

and consequently

$$\frac{1}{2}\cdot \beta(X \mid \mathsf{L}_{\mathsf{Noisy}}(X)) \leq \Delta(X;(X \mid \mathsf{L}_{\mathsf{Noisy}}(X)) \leq \frac{\sqrt{|\mathcal{X}|}}{2}\cdot \beta(X \mid \mathsf{L}_{\mathsf{Noisy}}(X))\ . \tag{8.3}$$

Hence, we decide to stick to the "Δ distance" in this thesis. However, to allow for comparison between our work and the one of [PR13], we will at the end of Chapter 9

[4] This formally means that for every $\mathcal{A}$ we have $|\Pr[\mathcal{A}(X)=1] - \Pr[\mathcal{A}(Y)=1]| \leq \delta$.

present our results also in terms of the β measure. This translation will be straightforward, thanks to the inequalities in (8.3). In [PR13, Theorem 4], the result is stated in form of Shannon information theory. While such an information theoretic approach may be useful in certain settings [SMY09], we follow the more "traditional" approach and provide an efficient simulation argument. As discussed in the introduction, this also covers a setting where the adversary exploits joint information of the leakage and, e.g., the plaintext/ciphertext pairs. We emphasize, however, that our results can easily be expressed in the information theoretic language as we will show in Chapter 10.

The Issue of "Efficient Simulation"

To achieve the strong simulation-based security notion, we need an additional requirement on the leakage, namely, that the leakage can efficiently be "simulated", which typically requires that the noise function is efficiently computable. In fact, for our proofs to go through, we actually need something slightly stronger, namely that $\mathsf{L}_{\mathsf{Noisy}}$ *is efficiently decidable.*

Definition 8.6 (Efficiently decidable noise)**.** *A function* $\mathsf{L}_{\mathsf{Noisy}} : \mathcal{X} \to \mathcal{Y}$ *is* efficiently decidable *if*

1. *there exists a ppt algorithm that computes it and*
2. *the set* $\mathcal{Y}$ *is finite and for every* x *and* y *the value of* $\Pr[\mathsf{L}_{\mathsf{Noisy}}(x) = y]$ *is computable in polynomial time.*

While the second requirement may look like a strong assumption we note that in practice for most "natural" noise functions (like the Gaussian noise with a known parameter, measured with a very good, but finite, precision) it is easily satisfiable.
The results of [PR13] are stated without taking into consideration the issue of the "efficient simulation". Hence, if one wants to compare our results with [PR13] then one can simply drop the efficient decidability assumption on the noise. To keep our presentation concise and clean, also in this case the results will be presented in a form "for every adversary $\mathcal{A}$ there exists an (inefficient) simulator $\mathcal{S}$". Here the "inefficient simulator" can be an arbitrary algorithm, capable, e.g., of sampling elements from *any* probability distributions.

8.5.2 Adversarial Model

We can now formally define our *noisy model.*

Definition 8.7 (δ-noisy adversary)**.** *For* $\delta \geq 0$*, a* δ-noisy adversary on $\mathcal{X}^\ell$ *is an algorithm* $\mathcal{A}$ *that plays the following scenario against an oracle that knows* $(x_1, \ldots, x_\ell) \in \mathcal{X}^\ell$*:*

1. $\mathcal{A}$ *specifies a sequence* $\{\mathsf{L}_{\mathsf{Noisy}_i} : \mathcal{X} \to \mathcal{Y}\}_{i=1}^{\ell}$ *of noisy functions such that every* $\mathsf{L}_{\mathsf{Noisy}_i}$ *is* δ_i'*-noisy, for some* $\delta_i' \leq \delta$ *and mutually independent noises.*

2. *$\mathcal{A}$ receives $\mathsf{L}_{\mathsf{Noisy}_1}(x_1), \ldots, \mathsf{L}_{\mathsf{Noisy}_\ell}(x_\ell)$ and outputs some value that we denote by $\mathsf{out}_{\mathcal{A}}(x_1, \ldots, x_\ell)$.*

Like before, $\mathsf{out}_{\mathcal{A}}(x_1, \ldots, x_\ell)$ is a random variable combining $\mathcal{A}$ and the randomness of the functions $\mathsf{L}_{\mathsf{Noisy}_i}$. If $\mathcal{A}$ works in polynomial time and the noise functions specified by $\mathcal{A}$ are efficiently decidable, then we say that $\mathcal{A}$ is poly-time-noisy.

8.5.3 Other Models

The work of Faust et al. [FRR+10] also considers circuit compilers for noisy models. Specifically, they propose a construction with security in the binomial noise model, where each value on a wire is flipped independently with probability $p \in (0, 1/2)$. In contrast to the work of [PR13] and our work, the noise model is restricted to binomial noise, but the noise rate is significantly better (constant instead of linear noise). Similar to [PR13] the work of Faust et al. also uses leak-free components. Besides these works on masking schemes, several works consider noisy leakages for concrete cryptographic schemes [DP08, NS09a, KV09]. Typically, the noise model considered in these works is significantly stronger than the noise model that is considered for masking schemes. In particular, no strong assumption about the independence of the noise is made.

Chapter 9

Unifying Leakage Models

This chapter presents a joint work with Prof. S. Dziembowski and Prof. S. Faust and was published in [DDF14]. While there is still a large gap between what theoretical models can achieve and what side-channel information is measured in practice, some recent important works introduce models that are better in line with the perspective of cryptographic engineering [SMY09, PR13, SPY13]. Our work follows this line of research by analyzing the security of a common countermeasure – the so-called masking countermeasure – in the model of Prouff and Rivain [PR13]. Our analysis works by showing that security in certain theoretical leakage models implies security in the model of [PR13], and hence may be seen as a first attempt to *unify* the large class of different leakage models used in recent results.
We briefly described Prouff and Rivain's work [PR13] in the previous chapter and pointed out some shortcomings. We discuss them more in the next section.

9.1 The Work of Prouff and Rivain [PR13]

Prouff and Rivain [PR11] analyze the security of a block-cipher implementation that is masked with an additive masking scheme working over a finite field $\mathbb{F}$. More precisely, let t be the security parameter, then a secret $s \in \mathbb{F}$ is represented by an encoding $(X_1, \ldots, X_t)$ such that each $X_i \leftarrow \mathbb{F}$ is uniformly random subject to $s = X_1 \oplus \ldots \oplus X_t$. As discussed above, the main difficulty in designing secure masking schemes is to devise masked operations that work on masked values. To this end, Prouff and Rivain use the original scheme of Ishai et al. [ISW03] augmented with some techniques from [CGP$^+$12a, RP10] to work over larger fields and to obtain a more efficient implementation. The masked operations are built out of several smaller components. First, a leak-free operation that refreshes encodings, i.e., it takes as input an encoding $(X_1, \ldots, X_t)$ of a secret s and outputs a freshly and independently chosen encoding of the same value. Second, a number of leaky elementary operations that work on a constant number of field elements. For each of these elementary operations the adversary is given leakage $f(X)$, where X are the inputs of the operation and f is a noisy function.

Clearly, the noise-level has to be high enough so that given $f(X)$ the values of X is not completely revealed. To this end, the authors introduce the noisy leakage model presented in Section 8.5.
While noisy leakages are certainly a step in the right direction to model physical leakage, we detail below some of the limitations of the security analysis of Prouff and Rivain [PR13]:

1. *Leak-free components:* The assumption of leak-free computation has been used in earlier works on leakage resilient computation [FRR+10, DF12]. It is a strong assumption on the physical hardware and, as stated in [PR13], an important limitation of the current proof approach. The leak-free component of [PR13] is a simple operation that takes as input an encoding and refreshes it. While the computation of this operation is supposed to be completely shielded against leakage, the inputs and the outputs of this computation may leak. Notice that the leak-free component of [PR13] depends on the computation that is carried out in the circuit by taking inputs. In particular, this means that the computation of the leak-free component depends on secret information, which makes it harder to protect in practice and is different from earlier works that use leak-free components [FRR+10, DF12].

2. *Random message attacks:* The security analysis is given only for random message attacks. In particular, it is assumed that every masked secret is a uniformly random value. This is in contrast to most works in cryptography, which usually consider at least a chosen message attack. When applied to a block-cipher, their proof implies that the adversary has only access to the *leakage* of the system without knowing which plaintext was used nor which ciphertext was obtained. Hence, the proof does not cover chosen plaintext or chosen ciphertext attacks. However, it is true that it is not clear how chosen message attacks change the picture in standard DPA attacks [VCS10].

3. *Mutual-information-based security statement:* The final statement of Theorem 4 in [PR13] only gives a bound on the mutual information of the key and the leakages from the cipher. In particular, this does not include information that an adversary may learn from exploiting joint information from the leakages and plaintext/ciphertext pairs. Notice that the use of mutual information gets particularly problematic under continuous leakage attacks, as multiple plaintext/ciphertext pairs information theoretically completely reveal the secret key. The standard security notion used, e.g., in Ishai et al. is simulation-based and covers such subtleties when dealing with Shannon information theory.

4. *Strong noise requirements:* The amount of noise that is needed depends on the number of shares and on the size of the field which might be a bit unnatural. Moreover, the noise is independently sampled for each of the elementary operation that have constant size.

9.2 Our Contribution

We show in this work how to eliminate limitations 1-3 by a simple and elegant simulation-based argument and a reduction to the so-called t-probing adversarial setting [ISW03] (that in this thesis we call the *t-threshold-probing model* to emphasize the difference between this model and the *random*-probing model defined later.). The t-threshold-probing model considers an adversary that can learn the value of t intermediate values that are produced during the computation and is often considered as a good approximation for modelling higher-order attacks. We notice that limitation 4 from above is what enables our security analysis. The fact that the noise is independent for each elementary operation allows us to formally prove security under an *identical* noise model as [PR13], but using a simpler and improved analysis. In particular, we are able to show that the original construction of Ishai et al. satisfies the standard simulation-based security notion under noisy leakages without relying on any leak-free components. We emphasize that our techniques are very different (and much simpler) than the recent breakthrough result of Goldwasser and Rothblum [GR12], which shows how to eliminate leak-free gates in the bounded leakage model. We will further discuss related works in Section 9.3.
Our proof considers three different leakage models and shows connections between them. One may view our work as a first attempt to "reduce" the number of different leakage models, which is in contrast to many earlier works that introduced new leakage settings. Eventually, we are able to reduce the security in the noisy leakage model to the security in the t -threshold-probing model. This shows that, for the particular choice of parameters given in [PR13], security in the t-threshold-probing model implies security in the noisy leakage model. This goes align with the common approach of showing security against t-order attacks, which usually requires to prove security in the t–threshold-probing model. Moreover, it shows that the original construction of Ishai et al. that has been used in many works on masking (including the work of Prouff and Rivain) is indeed a sound approach for protecting against side-channel leakages when assuming that they are sufficiently noisy. We give some more details on our techniques below.

From noisy leakages to random probes. As a first step in our security proof, we show that we can simulate any adversary in the noisy leakage model of Prouff and Rivain with a *random probing adversary* that we defined in Section 8.4.

From random probes to the t-threshold-probing model. We show how to go from the random probing adversary setting to the more standard t-threshold-probing adversary of Ishai et al. in [ISW03] (see Section 8.3). This step is rather easy as due to the independence of the noise we can apply Chernoff's bound almost immediately. One technical difficulty is that the work of Prouff and Rivain considers joint noisy leakage from elementary operations, while the standard t-threshold-probing setting only talks about leakage from wires. Notice, however, that the elementary operations of [PR13] only depend on two inputs and, hence, it is not hard to extend the result of Ishai et

al. to consider "gate probing adversary" by tolerating a loss in the parameters. Finally, our analysis enables us to show security of the masking based countermeasure without the limitations 1-3 discussed above.

Leakage resilient circuits with simulation-based security. In our security analysis we use the framework of leakage resilient circuits introduced in the seminal work of Ishai et al. [ISW03].
Moreover, our work uses the well-established simulation paradigm to state the security guarantees we achieve. Intuitively, simulation-based security says that whatever attack an adversary can carry out when knowing the leakage, he can also run (with similar success probability) by just having black-box access to C. In contrast to the approach based on Shannon information theory, our analysis includes attacks that exploit joint information from the leakage and plaintext/ciphertext pairs. It seems impossible to us to incorporate the plaintext/ciphertext pairs into an analysis based on Shannon information theory. To see this, consider a block-cipher execution, where, clearly, when given a couple of plaintext/ciphertext pairs, the secret key is information theoretically revealed.[1] The authors of [PR13] are well aware of this problem and explicitly exclude such joint information. A consequence of the simulation-based security analysis is that we require an additional mild assumption on the noise – namely, that it is efficiently computable (see Section 9.4 for more details). While this is a standard assumption made in most works on leakage resilient cryptography, we emphasize that we can easily drop the assumption of efficiently computable noise (and hence considering the same noise model as [PR13]), when we only want to achieve the weaker security notion considered in [PR13]. Notice that in this case, we are still able to eliminate the limitations 1 & 2 mentioned above.

9.3 Related Work

Masking & leakage resilient circuits. A large body of work has proposed various masking schemes and studies their security in different security models (see, e.g., [GP99, BGK04, OMPR05, SVCO+10, RP10]). The already mentioned t-threshold-probing model has been considered in the work of Rivain and Prouff [RP10], who show how to extend the work of Ishai et al. to larger fields and propose efficiency improvements. In [PR11] it was shown that techniques from multiparty computation can be used to show security in the t-threshold-probing model. The work of Standaert et al. [SVCO+10] studies masking schemes using the information theoretic framework of [SMY09] by considering the Hamming weight model. Many other works analyze the security of the masking countermeasure and we refer the reader for further details to [PR13].

[1] More concretely: imagine an adversary that attacks a block-cipher implementation E_K, where K is the secret key. Then just by launching a known-plaintext attack he can obtain several pairs $V = (M_0, E_K(M_0)), (M_1, E_K(M_1)), \ldots$. Clearly a small number of such pairs is usually enough to determine K *information-theoretically*. Hence it makes no sense to require that "K is information-theoretically hidden given V and the side-channel leakage."

With the emerge of leakage resilient cryptography [MR04, AGV09, DP08], several works have proposed new security models and alternative masking schemes. The main difference between these new security models and the t-threshold-probing model is that they consider *joint leakages* from large parts of the computation. The work of Faust et al. [FRR+10] extends the security analysis of Ishai et al. beyond the t-threshold-probing model by considering leakages that can be described by low-depth circuits (so-called AC^0 leakages). Faust et al. use leak-free component that have been eliminated by Rothblum in [Rot12] using computational assumptions. The recent work of Miles and Viola [MV13] presents a new circuit transformation using alternating groups and shows security with respect to AC^0 and TC^0 leakages.
Another line of work considers circuits that are provably secure in the so-called continuous bounded leakage model [JV10, GR10, DF12, GR12]. In this model, the adversary is allowed to learn arbitrary information from the computation of the circuit as long as the amount of information is bounded. The proposed schemes rely additionally on the assumption of "only computation leaks information" of Micali and Reyzin [MR04].

9.4 Noise from Set Elements

We start with describing the basic framework for reasoning about the noise from elements of a finite set $\mathcal{X}$. Later, in Section 9.5, we will consider the noisy leakage from the vectors over $\mathcal{X}$, and then, in Section 9.6, from the entire computation. The reason why we can smoothly use the analysis from Section 8.5.1 in the later sections is that, as in the work of Prouff and Rivain, we require that the noise is independent for all elementary operations. By elementary operations, [PR13] considers the basic underlying operations over the underlying field $\mathcal{X}$ used in a masked implementation. In this work, we consider the same setting and type of underlying operations (in fact, notice that our construction is identical to theirs – except that we eliminate the leak-free gates and prove a stronger statement). Notice that instead of talking about elementary operations, we consider the more standard term of "gates" that was used in the work of Ishai et al. [ISW03].

9.4.1 Simulating Noise by ϵ-identity Functions

Lemma 9.1 below is our main technical tool. Informally, it states that every δ-noisy function $\mathsf{L}_{\mathsf{Noisy}} : \mathcal{X} \to \mathcal{Y}$ can be represented as a composition $\mathsf{L}_{\mathsf{Noisy}}{}' \circ \varphi$ of efficiently computable randomized functions $\mathsf{L}_{\mathsf{Noisy}}{}'$ and φ, where φ is a "$\delta \cdot |\mathcal{X}|$-identity function", defined in Definition 8.2.
This will allow us to reduce the "noisy attacks" to the "random probing attacks", where the adversary learns each wire (or a gate, see Section 9.6.5) of the circuit with probability ϵ. Observe also, that thanks to the assumed independence of noise, the events that the adversary learns each element are independent, which, in turn, will allow us to use the Chernoff bound to prove that with a good probability the number of wires that the adversary learns is small.

Lemma 9.1. *Let $\mathsf{L}_{\mathsf{Noisy}} : \mathcal{X} \to \mathcal{Y}$ be a δ-noisy function. Then there exist $\epsilon \leq \delta \cdot |\mathcal{X}|$ and a randomized function $\mathsf{L}_{\mathsf{Noisy}}' : \mathcal{X} \cup \{\perp\} \to \mathcal{X}$ such that for every $x \in \mathcal{X}$ we have*

$$\mathsf{L}_{\mathsf{Noisy}}(x) \stackrel{d}{=} \mathsf{L}_{\mathsf{Noisy}}'(\varphi(x)), \tag{9.1}$$

where $\varphi : \mathcal{X} \to \mathcal{X} \cup \{\perp\}$ is an ϵ-identity function. Moreover, if $\mathsf{L}_{\mathsf{Noisy}}$ is efficiently decidable (see Definition 8.6) then $\mathsf{L}_{\mathsf{Noisy}}'(\varphi(x))$ is computable in time that is expected polynomial in $|\mathcal{X}|$.

Proof. We consider only the case when $\mathsf{L}_{\mathsf{Noisy}}$ is efficiently decidable, and hence the $\mathsf{L}_{\mathsf{Noisy}}'$ function that we construct will be efficiently computable. The case when $\mathsf{L}_{\mathsf{Noisy}}$ is not efficiently decidable is handled in an analogous way (the proof is actually simpler as the only difference is that we do not need to argue about the efficiency of the sampling algorithms). Let X and X' be uniform over $\mathcal{X}$. For every $y \in \mathcal{Y}$ define

$$\pi(y) = \min_{x \in \mathcal{X}}(\Pr[\mathsf{L}_{\mathsf{Noisy}}(x) = y]). \tag{9.2}$$

Clearly π is computable in time polynomial in $|\mathcal{X}|$. Obviously, π is usually not a probability distribution as it does not sum up to 1. The good news is that it sums up "almost" to 1 provided δ is sufficiently small. This is shown below. Let $\epsilon := 1 - \sum_{y \in \mathcal{Y}} \pi(y)$. We now have

$$\begin{aligned}
\epsilon &= \overbrace{\sum_{y \in \mathcal{Y}} \Pr[\mathsf{L}_{\mathsf{Noisy}}(X') = y]}^{=1} - \sum_{y \in \mathcal{Y}} \pi(y) \\
&= \sum_{y \in \mathcal{Y}} \Pr[\mathsf{L}_{\mathsf{Noisy}}(X') = y] - \min_{x \in \mathcal{X}}(\Pr[\mathsf{L}_{\mathsf{Noisy}}(x) = y]) \\
&= \sum_{y \in \mathcal{Y}} \max_{x \in \mathcal{X}}(\Pr[\mathsf{L}_{\mathsf{Noisy}}(X') = y] - \Pr[\mathsf{L}_{\mathsf{Noisy}}(x) = y]) \\
&\leq \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} \max(0, \Pr[\mathsf{L}_{\mathsf{Noisy}}(X') = y] - \Pr[\mathsf{L}_{\mathsf{Noisy}}(x) = y])) \qquad (9.3) \\
&= \sum_{x \in \mathcal{X}} \Delta(\mathsf{L}_{\mathsf{Noisy}}(x); \mathsf{L}_{\mathsf{Noisy}}(X')) \\
&= |\mathcal{X}| \cdot \Delta((\mathsf{L}_{\mathsf{Noisy}}(X); \mathsf{L}_{\mathsf{Noisy}}(X')) \mid X) \\
&= \delta \cdot |\mathcal{X}|, \qquad (9.4)
\end{aligned}$$

where (9.3) comes from the fact that the maximum of positive values cannot be larger than their sum[2], and (9.4) follows from the assumption that the $\mathsf{L}_{\mathsf{Noisy}}$ function is δ-noisy.

[2]More precisely, for every $\{Z_x\}_{x \in \mathcal{X}}$ we have:

$$\begin{aligned}
\max_{x \in \mathcal{X}}(Z_x) &\leq \sum_{x : Z_x \geq 0} Z_x \\
&= \sum_x \max(0, Z_x),
\end{aligned}$$

Now, let $\mathsf{L_{Noisy}}'(x)$ be a distribution defined as follows: for every $y \in \mathcal{Y}$ and every $x \neq \perp$ let:

$$\Pr[\mathsf{L_{Noisy}}'(x) = y] = (\Pr[\mathsf{L_{Noisy}}(x) = y] - \pi(y))/\epsilon, \tag{9.5}$$

and otherwise:

$$\Pr[\mathsf{L_{Noisy}}'(\perp) = y] = \pi(y)/(1-\epsilon). \tag{9.6}$$

We will later show how to sample $\mathsf{L_{Noisy}}'$ efficiently. Obviously, this will automatically imply that (9.5) and (9.6) define probability distributions over $\mathcal{Y}$ (which may not be obvious at the first sight). First, however, let us show (9.1). To this end take any $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ and observe that

$$\begin{aligned}
&\Pr[\mathsf{L_{Noisy}}'(\varphi(x)) = y] \\
&\quad = \Pr[\varphi(x) = x] \cdot \Pr[\mathsf{L_{Noisy}}'(x) = y] + \Pr[\varphi(x) = \perp] \cdot \Pr[\mathsf{L_{Noisy}}'(\perp) = y] \\
&\quad = \epsilon \cdot (\Pr[\mathsf{L_{Noisy}}(x) = y] - \pi(y))/\epsilon + (1-\epsilon) \cdot \pi(y)/(1-\epsilon) \\
&\quad = \Pr[\mathsf{L_{Noisy}}(x) = y] - \pi(y) + \pi(y) \\
&\quad = \Pr[\mathsf{L_{Noisy}}(x) = y].
\end{aligned}$$

Which implies (9.1). What remains is to show how to sample $\mathsf{L_{Noisy}}'$ efficiently. Let us first show an efficient algorithm $\mathsf{Alg_1}(x)$ for computing $\mathsf{L_{Noisy}}'(x)$ for $x \neq \perp$:

1: **algorithm** $\mathsf{Alg_1}(x)$
2: Sample y from $\mathsf{L_{Noisy}}(x)$.
3: With probability $\pi(y)/\Pr[\mathsf{L_{Noisy}}(x) = y]$ resample y, i.e.: go back to Step 2.
4: Output y.
5: **end algorithm**

We now argue that $\mathsf{Alg_1}(x)$ indeed computes $\mathsf{L_{Noisy}}'(x)$ efficiently. Let $R_1 \in \{1, 2, \ldots\}$ be a random variable denoting the number of times the algorithm $\mathsf{Alg_1}(x)$ performed Step 2. First observe that the probability of jumping back to Step 2 in Step 3 is equal to

$$\sum_y \Pr[\mathsf{L_{Noisy}}(x) = y] \cdot \pi(y)/\Pr[\mathsf{L_{Noisy}}(x) = y] = \sum_y \pi(y) \tag{9.7}$$

$$= 1 - \epsilon \tag{9.8}$$

Therefore the probability of *not* jumping back to Step 2 in Step 3 is ϵ, and hence the expected number $\mathbb{E}[R_1]$ of the executions of Step 2 in $\mathsf{Alg_1}(x)$ is equal to $\sum_{i=1}^{n} i \cdot (1 -$

where in our case $Z_x := \Pr[\mathsf{L_{Noisy}}(x) = y]$.

$\epsilon)^{i-1} \cdot \epsilon = 1/\epsilon$. Moreover for every $i = 0, 1, \ldots$ we have:

$$\begin{aligned}
&\Pr[\mathsf{Alg}_1(x) = y \wedge R_1 = i \mid R_1 \geq i] \\
&\quad = \Pr[\mathsf{L}_{\mathsf{Noisy}}(x) = y] \cdot (1 - (\pi(y)/\Pr[\mathsf{L}_{\mathsf{Noisy}}(x) = y])) \\
&\quad = \Pr[\mathsf{L}_{\mathsf{Noisy}}(x) = y] - \pi(y)
\end{aligned}$$

Hence

$$\begin{aligned}
&\Pr[\mathsf{Alg}_1(x) = y] \\
&\quad = \sum_{i=0}^{\infty} \Pr[\mathsf{Alg}_1(x) = y \wedge R_1 = i] \\
&\quad = \sum_{i=0}^{\infty} \Pr[\mathsf{Alg}_1(x) = y \wedge R_1 = i \mid R_1 \geq i] \cdot \Pr[R_1 \geq i] \\
&\quad = (\Pr[\mathsf{L}_{\mathsf{Noisy}}(x) = y] - \pi(y)) \cdot \sum_{i=1}^{\infty} \Pr[R_1 \geq i] \\
&\quad = (\Pr[\mathsf{L}_{\mathsf{Noisy}}(x) = y] - \pi(y)) \cdot \mathbb{E}[R_1] \\
&\quad = (\Pr[\mathsf{L}_{\mathsf{Noisy}}(x) = y] - \pi(y))/\epsilon,
\end{aligned}$$

as required in (9.5). We now present an efficient algorithm Alg_2 for computing $\mathsf{L}_{\mathsf{Noisy}}{}'(\perp)$. Fix an arbitrary element $x_0 \in \mathcal{X}$, and execute the following.

1: **algorithm** $\mathsf{Alg}_2()$
2: Sample y from $\mathsf{L}_{\mathsf{Noisy}}(x_0)$.
3: With probability $1-(\pi(y)/\Pr[\mathsf{L}_{\mathsf{Noisy}}(x_0) = y])$ resample y, i.e., go back to Step 2.
4: Output y.
5: **end algorithm**

By a similar argument as for Alg_1 we obtain that the expected number R_2 of times the algorithm Alg_2 performs Step 2 is equal to $\mathbb{E}[R_2] = 1/(1-\epsilon)$. Moreover for every $i = 1, 2, \ldots$ we have:

$$\Pr[\mathsf{Alg}_2 = b \wedge R_2 = i \mid R_2 \geq i] = \pi(y),$$

which, in turn, implies that $\Pr[\mathsf{Alg}_2(x) = y] = \pi(y)/(1-\epsilon)$, and hence the output of Alg_2 satisfies (9.6). Clearly, the expected running time of both algorithms is polynomial in $|\mathcal{X}|$ and $\mathbb{E}[R]$, where R is the number of execution of Step 2 in Alg_1 or Alg_2. We obviously have

$$\begin{aligned}
\mathbb{E}[R] &= \mathbb{E}[R_1 \mid \varphi(x) \neq \perp] \cdot \Pr[\varphi(x) \neq \perp] + \mathbb{E}[R_2 \mid \varphi(x) = \perp] \cdot \Pr[\varphi(x) = \perp] \\
&= (1/\epsilon) \cdot \epsilon + (1/(1-\epsilon)) \cdot (1-\epsilon) \\
&= 2.
\end{aligned}$$

Hence, the expected running time of $\mathsf{L}_{\mathsf{Noisy}}{}'(\varphi(x))$ is polynomial in $|X|$. □

Informally speaking, it is based on an extension of the standard observation that for any two random variables A and B one can find two events $\mathcal{A}$ and $\mathcal{B}$ such that the distributions $P_{A|\mathcal{A}}$ and $P_{B|\mathcal{B}}$ are equal and $\Pr[\mathcal{A}], \Pr[\mathcal{B}] = \Delta(A; B)$ (see, e.g., [MT10, Section 1.3]).

9.5 Leakage from Vectors

In this section, we adapt our result from the previous section to leakage applied to vector of values. It is in this section that we show how to simulate a noisy adversary with a threshold-probing adversary (using for this as an intermediate step a random-probing adversary). The different adversarial models are presented in Section 8.2.

9.5.1 Simulating the Noisy Adversary by a Random-probing Adversary

The following lemma shows that every δ-noisy adversary can be simulated by a $\delta \cdot |\mathcal{X}|$-random probing adversary.

Lemma 9.2. *Let $\mathcal{A}$ be a δ-noisy adversary on $\mathcal{X}^\ell$. Then there exists a $\delta \cdot |\mathcal{X}|$-random-probing adversary $\mathcal{S}$ on $\mathcal{X}^\ell$ such that for every $(x_1, \ldots, x_\ell)$ we have*

$$\mathit{out}_{\mathcal{A}}(x_1, \ldots, x_\ell) \stackrel{d}{=} \mathit{out}_{\mathcal{S}}(x_1, \ldots, x_\ell). \tag{9.9}$$

Moreover, if $\mathcal{A}$ is poly-time-noisy, then $\mathcal{S}$ works in time polynomial in $|\mathcal{X}|$.

Proof. Without loss of generality assume that $\mathcal{A}$ simply outputs all the information that he gets. Thus (9.9) can be rewritten as:

$$(\mathsf{L}_{\mathsf{Noisy}_1}(x_1), \ldots, \mathsf{L}_{\mathsf{Noisy}_\ell}(x_\ell)) \stackrel{d}{=} \mathsf{out}_{\mathcal{S}}(x_1, \ldots, x_\ell), \tag{9.10}$$

where $\mathsf{L}_{\mathsf{Noisy}_i}$'s are the δ_i-noisy functions chosen by $\mathcal{A}$. By Lemma 9.1 for each i there exists $\epsilon_i \leq \delta_i \cdot |\mathcal{X}| \leq \delta \cdot |\mathcal{X}|$ and a randomized function $\mathsf{L}'_{\mathsf{Noisy}_i} : \mathcal{X} \cup \{\perp\} \to \mathcal{X}$, such that for every $x \in \mathcal{X}$ we have

$$\mathsf{L}_{\mathsf{Noisy}_i}(x) \stackrel{d}{=} \mathsf{L}'_{\mathsf{Noisy}_i}(\varphi_i(x)), \tag{9.11}$$

where $\varphi_i : \mathcal{X}_i \to \mathcal{X}_i \cup \{\perp\}$ is the ϵ_i-identity function and $Noise'_i(\varphi_i(x))$ is computable in time polynomial in $|\mathcal{X}|$. We now describe the actions of $\mathcal{S}$. The sequence that he specifies is $(\epsilon_1, \ldots, \epsilon_\ell)$. After receiving $(y_1, \ldots, y_\ell)$ (equal to $(\varphi_1(x_1), \ldots, \varphi_\ell(x_\ell))$) he outputs

$$\mathsf{out}(x_1, \ldots, x_\ell) := (\mathsf{L}'_{\mathsf{Noisy}_1}(y_1), \ldots, \mathsf{L}'_{\mathsf{Noisy}_\ell}(y_\ell))$$

(this clearly takes time that is expected polynomial in $\ell \cdot |\mathcal{X}|$). We now have

$$\begin{aligned}(\mathsf{L}_{\mathsf{Noisy}_1}'(y_1), \ldots, \mathsf{L}_{\mathsf{Noisy}_\ell}'(y_\ell)) &\stackrel{d}{=} (\mathsf{L}_{\mathsf{Noisy}_1}'(\varphi_1(x_1)), \ldots, \mathsf{L}_{\mathsf{Noisy}_\ell}'(\varphi_\ell(x_\ell))) \\ &\stackrel{d}{=} (\mathsf{L}_{\mathsf{Noisy}_1}(x_1), \ldots, \mathsf{L}_{\mathsf{Noisy}_\ell}(x_\ell)) \qquad (9.12)\end{aligned}$$

where (9.12) comes from (9.11). This implies (9.10) and hence it finishes the proof. □

Intuitively, this lemma easily follows from Lemma 9.1 applied independently to each element of $(x_1, \ldots, x_\ell)$.

9.5.2 Simulating the Random-probing Adversary by a Threshold-probing Adversary

In this section, we show how to simulate every δ-random probing adversary by a threshold adversary. This simulation, unlike the one in Section 9.5 will not be perfect in the sense that the distribution output by the simulator will be identical to the distribution of the original adversary only when conditioned on some event that happens with a large probability. We start with the following lemma, whose proof is a straightforward application of the Chernoff bound.

Lemma 9.3. *Let $\mathcal{A}$ be an ϵ-random-probing adversary on $\mathcal{X}^\ell$. Then there exists a $(2\epsilon\ell - 1)$-threshold-probing adversary $\mathcal{S}$ on $\mathcal{X}^\ell$ operating in time linear in the working time of $\mathcal{A}$ such that for every $(x_1, \ldots, x_\ell)$ we have*

$$\Delta(\mathsf{out}_{\mathcal{A}}(x_1, \ldots, x_\ell) \; ; \; \mathsf{out}_{\mathcal{S}}(x_1, \ldots, x_\ell) \mid \mathsf{out}_{\mathcal{S}}(x_1, \ldots, x_\ell) \neq \perp) = 0, \qquad (9.13)$$

where

$$\Pr[\mathsf{out}_{\mathcal{S}}(x_1, \ldots, x_\ell) = \perp] \leq \exp\left(-\frac{\epsilon\ell}{3}\right). \qquad (9.14)$$

Proof. As in the proof of Lemma 9.2, we assume that the simulated adversary $\mathcal{A}$ outputs all the information that he received. Moreover, since for $\epsilon' \leq \epsilon$ every ϵ'-identity function φ' can be simulated by the ϵ-identity function φ,[3] hence we can assume that each ϵ_i specified by $\mathcal{A}$ is equal to ϵ. Thus, we need to show a $2\epsilon\ell$-threshold-probing simulator $\mathcal{S}$ such that for every $(x_1, \ldots, x_\ell) \in \mathcal{X}^\ell$ we have

$$\Delta(\varphi_1(x_1), \ldots, \varphi_\ell(x_\ell) \; ; \; \mathsf{out}_{\mathcal{S}}(x_1, \ldots, x_\ell) \mid \mathsf{out}_{\mathcal{S}}(x_1, \ldots, x_\ell) \neq \perp) = 0\,, \qquad (9.15)$$

(where each φ_i is the ϵ-identity function) and (9.14) holds. The simulator $\mathcal{S}$ proceeds as follows. First a sequence $(Z_1, \ldots, Z_\ell)$ of independent random variables is chosen by

[3] Just set $\varphi'(x) := \varphi(x)$ with probability ϵ'/ϵ, and $\varphi'(x) = \perp$ otherwise. Then clearly $\Pr[\varphi'(x) = x] = \epsilon \cdot \epsilon'/\epsilon = \epsilon'$.

setting, for each i,

$$Z_i := \begin{cases} 1 & \text{with probability } \epsilon_i \\ 0 & \text{otherwise.} \end{cases}$$

Let Z denote the number of Z_i's equal to 1, i.e, $Z := \sum_{i=1}^{\ell} Z_i$. If $Z \geq 2\ell\epsilon$ then $\mathcal{S}$ outputs $\perp$. Otherwise, it specifies the set $\mathcal{I}$ as $\mathcal{I} := \{i : Z_i = 1\}$. The simulator receives $(x_{i_1}, \ldots, x_{i_{|\mathcal{I}|}})$. For all the remaining i's (i.e. those not in the set $\mathcal{I}$) the simulator sets $x_i := \perp$. It outputs $(x_1, \ldots, x_\ell)$. It is straightforward to see that $\mathcal{S}$ is $(2\epsilon\ell - 1)$-threshold-probing and that (9.15) holds. What remains is to show (9.14). As $\mathbb{E}[Z] = \epsilon\ell$,

$$\begin{aligned} \Pr[Z \geq 2\ell\epsilon] &= \Pr[Z \geq 2\mathbb{E}[Z]] \\ &\leq \exp\left(-\frac{\epsilon\ell}{3}\right), \end{aligned} \tag{9.16}$$

where (9.16) comes from the Chernoff bound with $\xi = 1$ (cf. Lemma 2.14). This finishes the proof. □

The following corollary combines Lemma 9.2 and 9.3 together and will be useful in the sequel.

Corollary 9.4. *Let $d, \ell \in \mathbb{N}$ with $\ell > d$ and let $\mathcal{A}$ be a $d/(4\ell \cdot |\mathcal{X}|)$-noisy adversary on $\mathcal{X}^\ell$. Then there exists an $(d/2 - 1)$-threshold-probing adversary $\mathcal{S}$ such that*

$$\Delta(\mathsf{out}_{\mathcal{A}}(x_1, \ldots, x_\ell) \ ; \ \mathsf{out}_{\mathcal{S}}(x_1, \ldots, x_\ell) \mid \mathsf{out}_{\mathcal{S}}(x_1, \ldots, x_\ell) \neq \perp) = 0 \tag{9.17}$$

and $\Pr[\mathsf{out}_{\mathcal{S}}(x_1, \ldots, x_\ell) = \perp] \leq \exp(-d/12)$. Moreover, if $\mathcal{A}$ is poly-time-noisy then $\mathcal{S}$ works in time polynomial $\ell \cdot |\mathcal{X}|$,

Proof. By Lemma 9.2, there exists a $d/(4\ell)$-random-probing adversary $\mathcal{A}'$ whose output is distributed identically to the output of $\mathcal{A}$. In turn, by Lemma 9.3 for $t = 2 \cdot (d/(4\ell)) \cdot \ell = d/2$ there exists a $(t-1)$-threshold-probing adversary $\mathcal{S}$ whose output, conditioned on not being equal to $\perp$, is distributed identically to the output of $\mathcal{A}'$, and such that $\Pr[\mathsf{out}_{\mathcal{S}}(x_1, \ldots, x_\ell) = \perp] \leq \exp(-d/12)$.
If $\mathcal{A}$ is poly-time noisy then clearly the expected working time of $\mathcal{A}'$ is polynomial in $\ell \cdot |\mathcal{X}|$. As the working time of $\mathcal{S}$ is linear in the working time of $\mathcal{A}$ hence this finishes the proof. □

9.6 Leakage from Computation

In this section, we address the main topic of this chapter, which is the noise-resilience of cryptographic computations. Our main model will be the model of arithmetic circuits over a finite field. First, in Section 9.6.1, we formally define arithmetic circuits as well as our security definitions, and then, in Section 9.6.2, we describe a secure "compiler" that

transforms any cryptographic scheme secure in the "black-box" model into one secure against the noisy leakage. It is essentially identical to the transformation of [ISW03] later extended in [RP10]. Finally, in the last section, we present our security results.

9.6.1 Arithmetic Circuits

We will model our computations using (stateful) arithmetic circuits that we formally define below.

Definition 9.5 (Stateful arithmetic circuit)**.** *A* (stateful arithmetic) circuit Γ over a field $\mathbb{F}$ *is a directed graph whose nodes are called* gates.
Gates: *each gate γ can be of one of the following types:*

- *an* input gate γ^{inp} *of fan-in zero and fan-out* 1,
- *an* output gate γ^{out} *of fan-in* 1 *and fan-out zero,*
- *a* random gate γ^{rand} *of fan-in zero and fan-out* 1,
- *a* multiplication gate $\gamma^{\times}$ *of fan-in* 2 *and fan-out* 1,
- *an* addition gate γ^{+} *of fan-in* 2 *and fan-out* 1,
- *a* subtraction gate γ^{-} *of fan-in* 2 *and fan-out* 1,
- *a* constant gate $\gamma_c^{\mathsf{const}}$ *for a value $c \in \mathbb{F}$ of fan-in zero and fan-out* 1,
- *and a* memory gate γ^{mem} *of fan-in* 1 *and fan out* 1.

The only cycles that are allowed in Γ must contain exactly 1 *memory gate. The* size $|\Gamma|$ of the circuit Γ *is defined to be the total number of its gates. The numbers of input gates, output gates and memory gates will be denoted $|\Gamma.\mathsf{inp}|, |\Gamma.\mathsf{out}|$, and $|\Gamma.\mathsf{mem}|$, respectively.*

Computations: *the computation of Γ is performed in several "rounds" numbered $1, 2, \ldots$. In each of them the circuit will take some input, produce an output and update the memory state. Initially, the memory gates of Γ are preloaded with some initial "state" $k_0 \in \mathbb{F}^{|\Gamma.\mathsf{mem}|}$. At the beginning of the ith round the input gates are loaded with elements of some vector $a_i \in \mathbb{F}^{|\Gamma.\mathsf{inp}|}$ called the* input for the ith round. *The computation of Γ in the ith round depends on a_i and on the memory state k_{i-1}. It proceeds in a straightforward way: if all the input wires of a given gate γ are known then the value on its output wire can be computed naturally (where all the operations are done over $\mathbb{F}$):*

- *if γ is a multiplication gate $\gamma^{\times}$ with input wires carrying values a and b, then its output wire will carry the value $a \cdot b$.*
- *If γ is an addition gate γ^{+} with input wires carrying values a and b, then its output wire will carry the value $a + b$.*

- *If γ is an subtraction gate γ^{-} with input wires carrying values a and b, then its output wire will carry the value $a - b$.*
- *If γ is a constant gate $\gamma_c^{\mathsf{const}}$ with value $c \in \mathbb{F}$ then its output will always be c.*
- *If γ is a random gate γ^{rand}, then, we assume that it produces a fresh random element in $\mathbb{F}$ in each round.*

The output of the ith round *is read off from the output gates and denoted $b_i \in \mathbb{F}^{|\Gamma.\mathsf{out}|}$. The* state after the ith round *is contained in the memory gates and denoted k_i. For $k \in \mathbb{F}^{|\Gamma.\mathsf{mem}|}$ and a sequence of inputs $(a_1, \ldots, a_m)$ (where each $a_i \in \mathbb{F}^{|\Gamma.\mathsf{inp}|}$) let $\Gamma(k, a_1, \ldots, a_m)$ denote the sequence $(B_1, \ldots, B_m)$ where each B_i is the output of Γ with $k_0 = k$ and inputs $a_1, \ldots, a_m$ in rounds $1, 2, \ldots, m$.*

Observe that, because Γ is randomized, hence $\Gamma(k, a_1, \ldots, a_m)$ is a random variable. We define now various adversaries interacting with circuits. We start with the most basic black-box circuit adversary which does not take advantage on any leakage from the circuit.

Definition 9.6 (Black-box circuit adversary)**.** *A* black-box circuit adversary $\mathcal{A}$ *is a machine that adaptively interacts with a circuit Γ via the input and output interface. Then $\mathsf{out}\left(\mathcal{A} \overset{bb}{\leftrightarrows} \Gamma(k)\right)$ denotes the output of $\mathcal{A}$ after interacting with Γ whose initial memory state is $k_0 = k$.*

We can now formally define adversaries against circuits that suffer from various leakages. We start first with the noisy-leakage model.

Definition 9.7 (δ-noisy circuit adversary)**.** *A* δ-noisy circuit adversary $\mathcal{A}$ *is a black-box circuit adversary that has the following additional ability: after each ith round $\mathcal{A}$ gets some partial information about the internal state of the computation via the noisy leakage functions. More precisely: let $(X_1, \ldots, X_\ell)$ be the random variable denoting the values on the wires of $\Gamma(k)$ in the ith round. Then $\mathcal{A}$ plays the role of a δ-noisy adversary in a game against $(X_1, \ldots, X_\ell)$ (see Definition 8.7). Namely, he choses a sequence $\{\mathsf{L}_{\mathsf{Noisy}_i} : \mathbb{F} \to \mathcal{Y}\}_{i=1}^{\ell}$ of functions such that every $\mathsf{L}_{\mathsf{Noisy}_i}$ is δ_i-noisy for some $\delta_i \leq \delta$ and he receives $\mathsf{L}_{\mathsf{Noisy}_1}(X_1), \ldots, \mathsf{L}_{\mathsf{Noisy}_\ell}(X_\ell)$. Let $\mathsf{out}\left(\mathcal{A} \overset{noisy}{\leftrightarrows} \Gamma(k)\right)$ denote the output of such an $\mathcal{A}$ after interacting with Γ whose initial memory state is $k_0 = k$.*

Similarly, we replace the δ-noisy adversary with an ϵ-random probing adversary and obtain the following definition.

Definition 9.8 (ϵ-random probing circuit adversary)**.** *An* ϵ-random probing circuit adversary $\mathcal{A}$ *is a black-box circuit adversary that has the following additional ability: after each ith round $\mathcal{A}$ gets some partial information about the internal state of the computation via ϵ_i-identity functions. More precisely: let $(X_1, \ldots, X_\ell)$ be the random variable*

denoting the values on the wires of $\Gamma(k)$ *in the ith round. Then* $\mathcal{A}$ *plays the role of a* ϵ*-random probing adversary in a game against* $(X_1, \ldots, X_\ell)$ *(see Definition 8.3). Namely, after each ith round,* $\mathcal{A}$ *choses a sequence* $(\epsilon_1, \ldots, \epsilon_\ell)$ *such that each* $\epsilon_i \leq \epsilon$ *and he learns* $\varphi_1(X_1), \ldots, \varphi_\ell(X_\ell)$*, where each* φ_i *is the* ϵ_i*-identity function. Let* $\mathsf{out}\left(\mathcal{A} \overset{\mathsf{rnd}}{\leftrightarrows} \Gamma(k)\right)$ *denote the output of such* $\mathcal{A}$ *after interacting with* Γ *whose initial memory state is* $k_0 = k$*.*

Analogously we can replace the "δ-noisy adversary" with the "t-threshold probing adversary" to obtain the following definition.

Definition 9.9 (t-threshold probing circuit adversary)**.** *An* t-threshold probing circuit adversary $\mathcal{A}$ *is a black-box circuit adversary that has the following additional ability: after each ith round* $\mathcal{A}$ *gets some partial information about the internal state of the computation by learning at most* t *values from each encoding. More precisely: let* $(X_1, \ldots, X_\ell)$ *be the random variable denoting the values on the wires of* $\Gamma(k)$ *in the ith round. Then* $\mathcal{A}$ *plays the role of a* t*-threshold probing adversary in a game against* $(X_1, \ldots, X_\ell)$ *(see Definition 8.1). Namely, after each ith round* $\mathcal{A}$ *learns* t *elements of* $(X_1), \ldots, (X_\ell)$*. Let* $\mathsf{out}\left(\mathcal{A} \overset{\mathsf{thr}}{\leftrightarrows} \Gamma(k)\right)$ *denote the output of such* $\mathcal{A}$ *after interacting with* Γ *whose initial memory state is* $k_0 = k$*.*

We can now define properly the resilience of a circuit against leakage. As a preliminary, we define formally the implementation of a circuit.

Definition 9.10 (Implementation of a circuit)**.** *Consider two stateful circuits* Γ *and* Γ' *(over some field* $\mathbb{F}$*) and a randomized encoding function* Enc*. We say that* Γ' *is a* implementation of a circuit Γ w.r.t. Enc *if for every* $k \in \mathbb{F}^{|\Gamma.\mathsf{inp}|}$*: the input-output behavior of* $\Gamma(k)$ *and* $\Gamma'(\mathsf{Enc}(k))$ *is identical, i.e., for every sequence of inputs* $a_1, \ldots, a_m$ *and outputs* $b_1, \ldots, b_m$ *we have*

$$\Pr[\Gamma(k, a_1, \ldots, a_m) = (b_1, \ldots, b_m)] = \Pr[\Gamma'(\mathsf{Enc}(k), a_1, \ldots, a_m) = (b_1, \ldots, b_m)] \,.$$

Definition 9.11 ((δ, ξ)-noise resilient implementation of a circuit)**.** *Consider two stateful circuits* Γ *and* Γ' *(over some field* $\mathbb{F}$*) and a randomized encoding function* Enc*. We say that* Γ' *is a* (δ, ξ)-noise resilient implementation of a circuit Γ w.r.t. Enc *if* Γ' *is an implementation of* Γ *w.r.t.* Enc *and for every* δ*-noisy circuit adversary* $\mathcal{A}$ *there exists a black-box circuit adversary* $\mathcal{S}$ *such that*

$$\Delta\left(\mathsf{out}\left(\mathcal{S} \overset{\mathsf{bb}}{\leftrightarrows} \Gamma(k)\right) \;;\; \mathsf{out}\left(\mathcal{A} \overset{\mathsf{noisy}}{\leftrightarrows} \Gamma'(\mathsf{Enc}(k))\right)\right) \leq \xi \,. \tag{9.18}$$

Definition 9.12 ((ϵ, ξ)-random probing resilient implementation of a circuit)**.** *Consider two stateful circuits* Γ *and* Γ' *(over some field* $\mathbb{F}$*) and a randomized encoding function* Enc*. We say that* Γ' *is a* (ϵ, ξ)-random probing resilient implementation of a circuit Γ w.r.t. Enc *if* Γ' *is an implementation of* Γ *w.r.t.* Enc *and for every* ϵ*-random-probing*

circuit adversary $\mathcal{A}$ there exists a black-box circuit adversary $\mathcal{S}$ such that

$$\Delta\left(\mathit{out}\left(\mathcal{S} \overset{bb}{\leftrightarrows} \Gamma(k)\right)\ ;\ \mathit{out}\left(\mathcal{A} \overset{rnd}{\leftrightarrows} \Gamma'(\mathsf{Enc}(k))\right)\right) \leq \xi\ .$$

Definition 9.13 ((t,ξ)-threshold probing resilient implementation of a circuit)**.** *Consider two stateful circuits Γ and Γ' (over some field $\mathbb{F}$) and a randomized encoding function* Enc*. We say that Γ' is a* (t,ξ)-threshold probing resilient implementation of a circuit Γ w.r.t. Enc *if Γ' is an implementation of Γ w.r.t.* Enc *and for every t-threshold-probing circuit adversary $\mathcal{A}$ there exists a black-box circuit adversary $\mathcal{S}$ such that*

$$\Delta\left(\mathit{out}\left(\mathcal{S} \overset{bb}{\leftrightarrows} \Gamma(k)\right)\ ;\ \mathit{out}\left(\mathcal{A} \overset{thr}{\leftrightarrows} \Gamma'(\mathsf{Enc}(k))\right)\right) \leq \xi\ .$$

Definition 9.14 (Implementation of a circuit with efficient simulation)**.** *In all the cases above, we will say that Γ' is an implementation* Γ with efficient simulation *if the simulator $\mathcal{S}$ works in time polynomial in $\Gamma' \cdot |\mathbb{F}|$ as long as $\mathcal{A}$ is poly-time and the noise functions specified by $\mathcal{A}$ are efficiently decidable.*

9.6.2 Implementing a Secure Circuit Compiler

In this section, we describe the circuit compiler of [ISW03], generalized to larger fields in [RP10]. As seen in the previous section, we need to define first an encoding function. We will use the following standard function.

Definition 9.15 (Additive masking (Enc_+))**.** *Let $d \in \mathbb{N}$ be a parameter. An element $x \in \mathbb{F}$ is encoded with the* additive masking encoding function Enc_+ *as $\mathsf{Enc}_+(x) \coloneqq (X_1, \ldots, X_d)$, where $X_1, \ldots, X_d$ are uniformly distributed in $\mathbb{F}$ such that $X_1 + \cdots + X_d = x$. We denote the inverse operation Dec_+.*

Let Γ be a stateful arithmetic circuit. At a high level, each wire w in the original circuit Γ is represented by a *wire bundle* in Γ', consisting of d wires $\boldsymbol{w} = (w_1, \ldots, w_d)$, that carry an *encoding* of w. The gates in C are replaced gate-by-gate with so-called *gadgets*, computing on encoded values. The main difficulty is to construct gadgets that remain "secure" even if their internals may leak.
Because the transformed gadgets in Γ' operate on encodings, Γ' needs to have a subcircuit at the beginning that encodes the inputs and another subcircuit at the end that decodes the outputs. We will now show how to convert every gate into a gadget.

Input encoding. The input encoding is easy to implement for our encoding function Enc_+: to encode an input x one simply uses the random gates to generate $d-1$ field elements $x_1, \ldots, x_{d-1}$ and then computes x_d as $x_1 + \cdots + x_{d-1} - x$. Clearly, this can be done using d addition and subtraction gates. Recall that the memory gates of Γ are assumed to be preloaded with field elements that already encode k using the encoding Enc_+, hence there is no need to encode k.

Constant gates. Each constant gate $\gamma_c^{\mathsf{const}}$ in Γ can be transformed into d constant gates in Γ', the first of them being $\gamma_c^{\mathsf{const}}$ and the remaining ones being $\gamma_0^{\mathsf{const}}$. This is trivially correct as $c = c + 0 + \cdots + 0$.

Random gates. Every random gate γ^{rand} in Γ is transformed into d random gates in Γ'. This works as, clearly, a uniformly random encoding $(X_1, \ldots, X_d)$ encodes a uniformly random element of $\mathbb{F}$.

What remains to show is how the operation (addition, subtraction, and multiplication) gates are handled. Consider a gate γ in Γ. Let a and b be its input wires and let $\boldsymbol{a} = (a_1, \ldots, a_d)$ and $\boldsymbol{b} = (b_1, \ldots, b_d)$ be their corresponding wire bundles in Γ'. Let the output wire bundle in Γ' be $(c_1, \ldots, c_d)$. The cases when γ is an addition or subtraction gate are actually easy to deal with, thanks to the linearity of the encoding function.

Addition gates. If γ is an addition gate γ^+, each c_i can be computed using an addition gate γ^+ in Γ' with input wires a_i and b_i (this is obviously correct as $(a_1 + b_1) + \cdots + (a_d + b_d) = (a_1 + \ldots a_d) + (b_1 + \cdots + b_d)$).

Subtraction gates. If γ is an subtraction gate γ^-, each c_i can be computed using an subtraction gate γ^- in Γ' with input wires a_i and b_i (this is obviously correct as $(a_1 - b_1) + \cdots + (a_d - b_d) = (a_1 + \ldots a_d) - (b_1 + \cdots + b_d)$).

Multiplication gates. When γ is the multiplication gate $\gamma^\times$, the change is more tricky. In this case, the circuit Γ' generates for every $1 \le i < j \le d$ a random field element $z_{i,j}$ (this is done using the random gates in Γ'). Then, for every $1 \le j < i \le d$ it computes

$$z_{i,j} \coloneqq a_i b_j + a_j b_i - z_{j,i} \ ,$$

and finally it computes each c_i (for $i = 1, \ldots, d$) as

$$c_i \coloneqq a_i b_i + \sum_{j \neq i} z_{i,j} \ .$$

In other words, we have

$$c_i \coloneqq a_i \sum_{j \neq i} b_j + b_i \sum_{j \neq i} a_j + a_i b_i + \sum_{j \neq i} \begin{cases} -z_{j,i} & \text{if } j < i \\ z_{i,j} & \text{else.} \end{cases}$$

To see why this computation is correct, consider the sum $c = c_1 + \cdots + c_d$ and observe that, for $i < j$, every $z_{i,j}$ in it appears exactly once with plus sign and once with a minus sign, and hence it cancels out. Moreover each term $a_i b_j$ appears in the formula for c

exactly once. Hence c is equal to

$$\sum_{i,j=1}^{d} a_i b_j = \left(\sum_{i=1}^{d} a_i\right)\left(\sum_{j=1}^{d} b_j\right) = ab\ .$$

It is straightforward to verify that the total number of gates in this gadget is $3.5 \cdot d^2$.

Refreshing. The multiplication gadget above turns out to be useful as a building block for "refreshing" of the encoding. More concretely, suppose we have a wire bundle $\boldsymbol{a} = (a_1, \ldots, a_d)$ and we wish to obtain another bundle $\boldsymbol{b} = (b_1, \ldots, b_d)$ such that $\boldsymbol{b}$ is a fresh encoding of $\mathsf{Dec}_+(\boldsymbol{a})$. This can be achieved by a Refresh sub-gadget constructed as follows. First, create an encoding $(1, 0, \ldots, 0)$ of 1 (using d constant gates), and multiply $(1, 0, \ldots, 0)$ and $\boldsymbol{a}$ together using the multiplication protocol above. As $(1, 0, \ldots, 0)$ is an encoding of 1, hence the result will be an encoding of $1 \cdot a = a$. The multiplication can be done with $3.5 \cdot d^2$ gates, and hence altogether this gadget uses $3.5 \cdot d^2 + 2 \cdot d$ gates.

Output gates. We can now use the Refresh sub-gadget to construct the output gadgets in Γ'. Let γ^{out} be an output gate in Γ with an input wire a. Then in Γ' it is transformed into the following: let $\boldsymbol{a}$ be the wire bundle corresponding to a. First apply the Refresh sub-gadget, and then calculate the sum $b_1 + \cdots + b_d$, where $(b_1, \ldots, b_d)$ is the output of Refresh, and output the result.

Memory gates. The refreshing gadget is also useful to provide security of the memory encoding in the multi-round scenario. More precisely, we assume that every memory state gets refreshed at the end of each round by the Refresh procedure. It is easy to see that without this "refreshing", the contents of the memory would eventually leak completely to the adversary even if he probes a very limited number (e.g., 1) of wires in each round.
This concludes the description of the compiler. For more details, we refer the reader to the original paper [ISW03].

9.6.3 Security in the Probing Model [ISW03]

In [ISW03], it is shown that the compiler from the previous section is secure against probing attacks in which the adversary can probe at most $\lfloor (d-1)/2 \rfloor$ wires in each round.[4] This parameter may be a bit disappointing as the number of probes that the adversary needs to break the security does not grow with the size of the circuit. This assumption may seem particularity unrealistic for large circuits Γ. Fortunately, [ISW03] also shows a small modification of the construction from Section 9.6.2 that is resilient to a larger number of probes, provided that the number of probes from each gadget is

[4] Strictly speaking, the proof of [ISW03] considers only the case when $\mathbb{F} = \mathrm{GF}(2)$. It was observed in [RP10] that it can be extended to any finite field, as the only properties of GF(2) that are used in the proof are the field axioms.

bounded. Before we present it, let us argue why the original construction is not secure against such attacks. To this end, assume that our circuit Γ has a long sequence of wires $a_1, \ldots, a_m$, where each a_i (for $i > 1$) is the result of adding to a_{i-1} (using an addition gate) a 0 constant (that was generated using a $\gamma_0^{\mathsf{const}}$ gate). It is easy to see that in the circuit Γ' all the wire bundles $\boldsymbol{a_1}, \ldots, \boldsymbol{a_m}$ (where each $\boldsymbol{a_i}$ corresponds to a_i) will be identical. Hence, the adversary that probes even a single wire in each addition gadget in Γ' will learn the encoding of a_1 completely as long as $m \geq d$. Fortunately, one can deal with this problem by "refreshing" the encoding after each subtraction and addition gate exactly in the same way as done before, i.e. by using the Refresh sub-gadget.

Lemma 9.16 ([ISW03])**.** *Let Γ be an arbitrary stateful arithmetic circuit over some field $\mathbb{F}$. Let Γ' be the circuit that results from the procedure described above. Then Γ' is a $(\lfloor (d-1)/2 \rfloor \cdot |\Gamma|, 0)$-threshold-probing resilient implementation of a circuit Γ (with efficient simulation), provided that the adversary does not probe each gadget more than $\lfloor (d-1)/2 \rfloor$ times in each round.*

We notice that [ISW03] also contains a second transformation with blow-up $\tilde{O}(d|\Gamma|)$. It may be possible that this transformation can provide better noise parameters as is achieved by Theorem 9.20. However, due to the hidden parameters in the $\tilde{O}$-notation we do not get a straightforward improvement of our result. In particular, using this transformation the size of the transformed circuit depends also on an additional statistical security parameter, which will affect the tolerated noise level.

9.6.4 Resilience to Noisy Leakage from the Wires

We now show that the construction from Section 9.6.3 is secure against the noisy leakage. More precisely, we show the following:

Theorem 9.17. *Let Γ be an arbitrary stateful arithmetic circuit over some field $\mathbb{F}$. Let Γ' be the circuit that results from the procedure described in Section 9.6.3. Then Γ' is a $(\delta, |\Gamma| \cdot \exp(-d/12))$-noise-resilient implementation of Γ (with efficient simulation), where*

$$\delta := ((28d + 16)|\mathbb{F}|)^{-1} = O(1/(d \cdot |\mathbb{F}|)).$$

Proof. Let $\mathcal{A}$ be a δ-noisy circuit adversary attacking Γ'. We construct an efficient black-box simulator $\mathcal{S}$ such that for every k it holds that

$$\Delta\left(\mathsf{out}\left(\mathcal{S} \overset{\mathsf{bb}}{\leftrightarrows} \Gamma(k)\right) \;;\; \mathsf{out}\left(\mathcal{A} \overset{\mathsf{noisy}}{\leftrightarrows} \Gamma'(\mathsf{Enc}(k))\right)\right) \leq |\Gamma| \cdot \exp(-d/12)\,. \tag{9.19}$$

Observe that in our construction, every gate gets transformed into a gadget of at most $3.5 \cdot d^2 + 2 \cdot d$ gates. As each gate can have at most 2 inputs hence the total number of wires in a gadget is $\ell := 7 \cdot d^2 + 4 \cdot d$. Let $\gamma^1, \ldots, \gamma^{|\Gamma|}$ be the gates of Γ. For each $i = 1, \ldots, \ell$ let the wires in the gadget in Γ' that corresponds to γ^i be denoted $(x_1^i, \ldots, x_\ell^i)$. As

$\delta = d/(4\ell|\mathbb{F}|)$, we can use Corollary 9.4 and simulate the noise from each $(x_1^i, \ldots, x_\ell^i)$ by a $(d/2-1)$-threshold-probing adversary $\mathcal{S}^i$ working in time polynomial in $\ell \cdot |\mathcal{X}|$. The simulation is perfect, unless $\mathcal{S}^i$ outputs $\perp$, which, by Corollary 9.4 happens with probability at most $\exp(-d/12)$. Hence, by the union-bound the probability that *some* $\mathcal{S}^i$ outputs $\perp$ is at most $|\Gamma| \cdot \exp(-d/12)$. Denote this event $\mathcal{E}$.
From Lemma 9.16, we know that every probing adversary that attacks Γ' by probing at most $\lfloor (d-1)/2 \rfloor \geq d/2 - 1$ wires from each gadget can be perfectly simulated in polynomial time by an adversary $\mathcal{S}$ with a black-box access to Γ . Hence, $\mathcal{A}$ can also be simulated perfectly by a black-box access to Γ conditioned on the fact that $\mathcal{E}$ did not occur. Hence we get

$$\Delta\left(\mathsf{out}\left(\mathcal{S} \overset{\mathsf{bb}}{\leftrightarrows} \Gamma(k)\right) | \neg\mathcal{E} \;;\; \mathsf{out}\left(\mathcal{A} \overset{\mathsf{noisy}}{\leftrightarrows} \Gamma'(\mathsf{Enc}(k))\right)\right) = 0\,.$$

This, by Lemma 2.5 (Section 2.5), implies (9.19). Obviously $\mathcal{S}$ works in time polynomial in $|\Gamma| \cdot d^2 \cdot |\mathbb{F}|$, which is polynomial in $\Gamma' \cdot |\mathbb{F}|$. This finishes the proof. □

In short, this theorem is proven by combining Corollary 9.4 that reduces the noisy adversary to the probing adversary, with Lemma 9.16 that shows that the construction from Section 9.6.3 is secure against probing.

9.6.5 Resilience to Noisy Leakage from the Gates

The model of Prouff and Rivain [PR13] is actually slightly different than the one considered in the previous section. The difference is that they assume that the noise is generated by the *gates*, not by the *wires*. This can be formalized by assuming that each noise function $\mathsf{L}_{\mathsf{Noisy}}$ is applied to the "contents of a gate". Observe that the contents of each gate γ can be described by at most 2 field elements: obviously if γ is a random gate, output gate, or memory gate then its entire state in a given round can be described by one field element, and if γ is an operation gate, then it can be described by two field elements that correspond to γ's input. Hence, without loss of generality we can assume that the noise function is defined over the domain $\mathbb{F} \times \mathbb{F}$.
To be more formal, we need to define an adversary that exploits leakage from gates.

Definition 9.18 (δ-gate-noisy circuit adversary)**.** *We define a* δ-gate-noisy circuit adversary $\mathcal{A}$ *as a machine that, besides of having black box access to a circuit* $\Gamma(k)$*, can, after each* i*th round, get some partial information about the internal state of the computation via* δ*-noisy leakage functions applied to the gates As described above, the content of a gate can be modeled as an element in* $\mathbb{F} \times \mathbb{F}$*.*
Let $\mathsf{out}\left(\mathcal{A} \overset{\mathsf{g\text{-}noisy}}{\leftrightarrows} \Gamma(k)\right)$ *denote the output of such* $\mathcal{A}$ *after interacting with* Γ *whose initial memory state is* $k_0 = k$*.*

We can accordingly modify the definition of noise-resilient circuit implementations (as in Definition 9.11).

Definition 9.19 ((δ,ξ)-gate-noise resilient implementation of a circuit)**.** *We say that* Γ' *is a* (δ,ξ)-gate-noise resilient implementation of a circuit Γ w.r.t. Enc *if it is a implementation of* Γ *and if for every* k *and every* δ*-gate-noisy circuit adversary* $\mathcal{A}$ *described above there exists a black-box circuit adversary* $\mathcal{S}$ *working in time polynomial in* $\Gamma' \cdot |\mathbb{F}|$ *such that*

$$\Delta\left(\mathit{out}\left(\mathcal{S} \overset{\mathit{bb}}{\leftrightarrows} \Gamma(k)\right)\ ;\ \mathit{out}\left(\mathcal{A} \overset{\mathit{g\text{-}noisy}}{\leftrightarrows} \Gamma'(\mathsf{Enc}(k))\right)\right) \leq \xi\ . \tag{9.20}$$

It turns out that the transformation from Section 9.6.3 also works in this model, although with different parameters. More precisely we have the following theorem.[5]

Theorem 9.20. *Let* Γ *be an arbitrary stateful arithmetic circuit over some field* $\mathbb{F}$*. Let* Γ' *be the circuit that results from the procedure described in Section 9.6.3. Then* Γ' *is a* $(\delta, |\Gamma| \cdot \exp(-d/24))$*-gate-noise-resilient implementation of* Γ *(with efficient simulation), where*

$$\delta := \left((28d+16)\cdot|\mathbb{F}|^2\right)^{-1} = O(1/(d\cdot|\mathbb{F}|^2))\ . \tag{9.21}$$

Proof. The proof is similar to the one of Theorem 9.17 so we only describe the key differences. Let $\mathcal{A}$ be a δ-noisy adversary. The number ℓ corresponds now to the number of gates in each gadget, and hence it is equal to $3.5{\cdot}d^2{+}2{\cdot}d$. It is therefore straightforward to calculate that δ defined in (9.21) is equal to $(d/2)/(4\ell\cdot|\mathbb{F}|^2)$. As the $\mathsf{L}_{\mathsf{Noisy}}$ function has domain of size $|\mathbb{F}|^2$, we can use Corollary 9.4 obtaining that $\mathcal{A}$ can be simulated by an adversary $\mathcal{S}$ that probes each gadget in less that $d/2$ positions. As now each "position" corresponds to a gate in the circuit, hence the adversary needs to probe up to two wires to determine its value. Therefore $\mathcal{S}$ probes less than d wires in each gadget. As d is now $1/2$ of what it was in the proof of Corollary 9.4, hence the error probability changes from $\exp(-d/12)$ to $\exp(-d/24)$. □

Comparison with [PR13]

As described in the introduction, our main advantage over [PR13] is the removal of the assumption about the existence of the leak-free gates, a stronger security model — chosen message attack, instead of a random message attack, and a more meaningful security statement. Still, it is interesting to compare our noise parameters with the parameters of [PR13]. Let us analyze how much noise is needed by [PR13] to ensure that the adversary obtains exponentially small information from leakage. The reader should keep in mind that both in our work and in [PR13] "more noise" means that a

[5] Note that our result holds only when the number of shares is large. For small values of d (e.g., $d = 2, 3, 4$) like those considered in [SVCO+10], our result does not give meaningful bounds. This is similar to the work of Prouff and Rivain [PR13] and it is an interesting open research question to develop security models that work for small security parameters. We show a step towards that direction in Chapter 10.

certain quantity, δ, in our case, is *smaller*. Hence, the larger δ is, the stronger the result becomes (as it means that *less* noise is required for the security to hold).
The main result of [PR13] is Theorem 4 on page 154. Unfortunately, the statement of this theorem is asymptotic treating $|\mathbb{F}|$ as constant, and hence to get a precise bound on how much noise is required one needs to inspect the proof. The bound on the noise can be deduced from the part of the proof entitled "Security of Type 3 Subsequences", where the required noise is inversely-proportional to "$\lambda(d)$", and this last value is linear in $d \cdot |\mathbb{F}|^3$ (note that $|\mathbb{F}|$ is denoted by N in [PR13], and d is a security parameter identical to ours). Hence their δ is $O(1/(d \cdot |\mathbb{F}|^3))$. Our Lemma 9.20 requires a more liberal bound (cf. (9.21)), and hence can be viewed as stronger, however, as explained in Section 9.4, the notion of distance in [PR13] is slightly different than the standard "statistical distance" that we use. Fortunately, one can use (8.3) to translate our bound into their language. It turns out that in this case our and their bounds are asymptotically identical ($O(1/(d \cdot |\mathbb{F}|^3))$). This is shown in Corollary 9.21 below. Note that this translation is unidirectional, in the sense that their "$O(1/(d \cdot |\mathbb{F}|^3))$" bound does *not* imply a bound "$O(1/(d \cdot |\mathbb{F}|^2))$" in our sense.

Corollary 9.21. *Let Γ be an arbitrary stateful arithmetic circuit over some field $\mathbb{F}$. Let Γ' be the circuit that results from the procedure described in Section 9.6.3. Then Γ' is a $(\delta', |\Gamma| \cdot \exp(-d/24))$-gate-noise-resilient implementation of Γ (with efficient simulation)* when the noise is defined using the β distance*, where*

$$\delta' = \left((14d + 8) \cdot |\mathbb{F}|^3\right)^{-1} = O(1/(d \cdot |\mathbb{F}|^3)) \ .$$

Proof. From (8.3) with $\mathcal{X} = \mathbb{F} \times \mathbb{F}$ it follows that if $\mathsf{L}_{\mathsf{Noisy}}$ is δ'-noisy with respect to the β distance, then it is $(|\mathbb{F}| \cdot \delta'/2)$-noisy in the standard sense. As this last value is equal to δ defined in (9.21), hence we can use Theorem 9.20 obtaining that Γ' is a $(\delta', |\Gamma| \cdot \exp(-d/24))$-noise-resilient implementation of Γ when the noise is defined using the β distance. □

Chapter 10

From Theory to Practice

This chapter presents part of some joint work with Prof. S. Faust and Prof. F.-X. Standaert which was published in [DFS15a].

10.1 Assessing Security of Concrete Devices

In view of what was shown in the previous chapter, one of the main remaining questions regarding the security of the masking countermeasure is whether its proofs can be helpful in the security evaluation of concrete devices. That is, can we state theorems for masking so that the hypotheses can be easily fulfilled by hardware designers, and the resulting guarantee is reflective of the actual security level of the target implementation. For this purpose, we first observe that the proofs from the previous chapter as well as the proofs in [PR13] express their hypothesis for the amount of noise in the shares' leakages based on a statistical distance. This is in contrast with the large body of published work where the mutual information metric introduced in [SMY09] is estimated for various implementations (e.g. [BFGV12, CDGM14, FMPR10, GM11, GSP13, MS11, PR10, RKSF11, RP12, SPV12, VMKS12]). The latter metric generally carries more intuition (see, e.g. [BJV04] in the context of linear cryptanalysis), and benefits from recent advances in leakage certification, allowing to make sure that its estimation is accurate and based on sound assumptions [DSV14]. Hence, in this chapter, we first provide a useful link between the statistical distance and mutual information, and also connect them with easy-to-interpret (but more specialized) tools such as the Signal-to-Noise Ratio (SNR). We then re-state some theorems presented in Chapter 9 based on the mutual information metric in two practically relevant scenarios. Namely, we consider both the security of an idealized implementation with a "leak-free refreshing" of the shares, and the one of a standard ISW-like encoding (i.e. capturing any type of leaking computation).
Interestingly, the implementation with leak-free refreshing corresponds to the frequently investigated (practical) context where a side-channel attack aims at key recovery, and only targets the d shares' leakage of a so-called sensitive intermediate variable (i.e. that depends on the plaintext and key) [CPR07]. So, despite being less interesting from a

theoretical point of view, this scenario allows us to compare the theorem bounds with concrete attacks. Taking advantage of this comparison, we discuss the bounds' tightness and separate parameters that are physically motivated from more "technical ones" (that most likely result of proof artifacts).

In [DFS15a], we also discuss the independence requirement between the leakages used in most theoretical work included what we presented in the previous chapter. We suggested there some heuristics to analyze non-independent leakages. Indeed, concrete experiments have shown that some deviations from this assumption frequently occur in practice (see, e.g., [BGG+14, CGP+12b, MPG05, RSV+11]). We also consider there the tradeoff between measurement complexity and time complexity when dealing with divide-and-conquer attacks. Previously known approaches for this purpose were based on launching key enumeration and/or rank estimation algorithms for multiple attacks, and to average results to obtain a success rate [VGRS12, VGS13]. We provide in [DFS15a] an alternative solution, where success rates (possibly obtained from estimations of the mutual information metric) are estimated/bounded for all the target key bytes of the divide-and-conquer attack first, and the impact of enumeration is evaluated only once afterwards. We connect the problem with a non-linear programming problem and provided heuristics to estimate good bounds on the enumeration cost. However, we will not cover these results in this thesis as they are mainly heuristic and recommend the interested reader to read [DFS15a].

Summarizing, the combination of these observations highlights that the security evaluation of a masked implementation boils down to the estimation of the mutual information between its shares and their corresponding leakages. Incidentally, the tools introduced in this chapter apply identically to unprotected implementations, or implementations protected with other countermeasures, as long as one can estimate the same mutual information metric for the target intermediate values. Therefore, our results clarify the long standing open question whether the (informal) link between information theoretic and security metrics in the Eurocrypt 2009 evaluation framework [SMY09] can be proved formally. They also have important consequences for certification bodies, as they translate the (worst-case) side-channel evaluation problem into the well-defined challenge of estimating a single metric, leading to significantly reduced evaluation costs.

10.2 Background

We recall that we use an additive masking scheme (see Section 8.1) to protect a sensitive value against side-channel attacks. A typical example of such a sensitive value would be the output of an S-box computation. Like in the previous chapter, a value $y \in \mathbb{F}$ is shared using d shares $y_1, \ldots, y_d$. As before, we will assume that each share leaks. In this chapter, we will represent the leakage as the output of a noisy leakage function $\mathsf{L}_{\mathsf{Noisy}}$ as it is the most meaningful model in practice.

10.2.1 Evaluation Metrics

In general, i.e., without assumptions on the leakage distribution), the noise condition on the shares can be expressed with an information theoretic metric. The Mutual Information (MI) advocated in [SMY09] is the most frequently used candidate for this purpose in the practical community. For this we need to recall some well-known definitions from information theory.

Definition 10.1 (Entropy)**.** *Let Y be a discrete random variable over the set $\mathcal{Y}$. Then, the* entropy *of Y (written $\mathrm{H}[Y]$) is*

$$\mathrm{H}[Y] \coloneqq -\sum_{y\in\mathcal{Y}} \Pr[Y=y]\log\left(\Pr[Y=y]\right) \ .$$

Definition 10.2 (Conditional Entropy)**.** *Let X (resp. Y) be a discrete random variable over the set $\mathcal{X}$ (resp. $\mathcal{Y}$). Then, the* condition entropy *of Y given X (written $\mathrm{H}[Y|X]$) is*

$$\mathrm{H}[Y \mid X] \coloneqq \sum_{x\in\mathcal{X}} \Pr[X=x]\ \mathrm{H}[Y \mid X=x] \ .$$

We can finally define mutual information.

Definition 10.3 (Mutual Information)**.** *Let X (resp. Y) be a discrete random variable over the set $\mathcal{X}$ (resp. $\mathcal{Y}$). Then, the* mutual information *between X and Y (written $\mathsf{MI}(X;Y)$) is*

$$\begin{aligned}\mathsf{MI}(X;Y) &\coloneqq \mathrm{H}[X] - \mathrm{H}[X \mid Y] = \mathrm{H}[X] + \mathrm{H}[Y] - \mathrm{H}[X,Y] \\ &= -\sum_{x\in\mathcal{X}} \Pr[X=x] + \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} \Pr[X=x, Y=y]\log(\Pr[X=x \mid Y=y]) \ .\end{aligned}$$

We recall that we will sometimes use the notation $\Pr[y] \coloneqq \Pr[Y=y]$ when clear from the context.
We can now express the mutual information between the share and its leakage in the following way:

$$\begin{aligned}&\mathsf{MI}(Y_i; \mathsf{L}_{\mathsf{Noisy}}(Y_i)) \\ &\quad = \mathrm{H}[Y_i] + \sum_{y_i\in\mathcal{Y}} \Pr[y_i] \cdot \sum_{\ell\in\mathcal{L}} \Pr[\mathsf{L}_{\mathsf{Noisy}}(Y_i)=\ell \mid y_i] \cdot \log \Pr[y_i \mid \mathsf{L}_{\mathsf{Noisy}}(Y_i)=\ell] \ ,\end{aligned} \tag{10.1}$$

where the support of each share is $\mathcal{Y}$ and the support of the leakage $\mathcal{L}$. Note that whenever trying to compute this quantity from an actual implementation, evaluators face the problem that the leakage's probability density function (PDF) is unknown and can only be sampled and estimated. As a result, one then computes the *Perceived Information (PI)*, which is the evaluator's best estimate of the MI [RSV+11]. For simplicity, we will ignore this issue and use the MI in our discussions. Our conclusions would

be identical with the PI.
We can now formally define an adversary working in this model.

Definition 10.4 (t-MI-advesary)**.** *For $t \geq 0$, a* t-MI-adversary on $\mathcal{X}^\ell$ *is an algorithm $\mathcal{A}$ that plays the following scenario against an oracle that knows $(x_1, \ldots, x_\ell) \in \mathcal{X}^\ell$:*

1. *$\mathcal{A}$ specifies a sequence $\{\mathsf{L}_{\mathsf{Noisy}_i} : \mathcal{X} \to \mathcal{Y}\}_{i=1}^{\ell}$ of noisy functions such that for every $\mathsf{L}_{\mathsf{Noisy}_i}$ we have $\mathsf{MI}(x_i; \mathsf{L}_{\mathsf{Noisy}_i}(x_i)) \leq t$ and mutually independent noises.*

2. *$\mathcal{A}$ receives $\mathsf{L}_{\mathsf{Noisy}_1}(x_1), \ldots, \mathsf{L}_{\mathsf{Noisy}_\ell}(x_\ell)$ and outputs some value that we denote by $\mathsf{out}_{\mathcal{A}}(x_1, \ldots, x_\ell)$.*

Like before, $\mathsf{out}_{\mathcal{A}}(x_1, \ldots, x_\ell)$ is a random variable combining $\mathcal{A}$ and the randomness of the functions $\mathsf{L}_{\mathsf{Noisy}_i}$.

The SNR introduced by Mangard at CT-RSA 2004 in [Man04] is of particular interest for our following discussions.

Definition 10.5 (Signal-to-Noise Ratio (SNR))**.** *Let X be some secret random variable and $L_X := f(X) + \mathcal{N}(\sigma^2)$ be its leakage, where f is an arbitrary function and $\mathcal{N}$ follows a Gaussian distribution with expected value 0 and variance σ^2. The* signal-to-noise ratio (SNR) *is defined as*

$$\mathrm{SNR} := \frac{\mathrm{var}(f(X))}{\sigma^2} \,, \tag{10.2}$$

i.e., as the ratio between the variance of the signal and the variance of the noise.
Given N samples $\boldsymbol{L}_X = \{f(X_1) + n_1, \ldots, f(X_N) + n_N\}$, the SNR can be estimated as

$$\mathrm{SNR} = \frac{\hat{\mathrm{var}}_X\left(\hat{\mathsf{E}}_n(\boldsymbol{L}_X)\right)}{\hat{\mathsf{E}}_X\left(\hat{\mathrm{var}}_n(\boldsymbol{L}_X)\right)} \,, \tag{10.3}$$

where $\hat{\mathsf{E}}$ is the sample mean operator[1] *and $\hat{\mathsf{var}}$ is the sample variance*[2]*.*

Summarizing, stating the noise condition based on the MI metric is more general (as it can capture any leakage PDF). By contrast, the SNR provides a simpler and more intuitive condition in a more specific but practically relevant context.
As previously mentioned, some of these metrics can be related under certain conditions. For example, in the context of univariate Gaussian random variables, the MI can be approximated from Pearson's correlation coefficient [MOS11], which was also connected to the SNR by Mangard [Man04]. The combination of those links corresponds to the classical MI bound that can be found in any information theory book, e.g., in Cover and

[1] Given samples $x_1, \ldots, x_N$, the sample mean is defined as $(1/N)\sum_i x_i$.
[2] Given samples $x_1, \ldots, x_N$, the sample variance is defined as $(1/N)\sum_i (x_i - \bar{x})^2$, where $\bar{x}$ is the sample mean of $x_1, \ldots, x_N$.

Thomas [CT06]:

$$\mathrm{MI}(Y_i; \mathsf{L}_{\mathsf{Noisy}}(Y_i)) \approx -\frac{1}{2}\log\left(1 - \left(\frac{1}{\sqrt{(1+\frac{1}{\mathrm{SNR}})}}\right)^2\right) \leq \frac{1}{2}\log\left(1+\mathrm{SNR}\right) \cdot \quad (10.4)$$

In Section 10.3.1, we show that the MI and SD metrics can be connected as well.

10.2.2 Metrics to Quantify the Security Result

Quantifying security requires defining the adversary's goal. Current practical side-channel attacks published in the literature mostly focus on key recovery. In this context, one can easily evaluate the exploitation of the leakages with the success rate defined in [SMY09], i.e. the probability that an adversary recovers the key given the observation of some (typically known or chosen) plaintexts, ciphertexts and leakages.

Definition 10.6 (Success rate for key recovery ($\mathsf{SR}^{\mathsf{kr}}$))**.** *Let $x_1, \ldots, x_d$ be leaking shares of a sensitive value x. We denote by $\mathsf{SR}^{\mathsf{kr}}$ the* success rate for key recovery*, i.e., the probability that any adversary $\mathcal{A}$ recovers x only given the leakage from the shares.*

Key recovery is a weak security notion from a cryptographic point of view. As a result, rigorous proofs for masking such as the one presented in Chapter 9 rather define security using the standard real/ideal world paradigm, which considers two settings: the ideal world where the adversary attacks the algorithm of a cryptographic scheme in a black-box way, and the real world where he additionally obtains leakages. A scheme is said to be secure in the real world, if for any adversary in the real world there exists an adversary in the ideal world. In other words: any attack that can be carried out given the leakages can also be carried out in a black-box manner. A proof of security usually involves constructing an efficient simulator that is able to simulate the leakages just giving black-box access to the attacked cryptographic scheme. Whenever considering this (standard) indistinguishability-based security notion, we will denote the adversary's success probability of distinguishing the two worlds with $\mathsf{SR}^{\mathsf{dist}}$. More formally:

Definition 10.7 (Success rate for distinguishing ($\mathsf{SR}^{\mathsf{dist}}$))**.** *Let Γ be an arithmetic circuit over some field $\mathbb{F}$ and Γ' be a leakage-resilient implementation of Γ in the noisy model. Let $\mathcal{S}$ be an efficient black-box simulator that simulates the leakage. We denote by $\mathsf{SR}^{\mathsf{dist}}$ the* success rate for distinguishing *between the two worlds, i.e.,*

$$\mathsf{SR}^{\mathsf{dist}} \coloneqq \Delta\left(\mathit{out}\left(\mathcal{S} \overset{bb}{\leftrightarrows} \Gamma(k)\right) \;;\; \mathit{out}\left(\mathcal{A} \overset{noisy}{\leftrightarrows} \Gamma'(\mathsf{Enc}(k))\right)\right) .$$

10.3 Making Proofs Concrete: Theory

In this section, we discuss theoretical tweaks allowing to improve the concreteness of masking proofs. Recall that the noisy leakage model describes many realistic side-channel

attacks and allows an adversary to obtain each intermediate value perturbed with a δ-noisy leakage function. As mentioned previously, a leakage function $\mathsf{L}_{\mathsf{Noisy}}$ is called δ-noisy if for a uniformly random variable Y (over the field $\mathbb{F}$) we have $\Delta(Y; Y|\mathsf{L}_{\mathsf{Noisy}}(Y)) = \delta$. In contrast with the conceptually simpler ϵ-probing model, the adversary obtains noisy leakages on each intermediate variable. For example, in the context of a masking $y = y_1 \oplus \cdots \oplus y_d$, he obtains leakage for all the shares y_i, which is more reflective of actual implementations where the adversary can potentially observe the leakage of all these shares, as they are all present in leakage traces. Recall that in Chapter 9, we showed that security against probing attacks implies security against noisy leakages up to a factor $|\mathbb{F}|$. In this section, we first connect the statistical distance (SD) with the mutual information metric (MI), which shows that both can be used to quantify the noise condition required for masking. Next, we provide alternative forms for the theorems presented in Chapter 9 and express

1. the security of the encoding used in (e.g. Boolean) masking and

2. the security of a complete circuit based on the ISW compiler.

10.3.1 From Statistical Distance to Mutual Information

The results from Chapter 9 require to have a bound on the SD between the shares and the shares given the leakage. For different reasons, expressing this distance based on the MI metric may be more convenient in practice (as witnessed by the numerous works where this metric has been computed, for various types of devices, countermeasures and technologies – see the list given in the introduction of this chapter). For example, the MI metric is useful to determine whether the leakage model used in a standard DPA is sound and for analyzing the impact of key enumeration in divide-and-conquer attacks. Very concretely, Equation (10.1) is also expressed in a way that requires summing over the intermediate values first and on the leakages afterwards, which corresponds to the way security evaluations are performed (i.e. fix the target device's state, and then perform measurements). Thus, we now show how to express the SD in function of the MI. We use a previous result from Dodis [Dod12], which proof follows [BTV12], that we rephrase with our notations.

Lemma 10.8 ([Dod12], Lemma 6)**.** *Let $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ be two (possibly dependent) random variables. Then:*

$$\frac{1}{2}\left(\sum_{(x\in\mathcal{X},y\in\mathcal{Y})} |\Pr[X=x, Y=y] - \Pr[X=x]\Pr[Y=y]|\right)^2 \leq \mathsf{MI}(X;Y) \ .$$

In the following, we will typically consider Y as the leakage of X. Using this lemma, we can now express the SD in function of the MI as follows.

Theorem 10.9. *Let $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ be two (possibly dependent) random variables. Then:*

$$2 \cdot \Delta(X; X \mid Y)^2 \leq \mathsf{MI}(X; Y) \ .$$

Proof. The proof follows the proof of [BTV12, Lemma 4.4]. We have:

$$\begin{aligned}
&\sum_{(x \in \mathcal{X}, y \in \mathcal{Y})} |\Pr[X = x, Y = y] - \Pr[X = x] \Pr[Y = y]| \ , \\
&\quad = \sum_{y \in \mathcal{Y}} \Pr[Y = y] \sum_{x \in \mathcal{X}} |\Pr[X = x \mid Y = y] - \Pr[X = x]| \ , \\
&\quad = 2 \cdot \Delta(X; X \mid Y) \ .
\end{aligned}$$

The final result directly derives from Lemma 10.8. □

If we apply Theorem 10.9 to a noisy leakage function and take $X \coloneqq Y_i$, $Y \coloneqq \mathsf{L}_{\mathsf{Noisy}}(Y_i)$, we obtain the following corollary:

Corollary 10.10. *Let Y_i be a share and $\mathsf{L}_{\mathsf{Noisy}}(Y_i)$ its noisy leakage. Then:*

$$2 \cdot \Delta(Y_i; Y_i \mid \mathsf{L}_{\mathsf{Noisy}}(Y_i))^2 \leq \mathsf{MI}(Y_i; \mathsf{L}_{\mathsf{Noisy}}(Y_i)) \ .$$

10.3.2 Security of the Encoding

In this section, we analyze the security of an encoding when m measurements are performed and the encoding is refreshed between each measurements using a leak-free gate. More precisely, we assume that a secret y is secret-shared into d shares $y_1, \ldots y_d$, using an additive masking scheme over a finite field $\mathbb{F}$. Between each measurement, we assume that we take fresh $y_1, \ldots, y_d$ values such that $y = y_1 + \cdots + y_d$ (e.g. it could be the Boolean encoding of Section 8.1). We also assume that this refreshing process does not leak. We recall Lemma 9.2 from Chapter 9 that relates the random probing model to the noisy model.
Lemma 9.2 enables us to work directly in the random-probing model instead of the noisy leakage model. Next, we study the security of the encoding. As mentioned in introduction, the adversary's goal in this case is to recover the encoded value, which is equivalent to key recovery if this value is a key. In order to make it completely comparable with actual attacks, we also add the number of measurements m used by the adversary as a parameter in our bounds.

Theorem 10.11. *Let d be the number of shares used for a key encoding, m be the number of measurements, and $t \leq 2/|\mathbb{F}|^2$. Then, if we refresh the encoding in a leak-free manner*

between each measurement, the probability of success of a key recovery t-MI-adversary under independent leakage is:

$$\mathsf{SR}^{\mathsf{kr}} \leq 1 - \left(1 - \left(|\mathbb{F}|\sqrt{t/2}\right)^{d}\right)^{m} . \tag{10.5}$$

Proof. In the random probing model with parameter ϵ, an adversary learns nothing about the secret if there is at least one share that did not leak. As all the measurements are independent and we use leak-free refreshing gates, we have:

$$\mathsf{SR}^{\mathsf{kr}} \leq 1 - \left(1 - \epsilon^{d}\right)^{m} . \tag{10.6}$$

Let A be a t-MI-adversary on $\mathbb{F}^d$. From Corollary 10.10, we know that A implies a $\sqrt{t/2}$-noisy-adversary on $\mathbb{F}^d$ and, by Lemma 9.2, we obtain a $|\mathbb{F}|\sqrt{t/2}$-random-probing adversary on $\mathbb{F}^d$. Letting $\epsilon := |\mathbb{F}|\sqrt{t/2}$ in (10.6) gives us the result. □

Note that (10.6) focuses on the impact of the adversary's measurement complexity m on the success rate, which is usually the dominating factor in concrete side-channel analyses. The impact of time complexity when considering key enumeration is discussed in [DFS15a, Section 4.2]. Besides and for readability, this equation only includes the terms corresponding to attacks taking advantage of the leakages. We ignore the additional terms corresponding to mathematical cryptanalysis (e.g. exhaustive search) that should be added for completeness. In order to allow us comparing this result with the case where we study the security of a complete circuit encoded with the ISW compiler, we also write our result according to the following corollary (which is less general than Theorem 10.11).

Corollary 10.12. *Let d be the number of shares used for a key encoding and m the number of measurements. Then, if we refresh the encoding in leak-free manner between each measurement and for any $\alpha > 0$, the probability of success of a key recovery t-MI-adversary under independent leakage is:*

$$\mathsf{SR}^{\mathsf{kr}} \leq m \cdot \exp\left(-\alpha d\right), \tag{10.7}$$

if we have:

$$t \leq 2\left(\frac{1}{e^{\alpha}|F|}\right)^{2} . \tag{10.8}$$

Proof. We have:

$$1 - \left(1 - \epsilon^{d}\right)^{m} \leq m e^{\log(\epsilon) d} .$$

We want $\log(\epsilon) = -\alpha$. Hence, from Theorem 10.11, we get our result. □

10.3.3 Security of the Whole Circuit

In this section, we restate the theorems from Chapter 9 when securing a whole circuit using the seminal ISW compiler in a more comprehensive way. Theorem 9.17 bounds the probability of success of a distinguishing adversary in the noisy leakage model. We provide an alternative version of this theorem and, as in the previous section, we relate it to the mutual information instead of the statistical distance.

Theorem 10.13. *Suppose that we have a circuit of size $|\Gamma|$ protected with the ISW compiler with d shares. Then, the probability of success of a distinguishing t-MI-adversary under independent leakage is:*

$$\mathsf{SR}^{\mathsf{dist}} \leq |\Gamma| \cdot \exp\left(-\frac{d}{12}\right) = |\Gamma| \cdot 2^{\left(-\frac{d \cdot \log_2(e)}{12}\right)} \leq |\Gamma| \cdot 2^{-d/9}, \tag{10.9}$$

if we have:

$$t \leq 2 \cdot \left(\frac{1}{|F| \cdot (28d + 16)}\right)^2 . \tag{10.10}$$

Similarly to what we did in the previous section, we also write this corollary.

Corollary 10.14. *Suppose that we have a circuit of size $|\Gamma|$ protected with the ISW compiler with d shares. Then, a distinguisher t-MI-adversary under independent leakage needs:*

$$d \geq \frac{1 - 16|F|\sqrt{\frac{1}{2}t}}{28|F|\sqrt{\frac{1}{2}t}} \tag{10.11}$$

shares in order to obtain:

$$\mathsf{SR}^{\mathsf{dist}} \leq |\Gamma| \cdot \exp\left(-\frac{d}{12}\right) \leq |\Gamma| \cdot \exp\left(-\frac{1 - 16|F|\sqrt{\frac{1}{2}t}}{336|F|\sqrt{\frac{1}{2}t}}\right) . \tag{10.12}$$

Note that the ISW compiler can actually be used to efficiently compute any circuit. For example, the work of Rivain and Prouff at CHES 2010 showed how to adapt the compiler to $|F| = 256$ which leads to efficient masked implementations of the AES [RP10] (see also various following works such as [CGP+12a, CPRR13, GPS14, RP12]).

10.4 Experimental Validation

In this section, we complement the previous theoretical results with an experimental analysis and provide and empirical evaluation of the encoding scheme in Section 10.3.2,

which allows us to discuss the noise condition and tightness of the bounds in our proofs. We use this discussion to conjecture a simple connection between the mutual information metric and the success rate of a (worst-case) side-channel adversary, and argue that it can lead to quite accurate approximations of the attacks' measurement complexity in practical scenarios.
In order to discuss the relevance of the proofs in the previous section, we take the (usual) context of standard DPA attacks defined in [MOS11]. More precisely, we are going to consider the following *practically meaningful setup* throughout this section in which the adversary targets a single S-box S from a block cipher (e.g., the AES).

Definition 10.15 (Simple DPA attack setup)**.** *Let S be an $|\mathbb{F}| \times |\mathbb{F}|$ S-box. We let $x \in \mathbb{F}$ be a plaintext, $k \in \mathbb{F}$ be a key and we let $y \coloneqq S(x + k), y \in \mathbb{F}$ be the output of the S-box. We consider now the masked version of this scheme. We let $y_1, \ldots, y_d$ be the shares of the output, i.e., $y \coloneqq y_1 \oplus \cdots \oplus y_d$, let $x_1, \ldots, x_d$ be the shares of the input, and let $k_1, \ldots, k_d$ be the shares of the key. We suppose in the following that the adversary knows x and sees noisy leakage from $y_1, \ldots, y_d$ which is denoted by $\mathsf{L}_{\mathsf{Noisy}}(y_1), \ldots, \mathsf{L}_{\mathsf{Noisy}}(y_d) \in \mathcal{L}$.*

For convenience, we will mainly consider the context of mathematically-generated Gaussian Hamming weight leakages, where $\mathsf{L}_{\mathsf{Noisy}}(y_i) = \mathsf{Hw}(y_i) + N_i$, with Hw the Hamming weight function and N_i a Gaussian-distributed noise, with variance σ^2. In this respect, we note that we did not mount concrete attacks because we would have had to measure hundreds of different implementations to observe useful trends in practice. Our experiments indeed correspond to hundreds of different noise levels. Yet, we note that devices that exhibit close to Hamming weight leakages are frequently encountered in practice [MOP07]. Furthermore, such a simulated setting is a well established tool to analyze masking schemes (see, e.g. [CPRR13] for polynomial masking and [BFGV12] for inner product masking). Besides, we also consider random Gaussian leakage functions, of which the deterministic part corresponds to random functions over $\mathcal{Y}$, to confirm that all the trends we put forward are also observed with leakage functions that radically differ from the usual Hamming weight one.

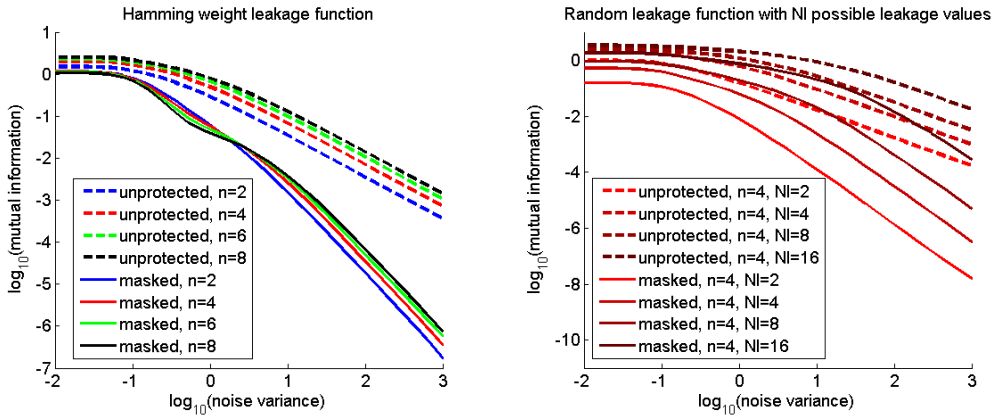
10.4.1 Intuition Behind the Noise Condition

Theorems 10.11 and 10.13 both require that the MI between the shares and their corresponding leakage is sufficiently small. In other words, they require the noise to be sufficiently large. In this section, we compute the MI metric for both an unprotected implementation (i.e., $d = 1$) and a masked one (i.e., $d = 2$) in function of different parameters. In order to illustrate the computation of this metric, we provide a simple open source code that evaluates the MI between a sensitive variable Y and its Hamming weights, for different noise levels, both via numerical integration (that is only possible for mathematically-generated leakages) and sampling (that is more reflective of the evaluation of an actual device).[3] In the latter case, an evaluator additionally has to make

[3] Code available on `http://perso.uclouvain.be/fstandae/PUBLIS/154.zip`.

sure that his estimations are accurate enough. Tools for ensuring this condition are discussed in [DSV14].

We start with the simplest possible plot, where the MI metric is computed in function of the noise variance σ^2. Figure 10.1 shows these quantities, both for Hamming weight leakage functions and for random ones with output range N_l (in the latter context, the functions for different N_l's were randomly picked up prior to the experiments, and stable across experiments). We also considered different secret sizes ($n := |\mathbb{F}| = 2, 4, 6, 8$). Positively, we see that in all cases, the curves reach a linear behavior, where the slope corresponds to the number of shares d. As the independent leakage condition is fulfilled in these experiments, this d corresponds to the smallest key-dependent moment in the leakage distribution. As the measurement (aka sampling) cost for estimating such moments is proportional to $(\sigma^2)^d$, we observe that the MI decreases exponentially in d for large enough noises. Note that this behavior is plotted for $d = 1, 2$, but was experimented for d's up to 4 in [SVCO+10], and in fact holds for any d, as it exactly corresponds to Theorem 10.11 in a context where its assumptions are fulfilled.



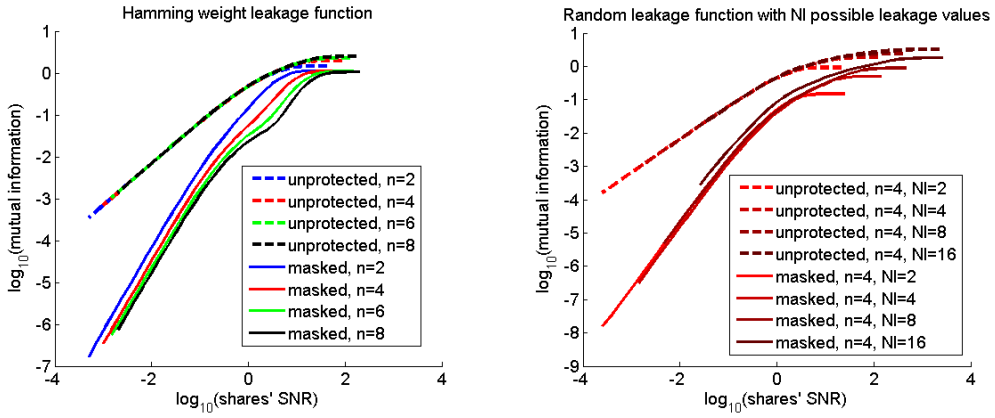
(a) using Hw leakage.

(b) using random leakage.

Figure 10.1 – MI metric in function of σ^2.

Negatively, we also see that the noise level that can be considered as high enough depends on the leakage functions. For example, the random leakage functions in the right part of the figure have signals that vary from approximately $\frac{2}{4}$ for $N_l = 2$ to $\frac{16}{4}$ for $N_l = 16$. It implies that the linearly decreasing part of the curves is reached for larger noises in the latter case. Yet, this observation in fact nicely captures the intuition behind the noise condition. That is, the noise should be high enough for hiding the signal. Therefore, a very convenient way to express it is to plot the MI metric in function of shares' signal to noise ratio (SNR), as in Figure 10.2. Here, we clearly see that as soon as the SNR is below a certain constant (10^{-1}, typically), the shape of the MI curves gets close to linear. This corroborates the condition in Theorem 10.11 that masking requires $\mathrm{MI}(Y_i; \mathsf{L}_{\mathsf{Noisy}}(Y_i))$ to be smaller than a given constant. Our experiments with different bit sizes also suggest that the $|\mathbb{F}|$ factor in this noise condition is a proof artifact. A step in this direction

was recently shown by Dziembowski, Faust and Skorski in [DFS15b]. In their work, they managed to show the equivalence of the noisy-leakage model to a new model: the *average probing model* in which they were able to construct a compiler (using leak-free gates). In their reductions, there is no $|\mathbb{F}|$ factor. We discuss more about their model in the conclusion of this thesis.



(a) using Hw leakage.

(b) using random leakage.

Figure 10.2 – MI metric in function of the shares' SNR.

10.4.2 Tightness of the Bounds

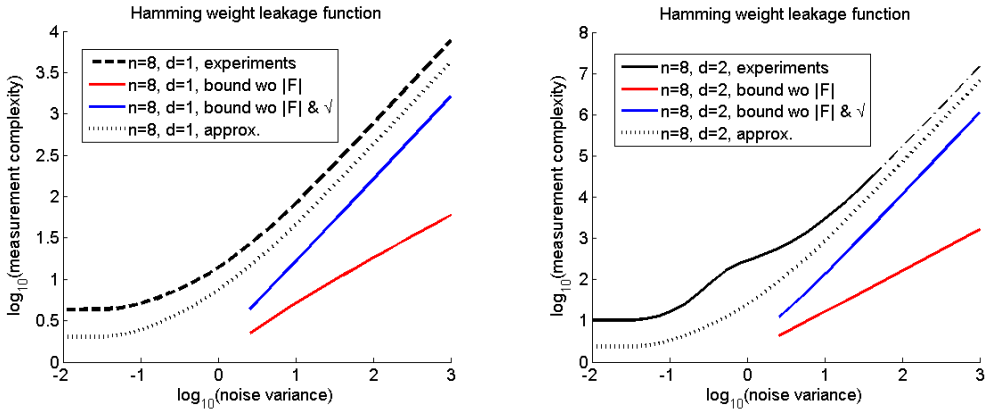
Given that the noise is high enough (as just discussed), Theorems 10.11 and 10.13 guarantee that the success rate of a side-channel adversary can be bounded based on the value of the share's leakage, measured with $\mathrm{MI}(Y_i; \mathsf{L}_{\mathsf{Noisy}}(Y_i))$. This directly leads to useful bounds on the measurement complexity to reach a given success rate, e.g. from (10.5) we can compute:

$$m \geq \frac{\log(1 - \mathsf{SR}^{\mathsf{kr}})}{\log\left(1 - \left(|\mathbb{F}|\sqrt{\frac{\mathrm{MI}(Y_i;\mathsf{L}_{\mathsf{Noisy}}(Y_i))}{2}}\right)^d\right)} \,. \tag{10.13}$$

We now want to investigate how tight this bound is. For this purpose, we compared it with the measurement complexity of concrete key recovery TA (using a perfect leakage model).[4] As previously mentioned, the $|\mathbb{F}|$ factor in this equation can be seen as a proof artifact related to the reduction in our theorems, so we tested a bound excluding this factor. For similar reasons, we also tested a bound additionally excluding the square root loss in the reductions (coming from Theorem 10.9). We believe that for meaningful

[4] Our attacks exploit the leakages of an S-box output. We took the PRESENT S-box for $n = 4$, the AES one for $n = 8$, and picked up two random S-boxes for $n = 2, 6$, as we did for the random leakage functions.

noises, this square root can be removed. As illustrated in Figure 10.3, the measurement complexity of the attacks is indeed bounded by Equation (10.13), and removing the square root loss allows the experimental and theoretical curves to have similar slopes. The latter observation fits with the upper bound $\mathsf{MI}(Y_i; \mathsf{L}_{\mathsf{Noisy}}(Y_i)) \leq \frac{|\mathbb{F}|}{\ln(2)} \cdot \Delta(Y_i; Y_i \mid \mathsf{L}_{\mathsf{Noisy}}(Y_i))$ given in [PR13] that becomes tight as the noise increases. As expected, the bounds become meaningless for too low noise levels. Intuitively, this is because we reach success rates that are stuck to one when we deviate from this condition. For completeness, we added approximations obtained by normalizing the shares' MI by $\mathrm{H}[K]$ to the figure, which provide hints about the behavior of a leaking device when the noise is too low.



(a) $d = 1$. (b) $d = 2$.

Figure 10.3 – Measurement complexity and bounds/approximations for concrete TA.

Interestingly, these results also allow us to reach a comprehensive view of the parameters in Theorem 10.13, where the security of a complete circuit encoded according to the ISW compiler is proven. That is, in this case as well, we expect the $|\mathbb{F}|$ and $1/9$ factors in (10.9) to be due to proof technicalities. By contrast, the $|\Gamma|$ factor is physically motivated, as it corresponds to the size of the circuit and fits the intuition that more computations inevitably means more exploitable leakage. The d factor appearing in the noise condition of (10.10) can also be explained, as it directly relates to the fact that in the ISW compiler, any multiplication will require to manipulate each share d times. It typically reflects the distance between standard (divide-and-conquer) side-channel attacks (such as analyzed in this section) and more powerful (multivariate) adversaries trying the exploit the leakage of all the intermediate computations in a block cipher, e.g. based on algebraic cryptanalysis (see [RS09a, RSV09] and follow up works).
In summary, we informally conjecture the following bound that seem to hold for practical scenarios:[5]

[5]We know that there are some twisted constructions for which this conjecture does not hold (the MI/SD inequality is tight).

Conjecture (informal)**.** *Suppose that we have a circuit of size* $|\Gamma|$ *masked with* d *shares such that the information leakage on each of these shares (using all available time samples) is bounded by* $\mathsf{MI}(Y_i; \mathsf{L}_{\mathsf{Noisy}}(Y_i))$*. Then, the probability of success of a distinguishing adversary using* m *measurements and targeting a single element (e.g. gate) of the circuit under independent and sufficiently noisy leakage is:*

$$\mathsf{SR}_1^{\mathsf{dist}} \leq 1 - \left(1 - \mathsf{MI}(Y_i; \mathsf{L}_{\mathsf{Noisy}}(Y_i))^d\right)^m , \tag{10.14}$$

and the probability of success targeting all $|\Gamma|$ *elements independently equals:*

$$\mathsf{SR}_{|\Gamma|}^{\mathsf{dist}} \leq 1 - (1 - \mathsf{SR}_1^{\mathsf{dist}})^{|\Gamma|} . \tag{10.15}$$

Interestingly, Equation (10.15) (like Theorem 10.13) assumes that the leakages of the $|\Gamma|$ gates (or target intermediate values) are exploited independently. This perfectly corresponds to the probing model in which the adversary gains either full knowledge or no knowledge of such computing elements. Thanks to what was presented in Chapter 9, it also implies a similar result against noisy leakages if the noise condition is fulfilled. However, as the noise level decreases, some advanced (e.g. algebraic) side-channel attacks can sometimes take advantage of different computations jointly in a more efficient manner. Note that this informal conjecture is backed up by the results in [BJV04, Theorem 6], where a similar bound is given in the context of statistical cryptanalysis. By using the approximation $\log(1 - x) \approx -x$ that holds for x's close to 0, Equation (10.14) directly leads to the following simple approximation of a standard DPA's measurement complexity for large noise levels:

$$m \geq \frac{\log(1 - \mathsf{SR}_1^{\mathsf{dist}})}{\log(1 - \mathsf{MI}(Y_i; \mathsf{L}_{\mathsf{Noisy}}(Y_i))^d)} \approx \frac{c}{\mathsf{MI}(Y_i; \mathsf{L}_{\mathsf{Noisy}}(Y_i))^d} , \tag{10.16}$$

where c is a small constant that depends on the target success rate.

Conclusion and Further Work

In this thesis, we tried to bring practice closer to theory. We did that by

- proposing concrete instances of our HELEN cryptosystem,
- giving the exact complexity of our algorithms solving hard problems,
- unifying a theoretical model with a more realistic model, and
- showing how this latest result could be expressed in a more practically-meaningful way.

Below, we conclude on each result and discuss further work.

HELEN

In Chapter 5, we presented HELEN which is a code-based public-key cryptosystem based on the hardness of some well-known problems. As its margin of progression is still large, HELEN can become a competitive cryptosystem with truly practical parameters. We take advantage of this thesis to emphasise the importance of giving practical instances for implementers when proposing a new cryptosystem.

Further Work. HELEN can be extended in multiple ways. A first idea is to use different $\mathcal{H}$ to reduce the probability of error and, hence, to reduce the transmission overhead. This implies also to verify that Assumption 5.4 holds for this new $\mathcal{H}$. Another idea would be to try to link HELEN with the Ring version of the LPN problem: Ring-LPN [HKL$^+$11, HKL$^+$12, BL12]. Indeed, results have shown that using the Ring counterpart of LPN (or LWE) often greatly improve the public parameters of a cryptosystem (e.g., [LPR13, SS11, BV11b, BGV12, GHS12b, GHS12a, MP12, GHPS12, LPR13]). The codes C_1 and C_2 described in Section 5.2 would need to be modified accordingly as well as the noise we add so that the match the specifications of the Ring-LPN problem. A first step in that direction was made in [Cho15] which shows that one should first study the Ring-LPN problem better. We believe that switching to Ring-LPN might make HELEN truly practical.

LWE and LWR

In Chapter 6 and 7, we introduced algorithms which were, when published, the best algorithms for solving the LWE and the LWR problem. Our algorithms use Fourier transforms and we provided a careful analysis of the rounded Gaussian distribution which can be of independent interest. In particular, we studied its variance and the expected value of its cosine. We obtained our LWR algorithm by applying the LWE algorithm to an LWR instance when q is prime. This algorithm is the first (and currently only) concrete LWR-solving algorithm.

Further Work. Further work includes the study of the Ring variants of LWE and LWR [LPR10, BPR12]. A step in that direction was very recently done in [ELOS15] in which Elias et al. study some weak instances of Ring-LWE. A useful further step in that direction would be, thus, to show how to exploit the ring structure of Ring-LWE to obtain improvements for any Ring-LWE instance.
Another interesting area would be to study variants of LWE, e.g., when the secret follows a non-uniform distribution (like in [AFFP14]) or when the noise follows a non Gaussian distribution. In particular, instances in which the noise follows a small uniform distribution [MP13, DM13] seem to be particularly interesting when trying to implement it on constrained hardware. A recent paper is also strictly improving our practical values for LWE [GJS15]. In that paper, Guo et al. adapt their idea of covering codes used for LPN [GJL14] to LWE. For this, they replaced quite naturally covering codes by lattice codes. Another recent result that will be presented at CRYPTO'15 is even obtaining an asymptotic improvement with respect to our work [KF15].
Finally, we want to mention a survey by Albrecht et al. that studies all the existing algorithms solving the LWE problem [APS15]. In this paper, the authors look both at lattice reduction techniques and at BKW-based algorithms and compare their performances in few meaningful case-studies, e.g., fully homomorphic encryption. In our point of view, this kind of survey is an essential step for the construction of any practical scheme based on LWE and should be encouraged.
Regarding LWR, we want to motivate the cryptography community to study more this interesting problem that avoids the tedious usage of randomness and has many great applications due to its deterministic nature. A first step would be to adapt our algorithm for non-prime q's. However, we believe that the biggest improvement might come from an algorithm that will exploit the deterministic nature of LWR.

Leakage Models

In Chapter 9, we showed how to link the practically meaningful noisy leakage model to the theoretically extremely convenient threshold-probing model. In particular, we show the security of the ISW construction [ISW03] in the noisy leakage setting and, hence, removing most of the shortcomings from the Eurocrypt 13 work of Prouff and

Rivain [PR13].
Later, in Chapter 10, we show how one can interpret our results in a different way to help the task of evaluating the security of a masked implementation. In particular, we link our result with the mutual information metric, a metric which is often used in concrete physical security evaluations.

Further Work. The leakage-resilient cryptography area is broad and further work can extend in many diverse directions. One interesting direction is taken by Dziembowski, Faust, and Skorski in [DFS15b] and the introduction of a new probing model shown to be equivalent to the noisy leakage model: the average probing model. Recall the definition of ϵ-identity function in Definition 8.2. In this definition, the probability to obtain $\perp$ was *identical for every input*. Dziembowski et al. generalized this definition into *ϵ-average-identity functions*. In the following definition, we write explicitly the randomness to show clearly the difference between the two definitions.

Definition 10.16 (ϵ-average-identity function [DFS15b])**.** *A randomized function $\varphi : \mathcal{X} \times \mathcal{R} \to \mathcal{X} \cup \{\perp\}$ is an* ϵ-average-identity function *if* $\Pr_{\substack{x \leftarrow \mathcal{X} \\ r \leftarrow \mathcal{R}}}[\phi(x, r) \neq \perp] = \epsilon$.

Dziembowski et al. then defined an adversary in this model which looks very similar to the adversary defined in Definition 8.3 except that we use ϵ-average-identity functions. The major difference with ϵ-identity functions is that the probability distribution of $\phi(x)$ depends now on the input x as well.

Definition 10.17 (ϵ-average-probing adversary [DFS15b])**.** *For $\epsilon \geq 0$ an* ϵ-average-probing adversary on $\mathcal{X}^\ell$ *is an algorithm $\mathcal{A}$ that plays the following scenario against an oracle that knows $(x_1, \ldots, x_\ell) \in \mathcal{X}^\ell$:*

1. *$\mathcal{A}$ specifies a sequence $(\epsilon_1, \ldots, \epsilon_\ell)$ such that each $\epsilon_i \leq \epsilon$.*
2. *$\mathcal{A}$ receives $(\varphi_1(x_1, r_1), r_1), \ldots, (\varphi_\ell(x_\ell, r_\ell), r_\ell)$ and outputs some value denoted by $\mathsf{out}_{\mathcal{A}}(x_1, \ldots, x_\ell)$, where each φ_i is the ϵ_i-average-identity function with mutually independent randomness r_i.*

The advantage of using this model with respect to the threshold probing model used in Chapter 9 is that the $|\mathbb{F}|$ factor in the reduction disappears. While they could show the security of the ISW compiler using leak free gates, a further step would be to design a compiler secure in this model even without these gates. It would be also insightful to find a direct link between the average probing model and the threshold probing model to understand their relation more.
Another direction that should be studied in more depth is when dealing with non-independent leakage. Indeed, some concrete experiments have shown that this assumption is frequently not met in practice (e.g., [BGG+14, CGP+12b, MPG05, RSV+11]). While we present in [DFS15a] some experiments and some heuristics showing how the security evolves when independence is not met, a more formal approach is needed.

Finally, we started in [DFS15a] to involve *computational power* in our analysis. The idea comes from the following observation showing that mutual information is not enough. Imagine two hypothetical side-channel attacks that both succeed with probability 1/100. In the first case, the adversary gains nothing with probability 99/100 and the full key with probability 1/100. In the second case, he always gains a set of 100 equally likely keys. Clearly, enumeration will be pretty useless in the first case, while extremely powerful in the second one. More generally, such examples essentially suggest that the computational cost of an enumeration does not only depend on the informativeness of the leakage function (e.g. measured with the MI) but also on its shape. Incidentally, this example also puts forward some limitations of the probing leakage model when measuring computational cost, as it describes an all-or-nothing strategy which is not the case for the noisy leakage setting. Hence, whereas the probing model is easier to manipulate in proofs, and therefore useful to obtain asymptotic results, noisy leakages are a more accurate tool to quantify concrete security levels as in this section.
The idea we introduced in [DFS15a] is, given a limit on the computational power c, to aggregate the c most likely keys together to obtain a new random variable. With this process, the adversary will only learn from the measurements in which class of keys the real key lies. He will then have to bruteforce all the c keys in the class and, hence, will use his computational power. By developing this idea, one should be able to characterize the success rate of an adversary with limited computational power. Further work will, thus, consist in formalizing completely the introduction of this computational capability. Once this success rate is found for a single S-box, we show in [DFS15a] that we can obtain bounds on the success rate of an adversary which is computationally limited and who is targeting more than one S-box. We showed that the problem is known as a "separable, non-linear integer programming problem". We give, then, an algorithm that heuristically solves it using a downsampling and merging algorithm which can be seen as an generalization of a greedy algorithm that considers only parts of the solutions which are locally close to optimal. However, our bound is rather loose in some extreme cases and based on a weak Maximum Likelihood approach [YEM14, Sta15], i.e., a non-optimal solution. Hence, one possible direction for further work would be to improve this estimate when targeting more than one S-box and find a more suitable algorithm.

Final Words

To conclude this thesis, we want to emphasize the need for theory and practice to work together. In particular, regarding new cryptosystems, we believe that designers should *always* put forward concrete parameters so that implementers can then build an instance. The same holds when showing some clever improvement of algorithms solving a hard problem. As we base our concrete security on the best attack against a problem, we need some way of computing the exact complexity of an algorithm to be able to deduce parameters that are believed secure for a certain security parameter.
Next, when proving security in a convenient model, it might be important to keep in

mind whether this model is practically meaningful or not. In the negative, one should try to link it to more physically-understood models. On the other hand, we would love to see the practical community have a look at these theoretical models and try to interpret them concretely. However, we believe that the best results can be obtained when both communities collaborate strongly together.

List of Algorithms

4.1 Oracle $\mathcal{A}_{\boldsymbol{s},\chi,\ell}$, for $0 < \ell < a$ 27
5.1 The HELEN encryption algorithm 33
5.2 HELEN decryption algorithm 33
5.3 HELEN key generation algorithm 34
6.1 Hypothesis testing algorithm for LWE. 51

List of Definitions

2.1 Negligible Function . . . 7
2.2 Sequence of Bernoulli trials (S_p^n) . . . 8
2.3 Statistical distance . . . 8
2.6 Public-key Encryption Scheme . . . 9
2.7 IND-CPA-security . . . 10
2.8 Simple real-or-random IND-CPA game security . . . 10
2.9 IND-CCA-security . . . 10
2.10 Distinguisher . . . 11
2.11 Discrete Fourier Transform (DFT) . . . 11
2.12 Continuous Fourier transform (FT) . . . 12

3.1 LPN Oracle . . . 17
3.2 Search Learning from Parity with Noise Problem (LPN) . . . 18
3.3 Decisional LPN Problem (D-LPN) . . . 18
3.5 Universal family of hash functions . . . 19
3.8 Minimum Distance Problem (MDP) . . . 20
3.10 LWE Oracle . . . 21
3.11 Search-LWE . . . 21
3.12 Rounded Gaussian distribution ($\bar{\Psi}_{\sigma,q}$) . . . 22
3.13 Discrete Gaussian distribution ($D_{\sigma,q}$) . . . 22
3.14 Rounding function ($\lceil\cdot\rfloor_p$) . . . 23
3.15 LWR Oracle . . . 23
3.16 Search-LWR . . . 23

5.8 Γ-uniformity . . . 40

6.4 $R_{\sigma,q,\chi}$ . . . 48

8.1 t-threshold-probing adversary . . . 70
8.2 ϵ-identity function . . . 70
8.3 ϵ-random-probing adversary . . . 71

8.4 δ-noisy function . . . 72
8.5 Euclidean Norm . . . 73
8.6 Efficiently decidable noise . . . 74
8.7 δ-noisy adversary . . . 74

9.5 Stateful arithmetic circuit . . . 88
9.6 Black-box circuit adversary . . . 89
9.7 δ-noisy circuit adversary . . . 89
9.8 ϵ-random probing circuit adversary . . . 89
9.9 t-threshold probing circuit adversary . . . 90
9.10 Implementation of a circuit . . . 90
9.11 (δ, ξ)-noise resilient implementation of a circuit . . . 90
9.12 (ϵ, ξ)-random probing resilient implementation of a circuit . . . 90
9.13 (t, ξ)-threshold probing resilient implementation of a circuit . . . 91
9.14 Implementation of a circuit with efficient simulation . . . 91
9.15 Additive masking (Enc_+) . . . 91
9.18 δ-gate-noisy circuit adversary . . . 95
9.19 (δ, ξ)-gate-noise resilient implementation of a circuit . . . 96

10.1 Entropy . . . 101
10.2 Conditional Entropy . . . 101
10.3 Mutual Information . . . 101
10.4 t-MI-advesary . . . 102
10.5 Signal-to-Noise Ratio (SNR) . . . 102
10.6 Success rate for key recovery ($\mathsf{SR}^{\mathsf{kr}}$) . . . 103
10.7 Success rate for distinguishing ($\mathsf{SR}^{\mathsf{dist}}$) . . . 103
10.15 Simple DPA attack setup . . . 108
10.16 ϵ-average-identity function [DFS15b] . . . 115
10.17 ϵ-average-probing adversary [DFS15b] . . . 115

Bibliography

[ABW10] Benny Applebaum, Boaz Barak, and Avi Wigderson. Public-key cryptography from different assumptions. In Schulman [Sch10], pages 171–180. *Cited on page:* 31.

[ACF+13] Martin R. Albrecht, Carlos Cid, Jean-Charles Faugère, Robert Fitzpatrick, and Ludovic Perret. On the complexity of the BKW algorithm on LWE. *Designs, Codes and Cryptography*, pages 1–30, 2013. *Cited on pages:* 3, 22, 25, 27, 43, 44, 45, 50, 51, 54, and 55.

[ACF+14] Martin R. Albrecht, Carlos Cid, Jean-Charles Faugère, Robert Fitzpatrick, and Ludovic Perret. Algebraic Algorithms for LWE Problems. *IACR Cryptology ePrint Archive*, 2014:1018, 2014. *Cited on page:* 43.

[ACPS09] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 595–618. Springer, 2009. *Cited on page:* 21.

[AFFP14] Martin R. Albrecht, Jean-Charles Faugère, Robert Fitzpatrick, and Ludovic Perret. Lazy Modulus Switching for the BKW Algorithm on LWE. In Hugo Krawczyk, editor, *Public Key Cryptography*, volume 8383 of *Lecture Notes in Computer Science*, pages 429–445. Springer, 2014. *Cited on pages:* 43 and 114.

[AFMV07] Jean-Philippe Aumasson, Matthieu Finiasz, Willi Meier, and Serge Vaudenay. TCHo: A Hardware-Oriented Trapdoor Cipher. In Josef Pieprzyk, Hossein Ghodosi, and Ed Dawson, editors, *ACISP*, volume 4586 of *Lecture Notes in Computer Science*, pages 184–199. Springer, 2007. *Cited on page:* 30.

[AG11] Sanjeev Arora and Rong Ge. New Algorithms for Learning in Presence of Errors. In Luca Aceto, Monika Henzinger, and Jiri Sgall, editors, *Automata,*

Languages and Programming - 38th International Colloquium, ICALP 2011, Zurich, Switzerland, July 4-8, 2011, Proceedings, Part I, volume 6755 of *Lecture Notes in Computer Science*, pages 403–415. Springer, 2011. *Cited on page:* 43.

[AGV09] Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. Simultaneous Hardcore Bits and Cryptography against Memory Attacks. In Reingold [Rei09], pages 474–495. *Cited on pages:* 67 and 81.

[Ajt11] Miklós Ajtai. Secure computation with information leaking to an adversary. In *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 715–724, 2011. *Cited on page:* 71.

[AKPW13] Joël Alwen, Stephan Krenn, Krzysztof Pietrzak, and Daniel Wichs. Learning with rounding, revisited - new reduction, properties and applications. In Canetti and Garay [CG13], pages 57–74. *Cited on pages:* 23, 24, 63, and 64.

[Ale03] Michael Alekhnovich. More on Average Case vs Approximation Complexity. In *FOCS*, pages 298–307. IEEE Computer Society, 2003. *Cited on pages:* 3, 29, and 30.

[AP13] Graeme D. Ruxton Arthur Pewsey, Markus Neuhäuser. *Circular statistics in R.* Oxford University Press, 2013. *Cited on page:* 49.

[APS15] Martin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of Learning with Errors. *IACR Cryptology ePrint Archive*, 2015:46, 2015. *Cited on page:* 114.

[Ban93] Wojciech Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(1):625–635, 1993. *Cited on page:* 48.

[BB86] Ronald Newbold Bracewell and RN Bracewell. *The Fourier transform and its applications*, volume 31999. McGraw-Hill New York, 1986. *Cited on page:* 13.

[BCD06] Julien Bringer, Hervé Chabanne, and Emmanuelle Dottax. HB^{++}: a Lightweight Authentication Protocol Secure against Some Attacks. In *SecPerU*, pages 28–33. IEEE Computer Society, 2006. *Cited on page:* 29.

[BDJR97a] Mihir Bellare, Anand Desai, E. Jokipii, and Phillip Rogaway. A Concrete Security Treatment of Symmetric Encryption: Analysis of the DES Modes of Operation (Full Version), 1997. Available at `http://cseweb.ucsd.edu/users/mihir`. *Cited on page:* 10.

[BDJR97b] Mihir Bellare, Anand Desai, E. Jokipii, and Phillip Rogaway. A Concrete Security Treatment of Symmetric Encryption (Extended Abstract). In *FOCS*, pages 394–403, 1997. *Cited on pages:* 10 and 36.

[Ber09] Daniel J. Bernstein. Introduction to post-quantum cryptography. In Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen, editors, *Post-Quantum Cryptography*, pages 1–14. Springer, 2009. *Cited on page:* 30.

[BFG15] Josep Balasch, Sebastian Faust, and Benedikt Gierlichs. Inner Product Masking Revisited. In Oswald and Fischlin [OF15], pages 486–510. *Cited on page:* 69.

[BFGV12] Josep Balasch, Sebastian Faust, Benedikt Gierlichs, and Ingrid Verbauwhede. Theory and Practice of a Leakage Resilient Masking Scheme. In Wang and Sako [WS12], pages 758–775. *Cited on pages:* 99 and 108.

[BFKL93] Avrim Blum, Merrick L. Furst, Michael J. Kearns, and Richard J. Lipton. Cryptographic Primitives Based on Hard Learning Problems. In Douglas R. Stinson, editor, *CRYPTO*, volume 773 of *Lecture Notes in Computer Science*, pages 278–291. Springer, 1993. *Cited on pages:* 17 and 30.

[BGG+14] Josep Balasch, Benedikt Gierlichs, Vincent Grosso, Oscar Reparaz, and François-Xavier Standaert. On the Cost of Lazy Engineering for Masked Software Implementations. In Joye and Moradi [JM15], pages 64–81. *Cited on pages:* 100 and 115.

[BGJ14] Anja Becker, Nicolas Gama, and Antoine Joux. A sieve algorithm based on overlattices. *LMS Journal of Computation and Mathematics*, 17:49–70, 1 2014. *Cited on page:* 43.

[BGK04] Johannes Blömer, Jorge Guajardo, and Volker Krummel. Provably Secure Masking of AES. In *Selected Areas in Cryptography*, pages 69–83, 2004. *Cited on pages:* 68 and 80.

[BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In Shafi Goldwasser, editor, *Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012*, pages 309–325. ACM, 2012. *Cited on page:* 113.

[BJMM12] Anja Becker, Antoine Joux, Alexander May, and Alexander Meurer. Decoding Random Binary Linear Codes in $2^{n/20}$: How $1 + 1 = 0$ Improves Information Set Decoding. In Pointcheval and Johansson [PJ12], pages 520–536. *Cited on page:* 20.

[BJV04] Thomas Baignères, Pascal Junod, and Serge Vaudenay. How Far Can We Go Beyond Linear Cryptanalysis? In Pil Joong Lee, editor, *Advances in*

Cryptology - ASIACRYPT 2004, 10th International Conference on the Theory and Application of Cryptology and Information Security, Jeju Island, Korea, December 5-9, 2004, Proceedings, volume 3329 of *Lecture Notes in Computer Science*, pages 432–450. Springer, 2004. *Cited on pages:* 99 and 112.

[BKW03] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, 2003. *Cited on pages:* 3, 18, 25, 26, 27, and 43.

[BL12] Daniel J. Bernstein and Tanja Lange. Never Trust a Bunny. In Jaap-Henk Hoepman and Ingrid Verbauwhede, editors, *RFIDSec*, volume 7739 of *Lecture Notes in Computer Science*, pages 137–148. Springer, 2012. *Cited on pages:* 18 and 113.

[BLMR13] Dan Boneh, Kevin Lewi, Hart William Montgomery, and Ananth Raghunathan. Key Homomorphic PRFs and Their Applications. In Canetti and Garay [CG13], pages 410–428. *Cited on page:* 23.

[BLP08] Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Attacking and Defending the McEliece Cryptosystem. In Johannes Buchmann and Jintai Ding, editors, *PQCrypto*, volume 5299 of *Lecture Notes in Computer Science*, pages 31–46. Springer, 2008. *Cited on page:* 38.

[BLP11] Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Smaller Decoding Exponents: Ball-Collision Decoding. In Rogaway [Rog11], pages 743–760. *Cited on pages:* 20 and 21.

[BLP+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In Boneh et al. [BRF13], pages 575–584. *Cited on pages:* 21, 22, and 43.

[BPR12] Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom Functions and Lattices. In Pointcheval and Johansson [PJ12], pages 719–737. *Cited on pages:* 3, 23, and 114.

[Bra12] Zvika Brakerski. Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP. In Safavi-Naini and Canetti [SC12], pages 868–886. *Cited on pages:* 3 and 21.

[BRF13] Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors. *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*. ACM, 2013.

[BSS00] Joe Buhler, Mohammad Amin Shokrollahi, and Volker Stemann. Fast and precise Fourier transforms. *IEEE Transactions on Information Theory*, 46(1):213–228, 2000. *Cited on page:* 50.

[BTV12] Mihir Bellare, Stefano Tessaro, and Alexander Vardy. Semantic Security for the Wiretap Channel. In Safavi-Naini and Canetti [SC12], pages 294–311. *Cited on pages:* 104 and 105.

[BTV15] Sonia Bogos, Florian Tramèr, and Serge Vaudenay. On Solving LPN using BKW and Variants -Implementation and Analysis-. *To appear in Cryptography and Communications, Discrete Structures, Boolean Functions and Sequences*, 2015. *Cited on pages:* 18, 19, and 49.

[BV11a] Zvika Brakerski and Vinod Vaikuntanathan. Efficient Fully Homomorphic Encryption from (Standard) LWE. In Rafail Ostrovsky, editor, *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 97–106. IEEE, 2011. *Cited on pages:* 3 and 21.

[BV11b] Zvika Brakerski and Vinod Vaikuntanathan. Fully Homomorphic Encryption from Ring-LWE and Security for Key Dependent Messages. In Rogaway [Rog11], pages 505–524. *Cited on page:* 113.

[BV15] Sonia Bogos and Serge Vaudenay. Personal communication, 2015. *Cited on pages:* 49, 54, and 61.

[Can12] Anne Canteaut, editor. *Fast Software Encryption - 19th International Workshop, FSE 2012*, volume 7549 of *Lecture Notes in Computer Science*. Springer, 2012.

[CC94] Anne Canteaut and Hervé Chabanne. A Further Improvement of the Work Factor in an Attempt at Breaking McEliece's Cryptosystem. In Pascale Charpin, editor, *EUROCODE*, 1994. *Cited on page:* 20.

[CC98] Anne Canteaut and Florent Chabaud. A New Algorithm for Finding Minimum-Weight Words in a Linear Code: Application to McEliece's Cryptosystem and to Narrow-Sense BCH Codes of Length 511. *IEEE Transactions on Information Theory*, 44(1):367–378, 1998. *Cited on page:* 20.

[CCD00] Christophe Clavier, Jean-Sébastien Coron, and Nora Dabbous. Differential Power Analysis in the Presence of Hardware Countermeasures. In *CHES*, pages 252–263, 2000. *Cited on page:* 71.

[CDGM14] Claude Carlet, Jean-Luc Danger, Sylvain Guilley, and Houssem Maghrebi. Leakage squeezing: Optimal implementation and security evaluation. *J. Mathematical Cryptology*, 8(3):249–295, 2014. *Cited on page:* 99.

[CG13] Ran Canetti and Juan A. Garay, editors. *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, volume 8042 of *Lecture Notes in Computer Science*. Springer, 2013.

[CGP+12a] Claude Carlet, Louis Goubin, Emmanuel Prouff, Michaël Quisquater, and Matthieu Rivain. Higher-Order Masking Schemes for S-Boxes. In Canteaut [Can12], pages 366–384. *Cited on pages:* 77 and 107.

[CGP+12b] Jean-Sébastien Coron, Christophe Giraud, Emmanuel Prouff, Soline Renner, Matthieu Rivain, and Praveen Kumar Vadnala. Conversion of Security Proofs from One Leakage Model to Another: A New Issue. In Werner Schindler and Sorin A. Huss, editors, *Constructive Side-Channel Analysis and Secure Design - Third International Workshop, COSADE 2012, Darmstadt, Germany, May 3-4, 2012. Proceedings*, volume 7275 of *Lecture Notes in Computer Science*, pages 69–81. Springer, 2012. *Cited on pages:* 100 and 115.

[Cho15] Gwanbae Choi. Extending the HELEN Cryptosystem. EPFL Master Semester Project, 2015. *Cited on pages:* 32, 39, and 113.

[CJRR99] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards Sound Approaches to Counteract Power-Analysis Attacks. In Wiener [Wie99], pages 398–412. *Cited on pages:* 4, 67, and 71.

[CJRT05] Chandra Chekuri, Klaus Jansen, José D. P. Rolim, and Luca Trevisan, editors. *Approximation, Randomization and Combinatorial Optimization, Algorithms and Techniques, 8th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2005 and 9th International Workshop on Randomization and Computation, RANDOM 2005, Berkeley, CA, USA, August 22-24, 2005, Proceedings*, volume 3624 of *Lecture Notes in Computer Science*. Springer, 2005.

[CK10] Jean-Sébastien Coron and Ilya Kizhvatov. Analysis and Improvement of the Random Delay Countermeasure of CHES 2009. In *CHES*, pages 95–109, 2010. *Cited on page:* 71.

[CN11] Yuanmi Chen and Phong Q. Nguyen. BKZ 2.0: Better Lattice Security Estimates. In Lee and Wang [LW11], pages 1–20. *Cited on page:* 43.

[CPR07] Jean-Sébastien Coron, Emmanuel Prouff, and Matthieu Rivain. Side Channel Cryptanalysis of a Higher Order Masking Scheme. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings*, volume 4727 of *Lecture Notes in Computer Science*, pages 28–44. Springer, 2007. *Cited on page:* 99.

[CPRR13] Jean-Sébastien Coron, Emmanuel Prouff, Matthieu Rivain, and Thomas Roche. Higher-Order Side Channel Security and Mask Refreshing. In Shiho Moriai, editor, *Fast Software Encryption - 20th International Workshop, FSE 2013, Singapore, March 11-13, 2013. Revised Selected Papers*, volume

8424 of *Lecture Notes in Computer Science*, pages 410–424. Springer, 2013. *Cited on pages:* 107 and 108.

[CS98] Anne Canteaut and Nicolas Sendrier. Cryptoanalysis of the Original McEliece Cryptosystem. In Kazuo Ohta and Dingyi Pei, editors, *ASIACRYPT*, volume 1514 of *Lecture Notes in Computer Science*, pages 187–199. Springer, 1998. *Cited on page:* 20.

[CT06] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (2. ed.)*. Wiley, 2006. *Cited on pages:* 37 and 103.

[DDF14] Alexandre Duc, Stefan Dziembowski, and Sebastian Faust. Unifying Leakage Models: From Probing Attacks to Noisy Leakage. In Nguyen and Oswald [NO14], pages 423–440. *Cited on pages:* 5 and 77.

[DDN91] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-Malleable Cryptography (Extended Abstract). In Cris Koutsougeras and Jeffrey Scott Vitter, editors, *STOC*, pages 542–552. ACM, 1991. *Cited on page:* 31.

[DDV10] Francesco Davì, Stefan Dziembowski, and Daniele Venturi. Leakage-resilient storage. In Juan A. Garay and Roberto De Prisco, editors, *SCN*, volume 6280 of *Lecture Notes in Computer Science*, pages 121–137. Springer, 2010. *Cited on pages:* 9 and 69.

[DF11] Stefan Dziembowski and Sebastian Faust. Leakage-Resilient Cryptography from the Inner-Product Extractor. In Lee and Wang [LW11], pages 702–721. *Cited on page:* 69.

[DF12] Stefan Dziembowski and Sebastian Faust. Leakage-Resilient Circuits without Computational Assumptions. In *TCC*, pages 230–247, 2012. *Cited on pages:* 71, 78, and 81.

[DFS15a] Alexandre Duc, Sebastian Faust, and François-Xavier Standaert. Making Masking Security Proofs Concrete - Or How to Evaluate the Security of Any Leaking Device. In Oswald and Fischlin [OF15], pages 401–429. *Cited on pages:* 6, 99, 100, 106, 115, and 116.

[DFS15b] Stefan Dziembowski, Sebastian Faust, and Maciej Skorski. Noisy Leakage Revisited. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II*, volume 9057 of *Lecture Notes in Computer Science*, pages 159–188. Springer, 2015. *Cited on pages:* 110, 115, and 122.

[DH76] Whitfield Diffie and Martin Hellman. New directions in cryptography. *Information Theory, IEEE Transactions on*, 22(6):644–654, 1976. *Cited on pages:* 2 and 67.

[DK07] Dang Nguyen Duc and Kwangjo Kim. Securing HB^+ against GRS man-in-the-middle attack. In *Institute of Electronics, Information and Communication Engineers, Symposium on Cryptography and Information Security*, 2007. *Cited on page:* 29.

[DM13] Nico Döttling and Jörn Müller-Quade. Lossy Codes and a New Variant of the Learning-With-Errors Problem. In Johansson and Nguyen [JN13], pages 18–34. *Cited on page:* 114.

[DMQN12] Nico Döttling, Jörn Müller-Quade, and Anderson C. A. Nascimento. IND-CCA Secure Cryptography Based on a Variant of the LPN Problem. In Wang and Sako [WS12], pages 485–503. *Cited on page:* 31.

[Dod12] Yevgeniy Dodis. Shannon Impossibility, Revisited. In Adam Smith, editor, *Information Theoretic Security - 6th International Conference, ICITS 2012, Montreal, QC, Canada, August 15-17, 2012. Proceedings*, volume 7412 of *Lecture Notes in Computer Science*, pages 100–110. Springer, 2012. *Cited on page:* 104.

[DP08] Stefan Dziembowski and Krzysztof Pietrzak. Leakage-Resilient Cryptography. In *FOCS*, pages 293–302, 2008. *Cited on pages:* 67, 75, and 81.

[DP12] Ivan Damgård and Sunoo Park. Is public-key encryption based on LPN practical? *IACR Cryptology ePrint Archive*, 2012:699, 2012. *Cited on page:* 31.

[DSV14] François Durvaux, François-Xavier Standaert, and Nicolas Veyrat-Charvillon. How to Certify the Leakage of a Chip? In Nguyen and Oswald [NO14], pages 459–476. *Cited on pages:* 99 and 109.

[DTV15] Alexandre Duc, Florian Tramèr, and Serge Vaudenay. Better Algorithms for LWE and LWR. In Oswald and Fischlin [OF15], pages 173–202. *Cited on pages:* 4, 43, and 57.

[DV12] Alexandre Duc and Serge Vaudenay. HELEN: a Public-key Cryptosystem Based on the LPN and the Decisional Minimal Distance Problems (Extended Abstract). In *Yet Another Conference on Cryptography*. 2012. *Cited on pages:* 3 and 29.

[DV13a] Alexandre Duc and Serge Vaudenay. HELEN: A Public-Key Cryptosystem Based on the LPN and the Decisional Minimal Distance Problems. In Amr Youssef, Abderrahmane Nitaj, and Aboul Ella Hassanien, editors, *Progress*

in Cryptology - AFRICACRYPT 2013, 6th International Conference on Cryptology in Africa, Cairo, Egypt, June 22-24, 2013. Proceedings, volume 7918 of *Lecture Notes in Computer Science*, pages 107–126. Springer, 2013. *Cited on pages:* 3 and 29.

[DV13b] Alexandre Duc and Serge Vaudenay. TCHo: A Code-Based Cryptosystem. In Evangelos Kranakis, editor, *Advances in Network Analysis and its Applications*, volume 18 of *Mathematics in Industry*, pages 149–179. Springer Berlin Heidelberg, 2013. *Cited on pages:* 30 and 38.

[EG84] Taher El Gamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In *CRYPTO*, pages 10–18, 1984. *Cited on pages:* 2 and 17.

[ELOS15] Yara Elias, Kristin E. Lauter, Ekin Ozman, and Katherine E. Stange. Provably Weak Instances of Ring-LWE. In Gennaro and Robshaw [GR15], pages 63–92. *Cited on page:* 114.

[FMI+06] Marc P. C. Fossorier, Miodrag J. Mihaljevic, Hideki Imai, Yang Cui, and Kanta Matsuura. An Algorithm for Solving the LPN Problem and Its Application to Security Evaluation of the HB Protocols for RFID Authentication. In Rana Barua and Tanja Lange, editors, *INDOCRYPT*, volume 4329 of *Lecture Notes in Computer Science*, pages 48–62. Springer, 2006. *Cited on pages:* 18 and 44.

[FMPR10] Guillaume Fumaroli, Ange Martinelli, Emmanuel Prouff, and Matthieu Rivain. Affine Masking against Higher-Order Side Channel Analysis. In Alex Biryukov, Guang Gong, and Douglas R. Stinson, editors, *Selected Areas in Cryptography - 17th International Workshop, SAC 2010, Waterloo, Ontario, Canada, August 12-13, 2010, Revised Selected Papers*, volume 6544 of *Lecture Notes in Computer Science*, pages 262–280. Springer, 2010. *Cited on page:* 99.

[FO99] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure Integration of Asymmetric and Symmetric Encryption Schemes. In Wiener [Wie99], pages 537–554. *Cited on pages:* 3, 40, and 41.

[FRR+10] Sebastian Faust, Tal Rabin, Leonid Reyzin, Eran Tromer, and Vinod Vaikuntanathan. Protecting Circuits from Leakage: the Computationally-Bounded and Noisy Cases. In Gilbert [Gil10], pages 135–156. *Cited on pages:* 71, 75, 78, and 81.

[FS09] Matthieu Finiasz and Nicolas Sendrier. Security Bounds for the Design of Code-Based Cryptosystems. In Matsui [Mat09], pages 88–105. *Cited on page:* 20.

[FV06] Matthieu Finiasz and Serge Vaudenay. When Stream Cipher Analysis Meets Public-Key Cryptography. In Eli Biham and Amr M. Youssef, editors, *Selected Areas in Cryptography*, volume 4356 of *Lecture Notes in Computer Science*, pages 266–284. Springer, 2006. *Cited on page:* 30.

[Gen09] Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009. `crypto.stanford.edu/craig`. *Cited on page:* 21.

[GHPS12] Craig Gentry, Shai Halevi, Chris Peikert, and Nigel P. Smart. Ring Switching in BGV-Style Homomorphic Encryption. In Ivan Visconti and Roberto De Prisco, editors, *Security and Cryptography for Networks - 8th International Conference, SCN 2012, Amalfi, Italy, September 5-7, 2012. Proceedings*, volume 7485 of *Lecture Notes in Computer Science*, pages 19–37. Springer, 2012. *Cited on page:* 113.

[GHS12a] Craig Gentry, Shai Halevi, and Nigel P. Smart. Fully Homomorphic Encryption with Polylog Overhead. In Pointcheval and Johansson [PJ12], pages 465–482. *Cited on page:* 113.

[GHS12b] Craig Gentry, Shai Halevi, and Nigel P. Smart. Homomorphic Evaluation of the AES Circuit. In Safavi-Naini and Canetti [SC12], pages 850–867. *Cited on page:* 113.

[Gil10] Henri Gilbert, editor. *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings*, volume 6110 of *Lecture Notes in Computer Science*. Springer, 2010.

[GJL14] Qian Guo, Thomas Johansson, and Carl Löndahl. Solving LPN Using Covering Codes. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*, volume 8873 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2014. *Cited on pages:* 18, 19, and 114.

[GJS15] Qian Guo, Thomas Johansson, and Paul Stankovski. Coded-BKW: Solving LWE Using Lattice Codes. In Gennaro and Robshaw [GR15], pages 23–42. *Cited on page:* 114.

[GKPV10] Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. Robustness of the Learning with Errors Assumption. In Andrew Chi-Chih Yao, editor, *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*, pages 230–240. Tsinghua University Press, 2010. *Cited on page:* 22.

[GM11] Louis Goubin and Ange Martinelli. Protecting AES with Shamir's Secret Sharing Scheme. In Preneel and Takagi [PT11], pages 79–94. *Cited on page:* 99.

[GNR10] Nicolas Gama, Phong Q. Nguyen, and Oded Regev. Lattice Enumeration Using Extreme Pruning. In Gilbert [Gil10], pages 257–278. *Cited on page:* 43.

[GP99] Louis Goubin and Jacques Patarin. DES and Differential Power Analysis (The "Duplication" Method). In *CHES*, pages 158–172, 1999. *Cited on pages:* 68 and 80.

[GPS14] Vincent Grosso, Emmanuel Prouff, and François-Xavier Standaert. Efficient Masked S-Boxes Processing - A Step Forward -. In David Pointcheval and Damien Vergnaud, editors, *Progress in Cryptology - AFRICACRYPT 2014 - 7th International Conference on Cryptology in Africa, Marrakesh, Morocco, May 28-30, 2014. Proceedings*, volume 8469 of *Lecture Notes in Computer Science*, pages 251–266. Springer, 2014. *Cited on page:* 107.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Cynthia Dwork, editor, *STOC*, pages 197–206. ACM, 2008. *Cited on pages:* 3, 21, and 30.

[GR10] Shafi Goldwasser and Guy N. Rothblum. Securing computation against continuous leakage. In *CRYPTO*, pages 59–79, 2010. *Cited on page:* 81.

[GR12] Shafi Goldwasser and Guy N. Rothblum. How to Compute in the Presence of Leakage. In *FOCS*, pages 31–40, 2012. *Cited on pages:* 79 and 81.

[GR15] Rosario Gennaro and Matthew Robshaw, editors. *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, volume 9215 of *Lecture Notes in Computer Science*. Springer, 2015.

[GRS05] H. Gilbert, M. Robshaw, and H. Sibert. Active attack against HB^+: a provably secure lightweight authentication protocol. *Electronics Letters*, 41(21):1169–1170, 2005. *Cited on page:* 29.

[GRS08a] Henri Gilbert, Matthew J. B. Robshaw, and Yannick Seurin. Good Variants of HB^+ Are Hard to Find. In Gene Tsudik, editor, *Financial Cryptography*, volume 5143 of *Lecture Notes in Computer Science*, pages 156–170. Springer, 2008. *Cited on page:* 29.

[GRS08b] Henri Gilbert, Matthew J. B. Robshaw, and Yannick Seurin. $HB^\#$: Increasing the Security and Efficiency of HB^+. In Nigel P. Smart, editor, *EUROCRYPT*, volume 4965 of *Lecture Notes in Computer Science*, pages 361–378. Springer, 2008. *Cited on page:* 30.

[GRS08c] Henri Gilbert, Matthew J. B. Robshaw, and Yannick Seurin. How to Encrypt with the LPN Problem. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *ICALP (2)*, volume 5126 of *Lecture Notes in Computer Science*, pages 679–690. Springer, 2008. *Cited on page:* 30.

[GS64] Israel M Gelfand and GE Shilov. Generalized functions. Vol. 1. Properties and operations., 1964. *Cited on page:* 13.

[GSP13] Vincent Grosso, François-Xavier Standaert, and Emmanuel Prouff. Low Entropy Masking Schemes, Revisited. In Aurélien Francillon and Pankaj Rohatgi, editors, *Smart Card Research and Advanced Applications - 12th International Conference, CARDIS 2013, Berlin, Germany, November 27-29, 2013. Revised Selected Papers*, volume 8419 of *Lecture Notes in Computer Science*, pages 33–43. Springer, 2013. *Cited on page:* 99.

[GST13] Daniel Genkin, Adi Shamir, and Eran Tromer. RSA Key Extraction via Low-Bandwidth Acoustic Cryptanalysis. *IACR Cryptology ePrint Archive*, 2013:857, 2013. *Cited on page:* 67.

[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based. In Canetti and Garay [CG13], pages 75–92. *Cited on pages:* 3 and 21.

[HB01] Nicholas J. Hopper and Manuel Blum. Secure Human Identification Protocols. In Colin Boyd, editor, *ASIACRYPT*, volume 2248 of *Lecture Notes in Computer Science*, pages 52–66. Springer, 2001. *Cited on page:* 29.

[HKL+11] Stefan Heyse, Eike Kiltz, Vadim Lyubashesvky, Christof Paar, and Krzysztof Pietrzak. An Efficient Authentication Protocol Based on Ring-LPN. ECRYPT Workshop on Lightweight Cryptography 2007, 2011. *Cited on pages:* 30 and 113.

[HKL+12] Stefan Heyse, Eike Kiltz, Vadim Lyubashevsky, Christof Paar, and Krzysztof Pietrzak. Lapin: An Efficient Authentication Protocol Based on Ring-LPN. In Canteaut [Can12], pages 346–365. *Cited on pages:* 30 and 113.

[Hoe63] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963. *Cited on page:* 13.

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A Ring-Based Public Key Cryptosystem. In Joe Buhler, editor, *ANTS*, volume 1423 of *Lecture Notes in Computer Science*, pages 267–288. Springer, 1998. *Cited on pages:* 30 and 38.

[HPS11a] Guillaume Hanrot, Xavier Pujol, and Damien Stehlé. Algorithms for the Shortest and Closest Lattice Vector Problems. In Yeow Meng Chee, Zhenbo Guo, San Ling, Fengjing Shao, Yuansheng Tang, Huaxiong Wang, and Chaoping Xing, editors, *Coding and Cryptology - Third International Workshop, IWCC 2011, Qingdao, China, May 30-June 3, 2011. Proceedings*, volume 6639 of *Lecture Notes in Computer Science*, pages 159–190. Springer, 2011. *Cited on page:* 43.

[HPS11b] Guillaume Hanrot, Xavier Pujol, and Damien Stehlé. Analyzing Blockwise Lattice Algorithms Using Dynamical Systems. In Rogaway [Rog11], pages 447–464. *Cited on page:* 43.

[HV09] Sean Hallgren and Ulrich Vollmer. Quantum computing. In Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen, editors, *Post-Quantum Cryptography*, pages 15–34. Springer, 2009. *Cited on page:* 2.

[ISW03] Yuval Ishai, Amit Sahai, and David Wagner. Private Circuits: Securing Hardware against Probing Attacks. In *CRYPTO*, pages 463–481, 2003. *Cited on pages:* xvii, 5, 67, 70, 71, 77, 79, 80, 81, 88, 91, 93, 94, and 114.

[IZ89] Russell Impagliazzo and David Zuckerman. How to Recycle Random Bits. In *FOCS*, pages 248–253. IEEE Computer Society, 1989. *Cited on page:* 19.

[JM15] Marc Joye and Amir Moradi, editors. *Smart Card Research and Advanced Applications - 13th International Conference, CARDIS 2014, Paris, France, November 5-7, 2014. Revised Selected Papers*, volume 8968 of *Lecture Notes in Computer Science*. Springer, 2015.

[JN13] Thomas Johansson and Phong Q. Nguyen, editors. *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, volume 7881 of *Lecture Notes in Computer Science*. Springer, 2013.

[Joh82] David S. Johnson. The NP-Completeness Column: An Ongoing Guide. *J. Algorithms*, 3(2):182–195, 1982. *Cited on page:* 20.

[JV10] Ali Juma and Yevgeniy Vahlis. Protecting Cryptographic Keys against Continual Leakage. In *CRYPTO*, pages 41–58, 2010. *Cited on page:* 81.

[JW05] Ari Juels and Stephen A. Weis. Authenticating Pervasive Devices with Human Protocols. In Victor Shoup, editor, *CRYPTO*, volume 3621 of *Lecture Notes in Computer Science*, pages 293–308. Springer, 2005. *Cited on page:* 29.

[Kea93] Michael J. Kearns. Efficient noise-tolerant learning from statistical queries. In S. Rao Kosaraju, David S. Johnson, and Alok Aggarwal, editors, *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing, May 16-18, 1993, San Diego, CA, USA*, pages 392–401. ACM, 1993. *Cited on page:* 17.

[KF15] Paul Kirchner and Pierre-Alain Fouque. An Improved BKW Algorithm for LWE with Applications to Cryptography and Lattices. In Gennaro and Robshaw [GR15], pages 43–62. *Cited on page:* 114.

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. In *CRYPTO'99*, pages 388–397, 1999. *Cited on page:* 67.

[Koc96] Paul C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In *CRYPTO'96*, pages 104–113, 1996. *Cited on page:* 67.

[KPC+11] Eike Kiltz, Krzysztof Pietrzak, David Cash, Abhishek Jain, and Daniele Venturi. Efficient Authentication from Hard Learning Problems. In Paterson [Pat11], pages 7–26. *Cited on page:* 30.

[KS06] Jonathan Katz and Ji Sun Shin. Parallel and Concurrent Security of the HB and HB$^+$ Protocols. In Serge Vaudenay, editor, *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 73–87. Springer, 2006. *Cited on page:* 18.

[KV09] Jonathan Katz and Vinod Vaikuntanathan. Signature Schemes with Bounded Leakage Resilience. In *ASIACRYPT*, pages 703–720, 2009. *Cited on page:* 75.

[LB88] Pil Joong Lee and Ernest F. Brickell. An Observation on the Security of McEliece's Public-Key Cryptosystem. In *EUROCRYPT*, pages 275–280, 1988. *Cited on page:* 20.

[LF06] Éric Levieil and Pierre-Alain Fouque. An Improved LPN Algorithm. In Roberto De Prisco and Moti Yung, editors, *SCN*, volume 4116 of *Lecture Notes in Computer Science*, pages 348–359. Springer, 2006. *Cited on pages:* 18, 19, 26, 27, 44, 53, and 54.

[LP11] Richard Lindner and Chris Peikert. Better Key Sizes (and Attacks) for LWE-Based Encryption. In Aggelos Kiayias, editor, *CT-RSA*, volume 6558 of *Lecture Notes in Computer Science*, pages 319–339. Springer, 2011. *Cited on pages:* 37, 38, and 43.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On Ideal Lattices and Learning with Errors over Rings. In Gilbert [Gil10], pages 1–23. *Cited on pages:* 30 and 114.

[LPR13] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. A Toolkit for Ring-LWE Cryptography. In Johansson and Nguyen [JN13], pages 35–54. *Cited on page:* 113.

[LW11] Dong Hoon Lee and Xiaoyun Wang, editors. *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, volume 7073 of *Lecture Notes in Computer Science*. Springer, 2011.

[Lyu05] Vadim Lyubashevsky. The Parity Problem in the Presence of Noise, Decoding Random Linear Codes, and the Subset Sum Problem. In Chekuri et al. [CJRT05], pages 378–389. *Cited on pages:* 19, 53, and 63.

[Man04] Stefan Mangard. Hardware Countermeasures against DPA ? A Statistical Analysis of Their Effectiveness. In *CT-RSA*, pages 222–235, 2004. *Cited on page:* 102.

[Mat09] Mitsuru Matsui, editor. *Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings*, volume 5912 of *Lecture Notes in Computer Science*. Springer, 2009.

[McE78] Robert J. McEliece. A public-key cryptosystem based on algebraic coding theory. *DSN progress report*, 42(44):114–116, 1978. *Cited on pages:* 30 and 37.

[MJ09] K.V. Mardia and P.E. Jupp. *Directional Statistics*. Wiley Series in Probability and Statistics. Wiley, 2009. *Cited on page:* 22.

[MMT11] Alexander May, Alexander Meurer, and Enrico Thomae. Decoding Random Linear Codes in $\tilde{\mathcal{O}}(2^{0.054n})$. In Lee and Wang [LW11], pages 107–124. *Cited on page:* 20.

[MOP07] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power Analysis Attacks: Revealing the Secrets of Smart Cards (Advances in Information Security)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007. *Cited on pages:* 68, 71, and 108.

[MOS11] Stefan Mangard, Elisabeth Oswald, and François-Xavier Standaert. One for all - all for one: unifying standard differential power analysis attacks. *IET Information Security*, 5(2):100–110, 2011. *Cited on pages:* 102 and 108.

[MP07] Jorge Munilla and Alberto Peinado. HB-MP: A further step in the HB-family of lightweight authentication protocols. *Computer Networks*, 51(9):2262–2267, 2007. *Cited on page:* 29.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In Pointcheval and Johansson [PJ12], pages 700–718. *Cited on page:* 113.

[MP13] Daniele Micciancio and Chris Peikert. Hardness of SIS and LWE with Small Parameters. In Canetti and Garay [CG13], pages 21–39. *Cited on page:* 114.

[MPG05] Stefan Mangard, Thomas Popp, and Berndt M. Gammel. Side-Channel Leakage of Masked CMOS Gates. In Alfred Menezes, editor, *Topics in Cryptology - CT-RSA 2005, The Cryptographers' Track at the RSA Conference 2005, San Francisco, CA, USA, February 14-18, 2005, Proceedings*, volume 3376 of *Lecture Notes in Computer Science*, pages 351–365. Springer, 2005. *Cited on pages:* 100 and 115.

[MR04] Silvio Micali and Leonid Reyzin. Physically Observable Cryptography (Extended Abstract). In *TCC*, pages 278–296, 2004. *Cited on pages:* 67 and 81.

[MS11] Marcel Medwed and François-Xavier Standaert. Extractors against side-channel attacks: weak or strong? *J. Cryptographic Engineering*, 1(3):231–241, 2011. *Cited on page:* 99.

[MT10] Ueli M. Maurer and Stefano Tessaro. A hardcore lemma for computational indistinguishability: Security amplification for arbitrarily weak prgs with optimal stretch. In Daniele Micciancio, editor, *TCC*, volume 5978 of *Lecture Notes in Computer Science*, pages 237–254. Springer, 2010. *Cited on page:* 85.

[MV13] Eric Miles and Emanuele Viola. Shielding circuits with groups. In Boneh et al. [BRF13], pages 251–260. *Cited on page:* 81.

[Ngu11] Phong Q. Nguyen. Lattice Reduction Algorithms: Theory and Practice. In Paterson [Pat11], pages 2–6. *Cited on page:* 43.

[Nie86] H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Problems of Control and Information Theory*, 15(2):159–166, 1986. *Cited on page:* 30.

[NO14] Phong Q. Nguyen and Elisabeth Oswald, editors. *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, volume 8441 of *Lecture Notes in Computer Science*. Springer, 2014.

[NS09a] Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. In *CRYPTO*, pages 18–35, 2009. *Cited on page:* 75.

[NS09b] Phong Q. Nguyen and Damien Stehlé. Low-dimensional lattice basis reduction revisited. *ACM Transactions on Algorithms*, 5(4), 2009. *Cited on page:* 43.

[OF15] Elisabeth Oswald and Marc Fischlin, editors. *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, volume 9056 of *Lecture Notes in Computer Science*. Springer, 2015.

[OMPR05] Elisabeth Oswald, Stefan Mangard, Norbert Pramstaller, and Vincent Rijmen. A Side-Channel Analysis Resistant Description of the AES S-Box. In *FSE*, pages 413–423, 2005. *Cited on pages:* 68 and 80.

[OOV08] Khaled Ouafi, Raphael Overbeck, and Serge Vaudenay. On the Security of HB$^{\#}$ against a Man-in-the-Middle Attack. In Josef Pieprzyk, editor, *ASIACRYPT*, volume 5350 of *Lecture Notes in Computer Science*, pages 108–124. Springer, 2008. *Cited on page:* 30.

[Pat11] Kenneth G. Paterson, editor. *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, volume 6632 of *Lecture Notes in Computer Science*. Springer, 2011.

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In Michael Mitzenmacher, editor, *STOC*, pages 333–342. ACM, 2009. *Cited on pages:* 21 and 30.

[Pet10] Christiane Peters. Information-Set Decoding for Linear Codes over F_q. In Nicolas Sendrier, editor, *PQCrypto*, volume 6061 of *Lecture Notes in Computer Science*, pages 81–94. Springer, 2010. *Cited on page:* 21.

[PJ12] David Pointcheval and Thomas Johansson, editors. *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, volume 7237 of *Lecture Notes in Computer Science*. Springer, 2012.

[PR10] Emmanuel Prouff and Thomas Roche. Attack on a Higher-Order Masking of the AES Based on Homographic Functions. In Guang Gong and Kishan Chand Gupta, editors, *Progress in Cryptology - INDOCRYPT 2010 - 11th International Conference on Cryptology in India, Hyderabad, India, December 12-15, 2010. Proceedings*, volume 6498 of *Lecture Notes in Computer Science*, pages 262–281. Springer, 2010. *Cited on page:* 99.

[PR11] Emmanuel Prouff and Thomas Roche. Higher-Order Glitches Free Implementation of the AES Using Secure Multi-party Computation Protocols. In Preneel and Takagi [PT11], pages 63–78. *Cited on pages:* 77 and 80.

[PR13] Emmanuel Prouff and Matthieu Rivain. Masking against Side-Channel Attacks: A Formal Security Proof. In Johansson and Nguyen [JN13], pages 142–159. *Cited on pages:* xvii, 4, 71, 72, 73, 74, 75, 77, 78, 79, 80, 81, 95, 96, 97, 99, 111, and 115.

[PT11] Bart Preneel and Tsuyoshi Takagi, editors. *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings*, volume 6917 of *Lecture Notes in Computer Science*. Springer, 2011.

[QS01] Jean-Jacques Quisquater and David Samyde. ElectroMagnetic Analysis (EMA): Measures and Counter-Measures for Smart Cards. In *E-smart*, pages 200–210, 2001. *Cited on page:* 67.

[Rab79] M.O. Rabin. Digitalized signatures and public-key functions as intractable as factorization. 1979. *Cited on pages:* 2 and 17.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *STOC*, pages 84–93. ACM, 2005. *Cited on pages:* 3, 18, 21, and 30.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), 2009. *Cited on pages:* 21, 22, 25, 43, 44, 54, 55, and 64.

[Reg10] Oded Regev. The learning with errors problem (invited survey). In *IEEE Conference on Computational Complexity*, pages 191–204. IEEE Computer Society, 2010. *Cited on page:* 25.

[Rei09] Omer Reingold, editor. *Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009, San Francisco, CA, USA, March 15-17, 2009. Proceedings*, volume 5444 of *Lecture Notes in Computer Science*. Springer, 2009.

[RKSF11] Mathieu Renauld, Dina Kamel, François-Xavier Standaert, and Denis Flandre. Information Theoretic and Security Analysis of a 65-Nanometer DDSLL AES S-Box. In Preneel and Takagi [PT11], pages 223–239. *Cited on page:* 99.

[Rog11] Phillip Rogaway, editor. *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, volume 6841 of *Lecture Notes in Computer Science*. Springer, 2011.

[Rot12] Guy N. Rothblum. How to Compute under AC0 Leakage without Secure Hardware. In Safavi-Naini and Canetti [SC12], pages 552–569. *Cited on page:* 81.

[RP10] Matthieu Rivain and Emmanuel Prouff. Provably Secure Higher-Order Masking of AES. In *CHES*, pages 413–427, 2010. *Cited on pages:* 77, 80, 88, 91, 93, and 107.

[RP12] Thomas Roche and Emmanuel Prouff. Higher-order glitch free implementation of the AES using Secure Multi-Party Computation protocols - Extended version. *J. Cryptographic Engineering*, 2(2):111–127, 2012. *Cited on pages:* 99 and 107.

[RS09a] Mathieu Renauld and François-Xavier Standaert. Algebraic Side-Channel Attacks. In Feng Bao, Moti Yung, Dongdai Lin, and Jiwu Jing, editors, *Information Security and Cryptology - 5th International Conference, Inscrypt 2009, Beijing, China, December 12-15, 2009. Revised Selected Papers*, volume 6151 of *Lecture Notes in Computer Science*, pages 393–410. Springer, 2009. *Cited on page:* 111.

[RS09b] Alon Rosen and Gil Segev. Chosen-Ciphertext Security via Correlated Products. In Reingold [Rei09], pages 419–436. *Cited on page:* 31.

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978. *Cited on pages:* 2, 17, and 67.

[RSV09] Mathieu Renauld, François-Xavier Standaert, and Nicolas Veyrat-Charvillon. Algebraic Side-Channel Attacks on the AES: Why Time also Matters in DPA. In Christophe Clavier and Kris Gaj, editors, *Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop, Lausanne, Switzerland, September 6-9, 2009, Proceedings*, volume 5747 of *Lecture Notes in Computer Science*, pages 97–111. Springer, 2009. *Cited on page:* 111.

[RSV+11] Mathieu Renauld, François-Xavier Standaert, Nicolas Veyrat-Charvillon, Dina Kamel, and Denis Flandre. A Formal Study of Power Variability Issues and Side-Channel Attacks for Nanoscale Devices. In Paterson [Pat11], pages 109–128. *Cited on pages:* 100, 101, and 115.

[Rud91] Walter Rudin. *Functional analysis*. McGraw-Hill, Inc., New York, 1991. *Cited on page:* 13.

[SC12] Reihaneh Safavi-Naini and Ran Canetti, editors. *Advances in Cryptology - CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*. Springer, 2012.

[Sch10] Leonard J. Schulman, editor. *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*. ACM, 2010.

[Sho97] Peter W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997. *Cited on page:* 2.

[SMY09] François-Xavier Standaert, Tal Malkin, and Moti Yung. A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks. In *EUROCRYPT*, pages 443–461, 2009. *Cited on pages:* 5, 74, 77, 80, 99, 100, 101, and 103.

[SPV12] François-Xavier Standaert, Christophe Petit, and Nicolas Veyrat-Charvillon. Masking with Randomized Look Up Tables - Towards Preventing Side-Channel Attacks of All Orders. In David Naccache, editor, *Cryptography and Security: From Theory to Applications - Essays Dedicated to Jean-Jacques Quisquater on the Occasion of His 65th Birthday*, volume 6805 of *Lecture Notes in Computer Science*, pages 283–299. Springer, 2012. *Cited on page:* 99.

[SPY13] François-Xavier Standaert, Olivier Pereira, and Yu Yu. Leakage-Resilient Symmetric Cryptography under Empirically Verifiable Assumptions. In Canetti and Garay [CG13], pages 335–352. *Cited on pages:* 67 and 77.

[SS11] Damien Stehlé and Ron Steinfeld. Making NTRU as Secure as Worst-Case Problems over Ideal Lattices. In Paterson [Pat11], pages 27–47. *Cited on page:* 113.

[SSTX09] Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. Efficient Public Key Encryption Based on Ideal Lattices. In Matsui [Mat09], pages 617–635. *Cited on page:* 30.

[Sta15] François-Xavier Standaert. Personnal communication, 2015. *Cited on page:* 116.

[Ste88] Jacques Stern. A method for finding codewords of small weight. In Gérard D. Cohen and Jacques Wolfmann, editors, *Coding Theory and Applications*, volume 388 of *Lecture Notes in Computer Science*, pages 106–113. Springer, 1988. *Cited on page:* 20.

[Str69] Volker Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13(4):354–356, 1969. *Cited on page:* 32.

[Str03] Robert S Strichartz. *A guide to distribution theory and Fourier transforms*. World Scientific, 2003. *Cited on page:* 13.

[SVCO+10] François-Xavier Standaert, Nicolas Veyrat-Charvillon, Elisabeth Oswald, Benedikt Gierlichs, Marcel Medwed, Markus Kasper, and Stefan Mangard. The World Is Not Enough: Another Look on Second-Order DPA. In *ASIACRYPT*, pages 112–129, 2010. *Cited on pages:* 67, 68, 80, 96, and 109.

[SW71] Elias M Stein and Guido L Weiss. *Introduction to Fourier analysis on Euclidean spaces*, volume 1. Princeton university press, 1971. *Cited on page:* 13.

[Var97] Alexander Vardy. The Intractability of Computing the Minimum Distance of a Code. *IEEE Transactions on Information Theory*, 43(6):1757–1766, 1997. *Cited on page:* 20.

[VCS10] Nicolas Veyrat-Charvillon and François-Xavier Standaert. Adaptive Chosen-Message Side-Channel Attacks. In Jianying Zhou and Moti Yung, editors, *ACNS*, volume 6123 of *Lecture Notes in Computer Science*, pages 186–199, 2010. *Cited on page:* 78.

[VGRS12] Nicolas Veyrat-Charvillon, Benoît Gérard, Mathieu Renauld, and François-Xavier Standaert. An Optimal Key Enumeration Algorithm and Its Application to Side-Channel Attacks. In Lars R. Knudsen and Huapeng Wu, editors, *Selected Areas in Cryptography, 19th International Conference, SAC 2012, Windsor, ON, Canada, August 15-16, 2012, Revised Selected Papers*, volume 7707 of *Lecture Notes in Computer Science*, pages 390–406. Springer, 2012. *Cited on page:* 100.

[VGS13] Nicolas Veyrat-Charvillon, Benoît Gérard, and François-Xavier Standaert. Security Evaluations beyond Computing Power. In Johansson and Nguyen [JN13], pages 126–141. *Cited on page:* 100.

[VMKS12] Nicolas Veyrat-Charvillon, Marcel Medwed, Stéphanie Kerckhof, and François-Xavier Standaert. Shuffling against Side-Channel Attacks: A Comprehensive Study with Cautionary Note. In Wang and Sako [WS12], pages 740–757. *Cited on page:* 99.

[Wie99] Michael J. Wiener, editor. *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*. Springer, 1999.

[WS12] Xiaoyun Wang and Kazue Sako, editors. *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, volume 7658 of *Lecture Notes in Computer Science*. Springer, 2012.

[YEM14] Xin Ye, Thomas Eisenbarth, and William Martin. Bounded, yet Sufficient? How to Determine Whether Limited Side Channel Information Enables Key Recovery. In Joye and Moradi [JM15], pages 215–232. *Cited on page:* 116.

Curriculum Vitae

Name: Alexandre Duc
E-mail : alexandre.duc@alumni.epfl.ch
Citizenship : Swiss/French

Education

2011-2015	**PhD in Computer, Communication and Information Sciences** **Area:** Cryptography Supervised by Prof. Serge Vaudenay Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland
2009-2011	**Master in Communication Systems** *Specialization in Information Security* EPFL, Switzerland Master's Thesis awarded with the *Kudelski Prize*
2006-2009	**Bachelor in Communication Systems** *Orientation in Mathematics* EPFL, Switzerland
2005	**Swiss Maturity** (equivalent to an international baccalaureate) Gymnase Auguste Piccard, Lausanne, Switzerland

Academic Honors

2014	**Eurocrypt 2014 Best Paper Award** for the paper *Unifying Leakage Models: from Probing Attacks to Noisy Leakage* cowritten with Stefan Dziembowski and Sebastian Faust EPFL, Switzerland
2013	**Outstanding Teaching Assistant Award** EPFL, Switzerland
2011	**Fellowship** from the doctoral school in Computer and Communication Science 2011. EPFL, Switzerland
2011	**Kudelski Prize** rewarding a Master thesis having significantly contributed to the field of cryptography and information systems security EPFL, Switzerland
2009	**Excellency Scholarship** EPFL, Switzerland

Teaching

- 2013-2015: member of the **teaching commission** of the IC faculty at EPFL.
- **Supervised seven students** for **research** and **implementation-oriented projects**.
- **Teaching Assistant** for **ten different classes**.

Languages

French (native), *English* (fluent), *German* (good)

Technical Skills

- **Programming Languages :** C, C++, Java, Matlab, Python, Scala
- **Programming Concepts :** Compiler construction, Functional/Logic/Imperative programming, Object Oriented programming
- **Operating Systems :** good UNIX skills

Miscellaneous

I play the piano since 1992.